Machine Learning and Alternative Data Analytics for Fashion Finance

Julija Bainiaksinaite

Supervisors: Prof. Philip Treleaven, UCL
Gabriel Straub, BBC

The Thesis submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
University College London

Department of Computer Science
University College London
March 17, 2020

Declaration of Authorship

I, Julija Bainiaksinaite, confirm that the work presented in this thesis is my own except where explicitly stated otherwise by reference or acknowledgement.

Julija Bainiaksinaite _____

Abstract

This dissertation investigates the application of Machine Learning, Natural Language Processing and computational finance to a novel area *Fashion Finance*. Specifically identifying investment opportunities within the Apparel industry using influential alternative data sources such as Instagram. Fashion investment is challenging due to the *ephemeral* nature of the industry and the difficulty for investors who lack an understanding of how to analyze trend-driven consumer brands. Unstructured online data (e-commerce stores, social media, online blogs, news, etc.), introduce new opportunities for investment signals extraction. We focus on how trading signals can be generated from the Instagram data and events reported in the news articles. Part of this research work was done in collaboration with Arabesque Asset Management. Farfetch, the online luxury retailer, and Living Bridge Private Equity provided industry advice.

Research Datasets The datasets used for this research are collected from various sources and include the following types of data:

- Financial data: daily stock prices of 50 U.S. and European Apparel and Footwear equities, daily U.S. Retail Trade and U.S. Consumer Non-Durables sectors indices, Form 10-K reports.
- Instagram data: daily Instagram profile followers for 11 fashion companies.
- News data: 0.5 mln news articles that mention selected 50 equities.

Research Experiments The thesis consists of the below studies:

1. **Relationship between Instagram Popularity and Stock Prices**. This study investigates a link between the changes in a company's popularity (daily followers counts) on Instagram and its stock price, revenue movements. We use cross-correlation analysis to find whether the signals derived from the followers' data could help to infer a company's future financial performance. Two hypothetical trading strategies are designed to test if the changes in a company's Instagram popularity could improve the returns. To test the hypotheses, Wilcoxon signed-rank test is used.

2. **Dynamic Density-based News Clustering**. The aim of this study is twofold: 1) analyse the characteristics of relevant news event articles and how they differ from the noisy/irrelevant news; 2) using the insights, design an unsupervised framework that clusters news articles and identifies events clusters without predefined parameters or expert knowledge. The framework incorporates the density-based clustering algorithm DBSCAN where the clustering parameters are selected dynamically with Gaussian Mixture Model and by maximizing the inter-cluster Information Entropy.

3. **ALGA: Automatic Logic Gate Annotator for Event Detection.** We design a news classification model for detecting fashion events that are likely to impact a company's stock price. The articles are represented by the following text embeddings: TF-IDF, Doc2Vec and BERT (Transformer Neural Network). The study is comprised of two parts: 1) we design a domain-specific automatic news labelling framework ALGA. The framework incorporates topic extraction (Latent Dirichlet Allocation) and clustering (DBSCAN) algorithms in addition to other filters to annotate the dataset; 2) using the labelled dataset, we train Logistic Regression classifier for identifying financially relevant news. The model shows the state-of-the-art results in the domain-specific financial event detection problem.

Contribution to Science This research work presents the following contributions to science:

- Introducing original work in Machine Learning and Natural Language Processing application for analysing alternative data on ephemeral fashion assets.
- Introducing the new metrics to measure and track a fashion brand's popularity for investment decision making.
- Design of the dynamic news events clustering framework that finds events clusters of various sizes in the news articles without predefined parameters.
- Present the original Automatic Logic Gate Annotator framework (ALGA) for automatic labelling of news articles for the financial event detection task.
- Design of the Apparel and Footwear news events classifier using the datasets generated by the ALGA's framework and show the state-of-the-art performance in a domain-specific financial event detection task.
- Build the *Fashion Finance Dictionary* that contains 320 phrases related to various financially-relevant events in the Apparel and Footwear industry.

Acknowledgements

I would like to express my sincere gratitude to my supervisors Prof. Philip Treleaven and Gabriel Straub for their continuous help, encouragement and knowledge. With their guidance, this research work became an invaluable experience of mine and a great foundation for my future career.

I would like to thank Arabesque Asset Management for collaborating with me and providing the support, technical expertise and computing resources for this research work.

I dedicate this thesis to my family, who encouraged and motivated me to pursue a PhD degree. I am grateful for their belief in me.

Research Impact Statement

The research work presented in this thesis is an original study of an application of Machine Learning and Natural Language Processing (NLP) for the alternative data analysis of ephemeral Apparel and Footwear assets. By linking computational finance, Machine Learning, NLP and focusing on a single industry vertical - fashion, we fill in a research gap in the literature and start a new domain-specific research area in *Fashion Finance*. This study opens new opportunities for further academic and industry research.

The findings in this research work demonstrate that signals generated from the non-traditional data sources (social media, news, blogs, etc.) bring additional insights into a fashion company's financial future. We show that a company's popularity on Instagram can be used to model its performance. The work introduces new metrics to measure and track a brand's popularity for investment decision making. Moreover, as the events reported in the news are an important source of information that move the stock prices, we present the original work in generating labelled news datasets for the financial analysis at scale. We build the **A**utomatic **L**ogic **G**ate **A**nnotator framework (ALGA) that is able to automatically label news articles for the financial event classification problem. The framework demonstrates transferability, scalability and interpretability. We also design a fashion news events classifier using the automatically labelled datasets and show the state-of-the-art performance in a domain-specific financial event detection task.

Both finance and fashion industries demonstrated an interest in this study. Our project partner, Arabesque Asset Management, is integrating and testing the models built during this research within their stock trading systems. Meanwhile, fashion companies expressed an interest in the insights on brand popularity, to better understand their competitors.

To summarise, this work opens a new niche research area in *Fashion Finance* that already received interest from the industry and academia.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter introduces the topic of the thesis outlining the problem, experiments and analysis performed during the research work. It gives a brief overview of the trend-driven apparel and footwear market and research opportunities within the fashion investment area.

1.1 Overview

With an increasing amount of data available online such as user engagement and content on social media, blogs, online product reviews, news articles, etc. new approaches to analyzing direct-to-consumer companies for investment decision making can be explored. Currently, there is still a lack of research done on how to analyze ephemeral, trend-driven consumer companies that utilize these alternative data sources available online. This thesis is focusing on exploring the niche Apparel and Footwear industry, designing new methods to generate signals from alternative data for investment decision making. The investment signal extraction methods proposed during this research could be used by the systematic trading funds, individual traders or fund managers to improve the portfolio construction. The key objective of this thesis work is to explore how the alternative data sources such as Instagram and news articles can be used for inferring the future financial performance of fashion companies.

1.2 Research Datasets

The datasets used for this research were collected from multiple sources and included the following types of data:

1. Financial Data: the datasets for this research are obtained from the Bloomberg Terminal, by scraping *EDGAR*[1] - SEC fillings database and also supplied by the project partner Arabesque Asset Management. The financial datasets contain the following:

[1]EDGAR - the Electronic Data Gathering, Analysis, and Retrieval system, is a database containing documents submitted by the U.S. companies as required by U.S. Securities and Exchange Commission - SEC (*https://www.sec.gov/edgar*).

- Stock prices of 50 U.S. and European Apparel and Footwear assets for the period between 2014 and 2019.
- U.S. Retail Trade and U.S. Consumer Non-Durables sectors indices for the period between 2014 and 2019.
- Form 10-K reports, in total 2,947 documents.

2. Social Media Data: daily Instagram profile followers datasets for 11 fashion companies are obtained by scraping the Instagram[2] platform directly and the historic followers data is scraped from the SocialBlade [3] website.

3. News Data: the news articles dataset is obtained from the EventRegistry [4] and by scraping a selection of online fashion news websites and blogs (more details about the scrapers can be found in Chapter 3). The news dataset contains 0.5 mln articles that mention selected 50 U.S. and European equities operating in the Apparel and Footwear industry.

1.3 Research Motivation

The motivation behind this research is to explore the new data sources and build domain-specific frameworks that could improve an alternative approach to analyzing and trading Apparel and Footwear securities. The aim is to utilize the alternative data sources such as social media and news data for the trend-driven consumer companies analysis. The concept of incorporating alternative data sources during the stock-picking process is relatively new and not much research work exists in this area especially with a focus on industry-specific security analysis (e.g. fashion).

1.4 Research Objective

The main objective of this research work is to propose new frameworks for generating fashion stocks trading signals from alternative data sources. There are three primary goals of this research work: 1) evaluate whether the changes in a company's popularity on the social media platform Instagram could infer the future financial performance of a brand; 2) build an unsupervised framework for identifying events in a noisy news dataset; 3) train a fashion industry-specific event classifier that is able to identify financially relevant news which is likely to move company's asset price.

[2]Instagram - a social media platform (*https://www.instagram.com*).

[3]SocialBlade - an online platform that contains historic followers data and information about various user accounts on the platforms such as Instagram, Twitter, YouTube and others (*https://www.socialblade.com*).

[4]EventRegistry - a news aggregator service (*https://www.eventregistry.org*).

1.5 Research Methodologies and Experiments

The following experiments are performed during this research work:

1. **Relationship between Instagram Popularity and Stock Prices**. This study investigates a relationship between the changes in a company's popularity (daily followers counts) on Instagram and its stock price, revenue movements. We use cross-correlation analysis and design two hypothetical trading strategies to test whether the followers' data can be used to derive trading signals. To test the hypotheses, Wilcoxon signed-rank test is used.

2. **Dynamic Density-based News Clustering**. In this experiment, we analyse the characteristics of news events articles and how they differ from the noisy irrelevant news. We build an unsupervised framework that extracts fashion news events from the noisy data without the need to predefine any model parameters. The Dynamic Density-based News Clustering framework incorporates a density-based clustering algorithm DBSCAN where its parameters (cluster density and size) are determined dynamically using Gaussian Mixture Model and a grid search method to find a set of parameters that maximises the inter-cluster Information Entropy.

3. **ALGA: Automatic Logic Gate Annotator for Event Detection.** We present the original work in designing a fashion domain-specific financial news events classifier. The news articles are represented using 3 types of text embeddings: TF-IDF, Doc2Vec and BERT (Transformer Neural Networks). First, we introduce a scalable and automatic solution to label the news articles for the financial event detection problem - ALGA (**A**utomatic **L**ogic **G**ate **A**nnotator) framework. To annotate the dataset, the framework incorporates topic extraction (Latent Dirichlet Allocation) and clustering (DBSCAN) algorithms in addition to other filters. Second, we use the labelled data to train a news event classifier (Logistic Regression) that achieves the state-of-the-art performance in the domain-specific financial event detection task.

1.6 Structure of the Thesis

The thesis comprises of the following chapters:

- **Chapter 2: Background and Literature Review** gives an overview of the research work done on the application of Machine Learning and Natural Language Processing in the finance industry. The chapter also identifies the research gaps and areas for further research.

- **Chapter 3: Research Datasets** the chapter describes all the different types of datasets used during this research work and how they are obtained, stored.

- **Chapter 4: Relationship between Instagram Popularity and Stock Prices** presents the first experiment that investigates a link between a fashion company's Instagram popularity and its financial performance.
- **Chapter 5: Dynamic Density-based News Clustering** the chapter presents the second experiment on the design of the unsupervised framework for news articles clustering.
- **Chapter 6: ALGA: Automatic Logic Gate Annotator for Event Detection** presents a final experiment where we design a unique automatic data labelling framework and train a domain-specific news events detection model.
- **Chapter 7: Conclusion and Future Work** the chapter gives a summary of the key research results and identifies the opportunities for further research in this domain.

1.7 Contribution to Science

This thesis presents the following contributions to science:

- Introducing original work in Machine Learning and Natural Language Processing application for analysing alternative data on ephemeral fashion assets for investment decision making.
- Quantifying the relationship between a brand's Instagram popularity and its financial performance. Introducing the new metrics to measure and track the popularity of a fashion brand for investing and trading.
- Design of the dynamic news events clustering framework that finds events clusters of various sizes in the news articles without the predefined parameters. The framework dynamically identifies the optimum cluster parameters based on the inter-cluster Information Entropy.
- The original research in generating labelled news datasets for financial analysis at scale. We present the Automatic Logic Gate Annotator framework (ALGA) that is able to automatically label news articles for the financial event classification problem. The framework demonstrates transferability, scalability and interpretability.
- Design of the Apparel and Footwear news events classifier using the datasets generated by the ALGA's framework and show the state-of-the-art performance in a domain-specific financial event detection task.
- Build the *Fashion Finance Dictionary* that contains 320 phrases related to various financially-relevant events in the Apparel and Footwear industry.

Chapter 2

Background and Literature Review

This chapter gives a comprehensive background and literature review on a company's valuation and the use of alternative data in investment decisions making. It provides an overview of the factors that have an impact on a company's value and its stock price movements together with a summary of the research work done in using the alternative data to predict stock prices. Moreover, it gives an overview of the Machine Learning and Natural Language Processing techniques used in finance. The chapter also identifies the areas for further research.

2.1 Introduction

Investing in a stock market is one of the most popular investing approaches among institutional and individual investors. When making investment decisions, portfolio managers analyze various data sources such as equity analyst reports, events reported in the news or social media chatter in order to try to estimate a company's future earnings and stock price movements. In 2017 the investment bank Nomura created a proof-of-concept tool that tracked various news, reports and social media data to evaluate their impact on the stock prices (Umezu, 2017). The tool used Natural Language Processing and Machine Learning algorithms to analyse news sentiment and predict the behaviour of the stock prices. The system achieved high accuracy rate showing that Natural Language Processing has the potential to improve investment decision-making process in Asset Management. This chapter gives an overview of 1) factors that have an influence on a company's value; 2) research work done to model stock prices using alternative data sources; 3) Machine Learning and Natural Language Processing methods used to synthesize alternative data and extract useful signals. In addition, we also identify gaps in the research.

2.2 Company's Valuation

2.2.1 Financial Metrics

An investment opportunity appears when the current price of the asset diverges from its true value as perceived by an investor. The key aim is to identify the discrepancies between the price of a stock and value. At the beginning of the XX century, B.

Graham introduced the foundations for value investing, the financial analysis models that are still widely used by investors as a fundamental approach to evaluate companies (Graham, 1949). The company valuation methods could be divided into two categories - fundamental analysis and technical analysis (Romero, 2014). The fundamental analysis describes a company's business operations and financial performance while the technical analysis is used to predict a stock price performance in the future based on its historic price data. The fundamental financial metrics are used for a company's valuation and describe its financial health, performance and future growth prospects. The metrics are derived from a company's Profit and Loss Account, Balance Sheet and other accounting sources reported by a company. The metrics can be subdivided into the following categories:

- Valuation Metrics - describing the prospects of future earnings and growth: Book Value (an asset-based valuation method that is a sum of assets and liabilities of a company), Price-to-Book (P/B) ratio (also referred to as Price-Equity (PE) ratio) - used to compare a company's stock value perceived by the market to its actual book value, Dividend Based Valuation - a stock value estimation based on discounted future dividends to be paid by a company to its investors, Earnings growth - future earnings projection Discounted Cash Flow analysis (DCF).
- Efficiency Metrics - how efficiently a firm is using its resources in order to improve financial performance. The metrics quantify a company's management strategy, activity and how efficient it is, for example, sales revenue per employee, inventory/products turnaround, etc.
- Liquidity Metrics - how easy a firm can liquidate its resources.
- Profitability Metrics - describe profit margins of a company, they depend on the industry where a company operates.
- Growth Metrics - a growth of revenues, market share and market capitalization.

2.2.2 Brand Value and Brand Equity

Brand value is a financial valuation of a company's brand as represented on the company's balance sheet and is referred to as an asset class. When considering consumer companies, especially in fashion, brand value is a very important aspect contributing to the overall company's value. It represents the sale value of a brand (Raggio, 2007). The brand value could be derived using the Customer-Based Brand Equity (CBBE) that is a measure reflecting how customers perceive a brand, their attachment, loyalty, knowledge of a brand. The brand equity should be leveraged to create brand value and therefore those two aspects are closely interlinked (Raggio, 2007).

CBBE can be approached from the two angles - from consumer psychology or from information economics (Christodoulides, 2009).

Considering fashion brands, their future financial performance and equity value highly depend on the Customer-Based Brand Equity which is an intangible asset defined by how consumers perceive the brand. A better understanding of intangible assets such as brand equity could provide a better way of measuring a company's performance (Srivastava, 1998). The brand equity could be defined in three different ways: 1) the total value of a brand represented on the balance sheet as an asset; 2) a measure of attachment to a brand by its customers; 3) loyalty to the brand by its customers (Feldwick, 1996). From the consumer point of view, the study (Aaker, 1991) defined the following aspects of the brand equity - brand awareness, associations or perceived brand image, quality, loyalty, patents, trademarks. Subsequently, another research work (Swait J., 1993) defined the Equalization Price (EP) a measure that quantifies a Customer-Based Brand Equity using multinomial logit. It combines attributes such as brand name, product features and price to give a monetary representation of the brand equity.

The research work (Yoo, 2001) developed a universal model to measure multidimensional brand equity, that is defined in the three-dimensional space of brand loyalty, quality, awareness and associations. Another work (Christodoulides, 2006) measured the brand equity for online retailers and identified five dimensions describing the brand equity that include emotional connection, customer experience, customer service/responsiveness, trust and fulfilment.

The study (Shankar, 2008) presented a model to estimate the brand equity value that incorporates financial data together with customer survey information. The model has two main components a net product value sold by a brand and relative brand importance. The relative brand importance is a measure that captures a brand's image and its impact on consumer purchases, it is driven by the following brand features: brand reputation, uniqueness, fit, associations, trust, innovation and popularity or fame. All the features defining the relative brand value were captured through the consumer surveys (Shankar, 2008).

When building a brand, a company implements one of the following strategies - a price premium or market share strategy, meaning that the brand would either be focusing to sell premium quality products/services or focusing on volume sales and cheaper products/services (Park, 1994). There is no universal metric to evaluate any brand equity across different industries (Christodoulides, 2009), the valuation approach should depend on the market sector where a brand operates and its lifetime (Baker, 2005).

The study (Christodoulides, 2009) identified the key measures that need to be taking into the account when evaluating any brand: a brand vision, organizational culture, understanding of its customers needs and product experience, brand category, industry and value drivers within the industry where it operates, customer experience online (if a brand sells via e-commerce platforms). The brand equity

valuation model should include motivational aspects and their dependency on purchase frequency, emotional connections, functionality.

2.3 Alternative Data in Finance

The traditional approaches for predicting stock price movements can be divided into the two main categories: technical analysis (time-series analysis of historic market data) and fundamental analysis (includes business analysis, geopolitics, financial/economic environment, etc.) (Hu, 2017). A company's financial statements are the foundation for asset valuation but they do not reflect the underlying risks and uncertainties of an asset or its future potential.

Public companies publish their accounts only every 3 months, therefore investors and portfolio managers need to rely on other sources of information that can help them to identify how companies are performing when their sales data is not available. For example, the research work (Luo, 2013) stated that due to the speed of the content on social media, its metrics could allow the investors to monitor and assess the performance of companies and even predict their future business value. The main objective of the study was to explore whether social media is related to a company's equity value. It was found that online blog posts are better indicators of a company's value than online consumer behaviour such as searches or web traffic.

The Internet-enabled a huge increase in user-generated content such as blogs, micro-blogs (Twitter), forums, product reviews and other publicly accessible information sources such as economic data, news, etc. Businesses and investors started to utilize these data sources to gain a competitive advantage on the market. Companies sales data is not available daily meanwhile the social media content is updated every day and spreads very fast, therefore it can be used to predict an impact on the stock market and provide with an instant information source for the investors about a company's performance (Yu, 2012). The study (Yu, 2012) showed that social and conventional media both have an impact on stock prices. Moreover, the researchers discovered that social media sentiment has a stronger impact on a firm's stock performance than conventional media. The researchers found that the sentiment on social media and blogs have a positive effect on a company's returns while the sentiment in forums - negative. It was suggested that when performing the sentiment analysis, it should be business domain-specific to yield better model performance as words from other domains may lead to model inaccuracies (Yu, 2012).

According to the "Wisdom of Crowds" principle (Surowiecki, 2004), a group judgment of an event could be significantly better than a judgment made by a single person. The crowd wisdom emerges from interactions between individuals and aggregation of opinions in groups. Taking into consideration that individuals influence the opinion of other members within a group, such relationships and collection of opinions provide useful information when building data-driven models. During the last decade, peer reviews influenced most of the decision-making processes not

only amongst the consumers but also investors (Research, 2008). The study (Chen, 2014) analyzed how well the content on the SeekingAlpha[1] platform, written by individual investors, could predict the stock market moves. The research found that the posts with negative sentiment and reader comments, represented a group judgment, predicted the negative performance of stocks. This showed that the analysis of crowd wisdom and aggregated opinions could be a good predictive tool for the financial markets. Another study (Nofer, 2014) analyzed investment forums and found that on average the returns achieved by the stock recommendations by the crowd were 0.59% higher than the ones recommended by the investment professionals alone.

The research work (Ranco, 2016) argued that the news sentiment analysis alone does not provide a strong signal for predicting stock price moves, instead the research work suggested to couple the sentiment analysis of the news articles together with their page views. This allows putting more weight on the articles that have a higher number of views that leads to better predictability of the stock movements. Meanwhile, the study (Zhang, 2018) proposed the model to integrate news events and social media sentiment to analyze their joint impact on the stock price movements using coupled matrix and tensor factorization models.

Another example of alternative information used to predict market moves is an analysis of Wikipedia usage patterns. The research (Preis, 2013) investigated the impact of information gathering via Wikipedia before the trading decisions were taken. It found that the page views about the financial terms on Wikipedia increased just before the financial crisis in 2008, this implies that before making the trading decisions investors tend to research the implications via the Internet.

The subsequent study (Dimpfl, 2016) analyzed the volume search queries for Dow Jones stock market index on Google and how they relate to its price movements. The research found a strong relationship between the number of queries and the movement of the index. This showed that the increase in the number of search queries is followed by the stock market volatility next day. This phenomenon is especially apparent during the high volatility periods, e.g. the financial crisis in 2008.

2.3.1 News Events

The events reported in the news are important evidence of stock price changes (Ding, 2014). An ability to extract the events from the news may give additional insights into the stock prices behaviour. The research work (Atkins, 2018) showed that information extracted from the news articles can predict the direction of asset volatility better than a price direction. Meanwhile, another study (Li, 2015) found that summarized articles predict the stock price movement better than a full article.

Below we review the research studies that use news content to depict the relationship between the new information and a company's returns. Here, we focus on

[1]SeekingAlpha - a crowd-sourced content platform for the financial industry (*https://seekingalpha.com*).

studies that used news events rather than news sentiment. Two main research directions exist that mainly differ by how the news articles are preprocessed before they are used for further analysis: 1) articles are represented by the headlines or the headlines and the bodies; 2) articles are represented by the event tuples (E) in the form of E = *(Actor, Action, Object, Time)* or the key topics that are extracted from the articles.

The most ubiquitous method is to represent an article by either its title or the title and the body, the text is then encoded using methods such as Bag-of-Words (Lavrenko, 2000; Schumaker, 2009; Luss, 2009) (the negative side of the Bag-of-Words approach is that it does not capture the relationship between different entities, word order in the sentences, synonyms, etc.), paragraph vectors (Akita, 2016), word (Word2Vec) (Mikolov, 2013) or sentence (Doc2Vec) embeddings (Hu, 2017; Liu, 2018a; Merello, 2018; Yang, 2019). The study (Li, 2018) proposed a framework that used Restricted Boltzmann Machines (RBM) to create document vectors. Besides, a few studies built news embeddings by concatenating the news titles with technical indicators (Vargas, 2017; Oncharoen, 2018; Liu, 2019). Meanwhile, the research work (Tan, 2019) utilized the tensor-based approach to integrate the text and technical data instead of concatenating it.

After the article embeddings are created they are used to model the relationship with stock prices across different time windows e.g. same-day closing price, next day price, etc. Researchers implemented a range of different Machine Learning algorithms to model the relationship: Long Short-Term Memory (LSTM) (Akita, 2016; Chang, 2016; Tan, 2019), bidirectional Gated Recurrent Unit (GRU) (Huynh, 2017), Attention Mechanisms (Vaswani, 2017) together with Recurrent Neural Networks (RNN) (Liu, 2018a; Liu, 2019), hybrid Convolutions Neural Networks (CNN) models (Deng, 2019).

The most common approach to study stock price changes based on news is to utilize the event study methodology (Craig MacKinlay, 1997; Konchitchki, 2011). Hence, another research direction is focused on detecting financially relevant events in the news and use them to model a company's returns. The event extraction methods are commonly classified into Machine Learning, knowledge-based and hybrid approaches (Han, 2018).

One way to extract the events from the news is by using topic modelling to find which topics have an impact on the stock price movements. The research works (Feuerriegel, 2016; Feuerriegel, 2018) analysed which topics reported in the news yield non-zero stock market returns. To extract the topics from the articles, scholars used the Latent Dirichlet Allocation (LDA) algorithm. Although, the key problem with the LDA method is that the number of topics is fixed (40 topics) (Feuerriegel, 2016) and requires an expert knowledge to predefined the number of topics within the data which in the news analysis scenario is unknown. Another study (Jiamiao Wang, 2017) proposed a density-based clustering method for extracting emerging new topics.

In addition to the topic modelling, researchers used dependency parsing to extract the events. The study (Ding, 2014) implemented this method to get the event (E) representations and then used Support Vector Machines (SVM) and Deep Neural Networks to model the relationship between the events and a company's returns. The events were encoded as sparse one-hot vectors. To reduce the sparsity of the structured event vectors, the work (Ding, 2015) introduced event embeddings that were trained on a news corpus using Neural Tensor Network (NTN). The event embeddings can capture syntactic and semantic information of events but not the relationship to other similar events. The subsequent studies (Ding, 2016; Liu, 2018b; Deng, 2019) improved the event embeddings further by combining a knowledge graph, article bodies together with event representations to train the embeddings.

The majority of the studies mentioned above create the training data for the task by labelling the articles based on the stock price movement. The key issue with this approach is that the news and stock price data is extremely noisy and it is hard to know what truly drives the price movements. It is important to differentiate the useful news from the noise and explain the reason for stock price movements (Yang, 2018). The research work (Hu, 2017) tried to solve this problem by proposing the attention-based RNN model to predict returns based on the news sequence. The effectiveness of predictive models depend on the quality of the articles, therefore it is important to differentiate between the useful and non-useful online content (Hu, 2017). A large quantity of the online content is low quality and may contain rumours. To tackle this problem, the work (Hu, 2017) proposed the Hybrid Attention Networks model to predict stock price movement based on news events sequence. The model consists of attention-based RNN and self-paced learning. In this research work, three principles of how humans analyse the news articles are explored:

1. Sequential context-dependency - a broad analysis of diverse news sources.
2. Diverse influence - different types of news have different impact on the stock price.
3. Effective and efficient learning - the periods during which news appears are not consistent or news is too vague to make a prediction.

To capture sequential context dependency and diverse news influence the study (Hu, 2017) designed HAN (Hybrid Attention-based Network).

Nevertheless, none of the studies tried to analyse the training data more in-depth to understand what the models are learning and how well they can differentiate between the useful news information/events and the noisy data.

There is a gap in the research for domain-specific event detection methods especially with a focus on the different financial industries mainly due to the lack of labelled news corpus for training such models. The study (Han, 2018) developed a business events extraction (semi-supervised) framework for Chinese financial news. The method consists of the following steps: define the business event taxonomy (a

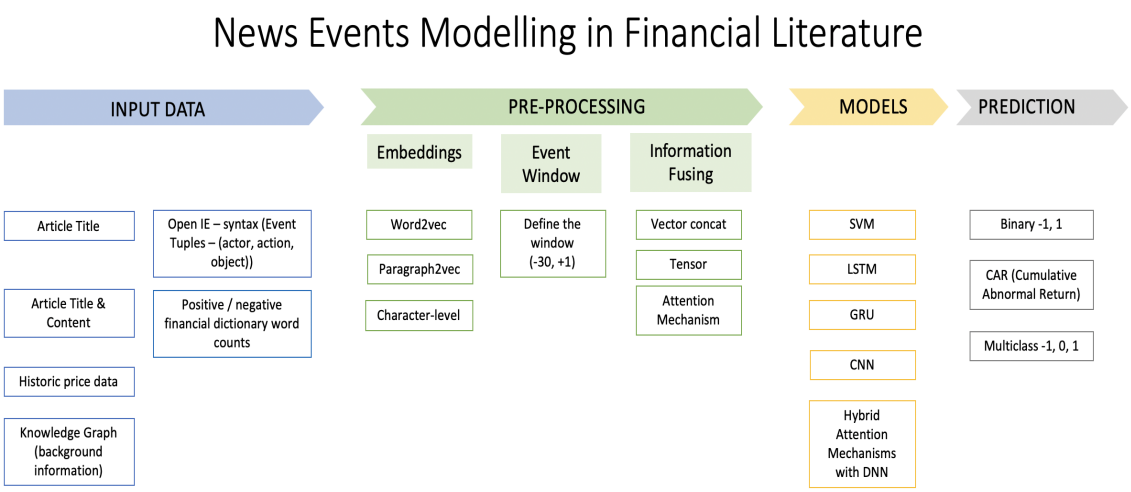


FIGURE 2.1: A summary of methods used to analyse news data for financial prediction found in the literature.

pre-defined vocabulary of business event types); expand the dictionary using word embeddings; extract the news articles based on the relevant event types, and using the articles with identified events find the relationship between the stock prices and news. Another study (Yang, 2018) designed an automatic article labelling technique for Chinese news articles and used to train news event extraction models. The Figure 2.1 presents a summary of the news analysis methods found in financial literature.

2.3.2 Sentiment

Over the last decade, there has been a lot of interest and work done in analyzing the user sentiment on the social media platforms (e.g. Twitter, Reddit [2], StockTwits [3]), in the news articles or blogs for predicting stock price movements or future sales of a product. The work (Sohangir, 2018) compared different Natural Language Processing models using the traditional bag-of-words approach and Deep Learning methods to predict stock price movement using the sentiment extracted from the investors' messages on StockTwits platform. The research implemented Deep Learning models - Long Short-Term Memory (LSTM) and Convolution Neural Networks (CNN). The study found that only the CNN model outperformed the traditional bag-of-words approach for sentiment prediction. In comparison with the traditional data mining techniques, Deep Learning models transform the data through more layers and in theory can extract semantics and word relationships better. In traditional data mining, word frequencies in a document are considered while the sequence and order of the words are not taken into the account. Meanwhile, the position of the word in the body of the text is important as it can change the overall sentiment. The research (Sohangir, 2018) found that the CNN models can be used to predict the sentiment of investors messages and the future movements of the stock market.

[2]Reddit - a social news aggregation and forum platform *https://www.reddit.com*.
[3]StockTwits - a social media platform for finance professionals *https://www.stocktwits.com*

Moreover, the language semantics are very important when analysing text for financial forecasting. According to the research study (Merello, 2018), the words '*bull*' or '*bear*' used in financial text do not have any relationship with animals in the same way as the pre-trained word embeddings (e.g. Word2Vec) would have trained on a general language corpus, therefore it is important to introduce domain-specific word embeddings into the models when analysing sentiment of the text or detecting events. The research work (Merello, 2018) used L. McDonald's financial domain dictionary (Loughran, 2011) in the news articles representations that helped to inject domain-specific language semantics during the modelling process.

The social chatter on the platforms such as Twitter, Instagram, Facebook, etc. contain not only the text data but also emojis - graphical representations of user's emotions or parts of a conversation (Miller, 2016). In 2017, 56.5% of the posts on Instagram included at least one emoji (Research, 2017). The study (Eisner, 2016) trained embeddings for emojis using their Unicode descriptions and released Emoji2Vec model. The research showed that by augmenting Word2Vec with the emoji embeddings helps to improve an overall accuracy for a classification task, especially for the text classification that contains emojis. The subsequent study (Wijeratne, 2017) trained emoji embeddings using emoji descriptions from the EmojiNet [4] website and improved the sentiment analysis of tweets by up to 63.6%, this model outperformed the one generated by the previous study (Eisner, 2016) (both research works using the same twitter data).

One of the key issues with applying Machine Learning models for sentiment modelling tasks is the scarcity of manually labelled data (ground-truth labels). To overcome this problem, the work (Zhou, 2018) proposed to use emojis as labels. The research work (Zhou, 2018) used Twitter conversations with emojis to train Neural Networks models and showed that emojis could successfully portrait the emotion of the sentence.

The work (Felbo, 2017) used LSTM model together with the dataset of 1.2 bln tweets to train the model that predicts the emotional content of a tweet based on an emoji in the text. The research showed that the emojis can classify the emotional content correctly in most of the cases. Previous research work incorporated a manual specification of an emotional category of an emoji or learnt representation of an emoji (embedding) (Eisner, 2016). The main drawback of emoji embeddings is that they do not capture the change in emoji meaning over time.

2.4 Machine Learning

Machine learning is a computer science field that studies how a machine can learn from the data without being explicitly programmed. Machine learning is subdivided into the following main categories: Supervised Learning, Unsupervised Learning

[4]EmojiNet - a website that contains emoji descriptions (*http://emojinet.knoesis.org*).

(Murphy, 2012 and Reinforcement Learning (not covered in this study). In the Supervised Learning problem the objective is to learn the mappings from inputs x (features) to outputs y (target), given a training set with correctly labelled input and output values as per equation 2.1, where D is a training set, N - number of training examples.

$$D = (x_i, y_i)_{i=1}^{N} \tag{2.1}$$

Training input x_i is a m dimensional vector ($x_i \in \mathbb{R}^{m+1}$), each x_i vector has corresponding output vector y_i. Depending on the type of the output values y_i, the problem can be either classification or regression, if y_i belongs to a finite set $y_i \in 1, ..., C$, it is a classification problem and if the y_i is a real valued scalar - regression. The examples of Supervised Learning algorithms include - Linear Regression, Logistic Regression, SVM, Neural Networks, etc.

On the other hand, in the Unsupervised Learning problem only the input dataset is given:

$$D = (x_i)_{i=1}^{N} \tag{2.2}$$

Here, the objective is to find patterns in the data, identify clusters or groups the data could belong to. Unsupervised Learning can be used for anomaly detection, news articles clustering, etc. The examples of Unsupervised Learning algorithms include - K-Means clustering, Gaussian Mixture Model, DBSCAN, etc.

Below we give an overview of Machine Learning algorithms that were used during this research work together with commonly used Neural Networks architectures (e.g. CNN, RNN, LSTMs).

Logistic Regression

Logistic Regression is a Supervised Learning classification algorithm. In a binary classification case, for a given training set $\mathcal{D}$ with N training samples $\mathcal{D} = \{(x_i, y_i)|i = 1 : N\}$, where $x_i \in \mathbb{R}^{m+1}$ and $y_i \in \{0, 1\}$, Logistic Regression is defined as per equation (2.3) where $\theta \in \mathbb{R}^{m+1}$ and cost function (equation 2.4) is used to minimise the loss during the training to find the optimal model weights (Murphy, 2012).

$$h_\theta(x_i) = \frac{1}{1 + \exp(-\theta x_i)} \tag{2.3}$$

$$J(\theta) = -\frac{1}{N} \sum_{i=1}^{N} y_i log(h_\theta(x_i)) + (1 - y_i) log(1 - h_\theta(x_i)) \tag{2.4}$$

Neural Networks

Neural Networks are another example of Supervised Learning algorithms. Their architecture is comprised of a series of combinations of the basis functions (equation 2.5) so that each basis function itself is a non-linear function of a linear combination

of the inputs (Bishop, 2006). For the classification task $f()$ is a nonlinear equation e.g. Sigmoid (see equation 2.7) and w_j- a coefficient/adaptive parameter.

$$y(x, w) = f(\sum_{j=1}^{T} w_j \phi_j(x)) \tag{2.5}$$

Basic Neural Network is a series of functional transformations that are constructed as per below:

1. Linear combinations of input variables $(x_1, ..., x_m)$ for $x_i \in \mathbb{R}^{m+1}$ are constructed using the equation 2.6, where (1) - refers to the values being in the first layer of the network, w_{ji} - weights, w_{j0} - bias parameter.

$$a_j = \sum_{i=1}^{m} w_{ji}^{(1)} x_i + w_{j0}^{(1)} \tag{2.6}$$

2. After computing the linear combinations of input variables (a_j) within the layer, they are transformed using activation function $h()$. The most commonly used activation functions are Sigmoid (2.7) and ReLU (Rectified Linear Unit) (2.8).

$$z_j = h(a_j) = \frac{1}{1 + exp(-a_j)} \tag{2.7}$$

$$z_j = h(a_j) = max(0, a_j) \tag{2.8}$$

   The number of activation functions corresponds to the number of hidden units in the network that is another parameter used to optimize the network.

3. Subsequently, the output values (z_j) are combined linearly and transformed again using selected activation function $h()$ (see equation 2.9).

$$z_k = h(a_k) = h(\sum_{j=1}^{T} w_{kj}^{(2)} z_j + w_{k0}^{(2)}) \tag{2.9}$$

4. In the last output layer, the Neural Network outputs the predicted target value that during the network training is compared to the true value and based on the error the network adjusts the weights using the backpropagation method. For the classification task, the most commonly used transformation function for the output layer is Softmax (equation 2.10). It transforms the network output to be bounded between [0,1] giving class probabilities. The Softmax used for both binary and multi-class classification problems.

$$y_k(x, w) = \frac{exp(a_k(x, w))}{\sum_j exp(a_j(x, w))} \tag{2.10}$$

Network training and Backpropagation During the Neural Network training, the main objective is to minimise the loss function to find the optimum model weights w that correctly describe the data. The Neural Network training task is split into two parts (Murphy, 2012):

1. **Forward propagation:** the Neural Network is created (number of layers and neurons are defined), random weight values for the network layers are initialised. Then the input data samples are forward propagated through the network. In the final output layer, the predicted output is calculated and then compared with the truth value to calculate the prediction error. The equation 2.11 defines a loss function (with respect to the model weights) and is used to calculate the error. Here, $\hat{y}$ - is a predicted target value, y_t - truth target value.

$$Loss(w) = \frac{1}{2}(\hat{y} - y_t)^2 \tag{2.11}$$

   For the classification problem, the cross-entropy loss (2.12) with the respect to the layer weights is used to calculate the error where t_n - target value, y_n - predicted value.

$$Loss(w) = -\sum_{n=1}^{N}\{t_n ln(y_n) + (1 - t_n ln(1 - y_n)\} \tag{2.12}$$

2. **Backpropagation:** during the backpropagation, the gradients of the loss function with respect to each weight in every layer are calculated (using the chain rule), then minimized using the gradient descent to find the optimum weight values for the network. The backpropagation is repeated with the new weights until the optimum weights are found that give the lowest error during the classification.

Below is an overview of the most commonly used Neural Networks architectures.

Recurrent Neural Network (RNN) Unlike the traditional Neural Network, Recurrent Neural Network carries information from previous iterations (Elman, 1990). RNN has loops within the structure allowing the information to persist. They are used to learn a time-dependent sequence.

In general the RNN output layer h_t at the time t is defined as a non-linear transformation of two aspects - the current input x_t and the output from from the previous hidden layer h_{t-1} (Liu, 2017), h_t - transformations through the hidden layers is defined by the equation 2.13, where f is a non-linear transformation function - Sigmoid 2.7 that is applied to the hidden layers. U and V are the weight matrices between the layers, b - the bias vector.

$$h_t = f(Uh_{t-1} + Vx_t + b) \tag{2.13}$$

The probability of class k, $k \in K$, as an output for the classifier is defined by the equation 2.14, where s is a sequence.

$$P(y_t = k|s, \theta) = \frac{exp(w_k^T h_t)}{\sum_{k=1}^{K} exp(w_k^T h_t)} \tag{2.14}$$

The objective during the model training is to minimize the negative log-likelihood (equation 2.15).

$$J(\theta) = \sum_{t=1}^{T} \sum_{k=1}^{K} y_t log P(y_t = k|s, \theta) \tag{2.15}$$

Long Short-Term Memory (LSTM) is another type of RNN designed to model the long term dependencies within the network (Hochreiter, 1997). LSTM network contains hidden units called memory blocks. A memory block is comprised of four parts:

- Memory cell c - a neuron with a self-connection.
- Input gate i - controls the input signal into the neuron.
- Output gate o - controls the impact of the neuron activation on the subsequent neurons.
- Forget gate f - resets the current state.

Below equations 2.16-2.20 give an overview of how the memory block is updated at each time step t:

$$i_t = \sigma(U_i \boldsymbol{h}_{t-1} + V_i \boldsymbol{x}_t + C_i \boldsymbol{c}_{t-1} + \boldsymbol{b}_i) \tag{2.16}$$

$$f_t = \sigma(U_f \boldsymbol{h}_{t-1} + V_f \boldsymbol{x}_t + C_f \boldsymbol{c}_{t-1} + \boldsymbol{b}_f) \tag{2.17}$$

$$c_t = i_t \odot g(U_c \boldsymbol{h}_{t-1} + V_c \boldsymbol{x}_t + \boldsymbol{b}_c) + \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} \tag{2.18}$$

$$o_t = \sigma(U_o \boldsymbol{h}_{t-1} + V_o \boldsymbol{x}_t + C_o \boldsymbol{c}_t + \boldsymbol{b}_o) \tag{2.19}$$

$$h_t = o_t \odot h(\boldsymbol{c}_t) \tag{2.20}$$

Where U_k, V_k, C_k represent the weight matrices between two consecutive hidden layers- between the input and output layers and between two consecutive cell activations which are represented by gate k, b_k is the bias vector and σ is the Sigmoid activation function (see equation 2.7), g and h are input and output tanh activation function. The symbol $\odot$ denotes element-wise multiplication.

Convolution Neural Network (CNN) Another type of Neural Network architecture that is commonly used for Language Modelling tasks such as word tagging, search, sentence modelling (Sohangir, 2018). CNN was developed for Computer Vision initially although showed good performance in Natural Language Modelling tasks too. The research work (Kim, 2014) used Convolution Neural Network to train sentence classifier. The work (Sohangir, 2018) compared three Neural Networks models for sentiment analysis and found that CNN performed the best. Unlike traditional Neural Network, the CNN network has a three- dimensional architecture where the third dimension is layers. CNN has three main principles - fields, shared weights and pooling. The inputs are two dimensional, in the hidden layer each neuron does not take the entire input but only a small sample of it (local receptive field), then the weights are generated for each local receptive field together with the overall bias. The sampling continues throughout the entire 2-dimensional input space. The same weights and bias are used across all local perceptive fields. The output from the hidden neuron $n_{j,k}$ is defined by the equation 2.21, where $w_{l,m}$ is 5×5 array of weights, $a_{j+l,k+m}$ input activation.

$$n_{j,k} = \sigma(b + \sum_{l=0}^{4} \sum_{m=0}^{4} w_{l,m} a_{j+l,k+m}) \tag{2.21}$$

In this way, the first hidden layer of CNN detects the same feature but in different locations across the 2- dimensional input space. To detect all features, multiple hidden layers (also known as feature maps) are produced using this approach. After the convolution layers are produced, the pooling layers simplify the information into the condensed feature map.

Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a generative probabilistic model (Blei, 2003) often used for topic extraction. It is an Unsupervised Learning algorithm. It is assumed that there are K groups/topics ($\beta_{1:k}$) across the corpus where each group is defined as a distribution over a fixed vocabulary (β_k). LDA is defined as the joint distribution (2.22) where θ_d is the topic proportions for document d and $\theta_{d,k}$ - topic k proportion in the document d, the topic assignment for the document d is z_d, where $z_{d,n}$ is the topic assignment for the n^{th} in d. The observed words in the document d - w_d and $w_{d,n}$ the n^{th} word in d (word from the corpus vocabulary). The posterior is defined as per equation (2.23). More details on Latent Dirichlet Allocation and its implementation are provided in the Experiment 3 (Chapter 6)

$$\begin{aligned} p(\beta_{1:k}, \theta_{1:D}, z_{1:D}, w_{1:D}) = \prod_{i=1}^{K} p(\beta_i) \prod_{d=1}^{D} p(\theta_d) \\ \left(\prod_{n=1}^{N} p(z_{d,n}|\theta_d) p(w_{d,n}|\beta_{1:K}, z_{d,n}) \right) \end{aligned} \tag{2.22}$$

$$p(\beta_{1:k}, \theta_{1:D}, z_{1:D}|w_{1:D}) = \frac{p(\beta_{1:k}, \theta_{1:D}, z_{1:D}, w_{1:D})}{p(w_{1:D})} \tag{2.23}$$

DBSCAN

Density Based Spatial Clustering of Applications with Noise (DBSCAN) is a density-based clustering model designed to find clusters of various shapes (unlike K-Means clustering can find only convex-shaped clusters) without a need to pre-define a number of clusters as an input parameter (c.f. K-Means clustering algorithm) (Ester, 1996). The algorithm finds clusters by identifying densely populated areas from the low-density areas based on two parameters: 1) eps - density radius - the pairwise distance between samples and 2) $minPts$ - minimum number of points within the cluster. A number of data points within a predefined radius (eps) is counted to calculate a point's density within the dataset (Heidari, 2019).

Gaussian Mixture Model

In Machine Learning Gaussian Mixture Model (GMM) is commonly used for unsupervised data clustering. GMM is a probabilistic model that assumes that the samples within the dataset are generated from a number (mixture) of *K* multivariate Gaussian distributions, where each distribution has a set of parameters - mean $\boldsymbol{\mu}_k$ vector and covariance matrix $\boldsymbol{\Sigma}_k$. Gaussian Mixture Model is defined as per equation 2.24, where π_k - is mixing weights, $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$ (Murphy, 2012).

$$p(x_i|\theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{2.24}$$

The parameters of the distributions are determined by Expectation Maximization (EM) algorithm - iterative method to compute maximum likelihood estimates of model parameters to fit the data with latent variables. More details on Gaussian Mixture Model implementation are described in Experiment 2 (Chapter 5).

2.5 Natural Language Processing

To analyze alternative unstructured data sources and extract meaningful information, Natural Language Processing (NLP) techniques are used. NLP is a set of computational techniques, models for analysis and interpretation of natural human language and speech. NLP tasks are divided into the following areas: text classification,

information retrieval, information summarizing, machine translation, text generation, sentiment analysis and opinion mining, natural language inference, grammatical text analysis, word sense disambiguation and speech recognition (Liu, 2017). Depending on the task, the key NLP steps/methods used for text analysis, modelling are described below.

2.5.1 Text Preprocessing

The initial step during any text analysis or modelling task is to normalize and clean the raw text. Text normalization is the process of converting the text into the standard form, make it lower case, deleting symbols, etc. It depends on the task how much preprocessing is done on the text but most commonly the following techniques are used.

- Lower case - making the all text lower case.
- Deleting symbols, punctuation, converting text into ASCII format.
- Tokenization - separating the text into tokens.
- Lemmatization - words are converted into their first form.
- Stemming - words are stripped to their stems e.g. using The Porter Stemmer algorithm (Porter, 1980)).

Part of Speech Tagging (POS) Specifically for information extraction tasks it is useful to extract nouns, articles or verbs, etc. from the text and use only them for further analysis. For this Part of Speech Tagging method is used. POS is a linguistic technique of subdividing the sentence into the syntactic categories such as noun, verb, pronoun, preposition, adverb, conjunction, participle and article. Hidden Markov Models algorithm can be used for this task.

On the other hand, text preprocessing introduces a problem of information loss. The techniques mentioned above normalize text and eliminate some important information present in its structure. Although the state-of-the-art language embeddings i.e. BERT (see the section below) tries to solve this issue by pre-training language models on original text with a minimum amount of pre-processing.

2.5.2 Text Representation

After the raw text is cleaned and normalized, the next step is to convert it into the numeric form to be used in Machine Learning models e.g. Logistic Regression or Neural Networks. The text can be represented at various levels (depending on the task), for example at character level, word level, sentence or document level. Below is an overview of the key methods used for text representations.

One-Hot Encoding

One-hot encoding is a method by which every word in a vocabulary is represented by a vector x_w, $x_i \in \mathbb{R}^N$ where N is a size of the vocabulary (a collection of unique words across all the documents in the corpus). One-hot vectors are sparse containing 0 values everywhere apart from a single 1 that presents the specific word location in the vocabulary.

One way to reduce the dimensions of one-hot vectors is to use Singular Value Decomposition (SVD). It is a method used for dimensionality reduction, it finds the most important dimensions within the data (where there are most variations) and rotates the axes of the original dataset into the new space. It was first introduced in Latent Semantic Indexing (LSI) (Deerwester, 1988).

TF-IDF

is a statistical measure that evaluates the importance of each word in the document collection (Sparck Jones, 1972). One of the most commonly used method to represent the text in a numerical format. TF-IDF is a multiplication of two factors - word (term) frequency (TF) (Luhn, 1957) in the document and inverse document frequency (IDF) (Sparck Jones, 1964). The score is defined by the equation 2.25, where tf_{ij} is the term's i frequency in the document j, N - the total number of documents in the collection, df_j is the number of documents in which the term i occurs. The inverse document frequency gives a higher score to the terms that occur in fewer documents and are unique. The main weakness of the TF-IDF score is that the order of the words in sentences and word semantics are ignored.

$$TF\text{-}IDF = tf_{ij} \times log(\frac{N}{df_j}) \tag{2.25}$$

Word2Vec Embeddings

The disadvantage of one-hot vectors or TF-IDF method is that they do not capture words similarities, do not scale and do not generalize for out of vocabulary (unseen) words. Therefore, the research study (Mikolov, 2013) introduced an unsupervised Neural Networks based approach to compute continuous vector representations for words trained on a large text corpus. Two models for training the Word2Vec embeddings were proposed:

1. Continuous Bag-of-Words model (CBOW) with an objective to predict a word using context of 8 words that surround it (4 words before and 4 after). A Feed-Forward Neural Network with a single fully-connected hidden layer is used to predict the target word.

2. Continuous Skip-gram model (Skip-gram) is an opposite model to CBOW, it takes a single word and predicts context words within a range from a given

word. Similarly, as in the CBOW model, Skip-gram uses the same Neural Network architecture.

After the Neural Networks are trained the weights of the hidden layer are used as word embeddings.

Doc2Vec Embeddings

Doc2Vec is an Unsupervised Learning method for learning text representations of various lengths e.g. sentences, paragraphs or documents (Le, 2014). The framework is based on a Neural Network model similar to the method used to learn Word2Vec embeddings (Mikolov, 2013). The objective is to learn the document vector representations by predicting the surrounding words in the contexts that are sampled from the documents.

BERT

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained language representation model (Devlin, 2018) used to encode text. BERT is a multi-layer bidirectional Transformer encoder (see the Figure 2.2) that utilizes the implementation of a Transformer architecture and bidirectional self-attention. The Figure 2.3 shows the Transformer architecture presented in the original research work (Vaswani, 2017). BERT model is trained on a large corpus of a general language. One of the advancements in NLP transfer learning from language models is to pre-train a model on a language model objective first and then fine-tune the pre-trained model on the task-specific text corpus (Devlin, 2018). Therefore, BERT text encoder can be used as an original general language encoder or a fine-tuned encoder for a specific task.

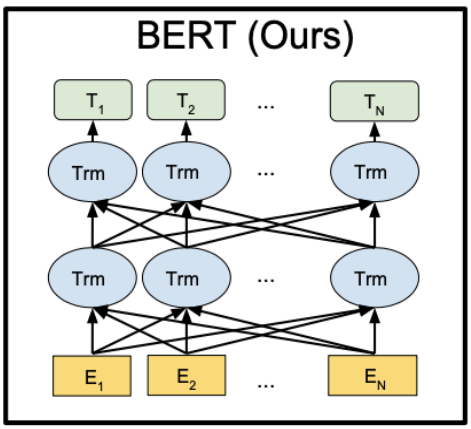


FIGURE 2.2: BERT - pre-training model architecture that uses a bidirectional Transformer. The model representations are jointly conditioned on left and right context in all layers (Devlin, 2018).

2.5.3 Text Similarity

In many NLP tasks, it is useful to measure the similarity between the documents or words. There are many different distance metrics used to calculate the similarity between the two vectors. Below we give an overview of three most commonly used similarity metrics.

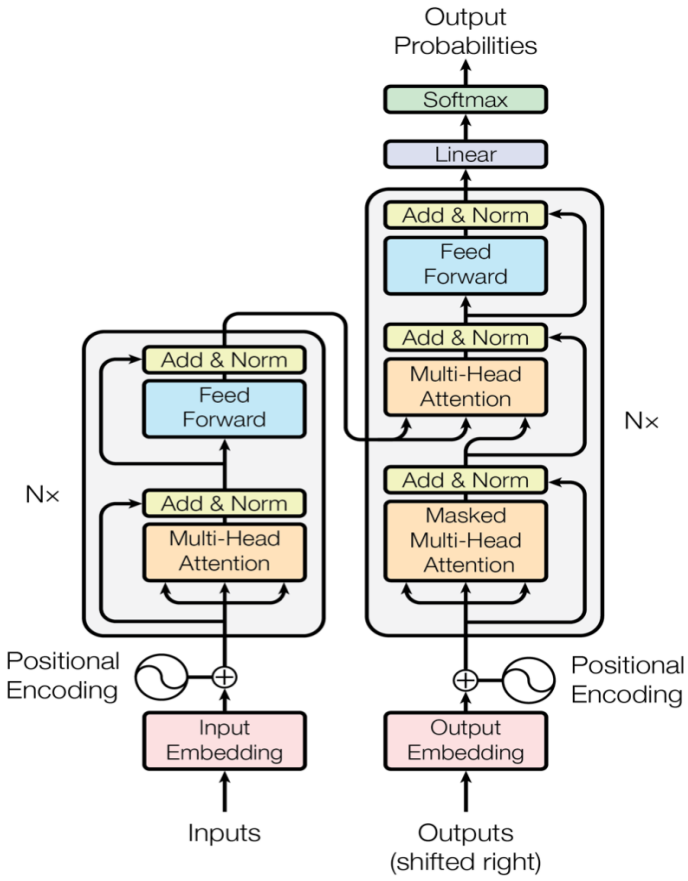


FIGURE 2.3: Transformer architecture that utilizes multi-head self-attention (Vaswani, 2017).

Euclidean Distance

Euclidean distance is one of the most commonly used metrics to find the proximity between two vectors. The equation 2.26 defines the Euclidean Distance between the two vectors $\mathbf{x}$ and $\mathbf{y}$, where $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$.

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{2.26}$$

Manhattan Distance

Manhattan distance represents the distance between two objects as a sum of the absolute differences of their Cartesian coordinates. The Manhattan Distance between

two vectors $\mathbf{x}$ and $\mathbf{y}$ (where $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$) is defined as per equation 2.27.

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} |x_i - y_i| \tag{2.27}$$

Minkowski Distance

Minkowski distance is a generalised form of Euclidean and Manhattan distances. Here, the distance between the two vectors $\mathbf{x}$ and $\mathbf{y}$ (where $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$) is defined by the equation 2.28. The metric is normally used with $p \in [1, 2]$, which corresponds to Manhattan distance when $p = 1$ and Euclidean distance with $p = 2$.

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}} \tag{2.28}$$

Cosine Similarity

Cosine Similarity computes an angle between two vectors measuring how close they are. The Cosine Similarity between the two vectors $\mathbf{x}$ and $\mathbf{y}$ (where $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$) is defined by the equation 5.3.

$$similarity(\mathbf{x}, \mathbf{y}) = cosine(\theta) = \frac{\mathbf{x} \cdot \mathbf{y}}{|\mathbf{x}||\mathbf{y}|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}} \tag{2.29}$$

2.5.4 Modelling and Evaluation Metrics

After the text is represented in the numeric form using the methods described above or similar techniques, the data can be used to train models for the specific tasks. Most commonly used model for various NLP classification tasks are RNN, LSTM, CNN, Attention Neural Networks, Transformer Neural Networks, etc. The classification models can then be evaluated using the metrics described below.

Precision

Classification Precision is defined by the equation 2.30. It describes the ability of a classifier not to assign the positive label to the negative sample. The best value of precision is 1 and the worst - 0.

$$Precision = \frac{T_p}{T_p + F_p} \tag{2.30}$$

Recall

Classification Recall is an evaluation metric defined by the ratio 2.31, it describes the ability of a classifier to find all the positive samples. The best value of recall is 1 and the worst - 0.

$$Recall = \frac{T_p}{T_p + F_n} \tag{2.31}$$

F1-Score

F1-score is a weighted average of Precision and Recall, where 1 is the best value for F1-score and 0 - the worst. It is defined by equation 2.32.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.32}$$

2.6 Conclusion

To summarize, after reviewing the literature, we identified the key gaps that currently exist. First, we found that there is a lack of research in domain-specific alternative data analysis that focuses on building industry-specific (i.e. fashion) models for investment signal extraction. Second, unlike other social media platforms such as Twitter, Facebook, Stocktwits, Instagram has not been widely explored for financial modelling, signal generation in academic literature. Third, due to the lack of annotated news corpus (Han, 2018), not much research is done in domain-specific news events detection problem with a focus on specific financial industries. With this research work, we try to fill in the research gaps mentioned above.

Chapter 3

Research Datasets

This chapter provides a detailed overview of the datasets used for the research work. Here, we describe the process of how the data is collected and stored. We outline the data storage software used and scraping tools built to extract the data.

3.1 Introduction

The key objective of this research work is to build new frameworks for generating signals from alternative data sources that could be used for making investment decisions in the Apparel and Footwear industry. The aim is to create novel models that are designed to analyse noisy online content (i.e. news or social media) and extract useful information that might have an impact on a company's financial performance. Traditionally, investors use financial metrics and fundamental analysis to estimate the value of a company. Although, the traditional analysis does not always provide a correct estimate of a true company's value. Therefore, increasingly more and more investors are starting to use alternative sources of information to help evaluate the companies. This thesis is exploring the use of alternative data sources in generating signals that may help to better infer the performance of the U.S. and European publicly traded fashion companies. Unlike any other industry, Apparel and Footwear has a huge online presence with a wide range of data points available. For example, fashion blogs, news, online shops, product reviews, consumer chatter and engagement information on social media platforms, etc. The types of datasets used for this research work include financial, social media (Instagram) and news. Below we describe the datasets in detail together with an overview of how they are obtained.

3.2 Fashion Equities

For this research work in total, we use 50 Apparel and Footwear assets traded in the U.S. and Europe. The examples of the fashion companies used for the research include *Nike*, *Adidas*, *Burberry*, *Michael Kors*, *Ralph Lauren*, *Hermes*, etc. For each experiment, we use a slightly different set of companies. During the first experiment, where we analyse the changes in a company's Instagram profile followers over the

time, we select only 11 fashion companies for the study where a company operates under one main brand (e.g. *Burberry*, *Moncler*, *Prada*, *Under Armour*) and has a single Instagram profile page associated with that company. For this experiment, companies with multiple brands were excluded, as it would be difficult to assess the contribution of each brand to the overall financial performance of their parent company. Meanwhile, for the second and third experiments, we select the assets based on data availability. For the second experiment, news articles about 40 U.S. and European assets are used and for the last experiment, 42 U.S.-only assets are used. In addition to equity prices, for the third experiment, we also use the U.S. Retail Trade and U.S. Consumer Non-Durables indices data. The datasets for the second and third experiments are provided by the industry partner Arabesque Asset Management.

3.3 Datasets

The datasets that are used during this research are subdivided into 3 main categories. Below is a description of each dataset.

3.3.1 Financial Data

The financial datasets used during this work together with the data sources are listed below.

- Stock prices of 50 U.S. and European Apparel and Footwear equities for the period between 2014 and 2019. The prices are obtained from the Bloomberg Terminal and also supplied by the research project partner Arabesque Asset Management.
- Sectors indices of the U.S. Retail Trade and U.S. Consumer Non-Durables sectors for the period between 2014 and 2019. The dataset is supplied by Arabesque Asset Management.
- Form 10-K reports (SEC filling reports) containing financial information about the U.S. companies, their performance. There are 2,947 documents in total. The documents are obtained by scraping *EDGAR*[1] database.

3.3.2 Social Media Data: Instagram

Instagram[2] is currently one of the most popular social media platforms (MediaKix, 2019). Naturally, it is one of the most popular marketing channels for fashion and

[1]EDGAR - the Electronic Data Gathering, Analysis, and Retrieval system, is a database containing documents submitted by the U.S. companies as required by U.S. Securities and Exchange Commission - SEC (*https://www.sec.gov/edgar*).

[2]Instagram - a social media platform (*https://www.instagram.com*).

consumer goods brands. Therefore, Instagram's data is chosen for this study. We obtain daily Instagram profile followers counts for 11 fashion companies. The followers' data is scraped daily directly from the Instagram and also to obtain the historic missing data we scrape SocialBlade[3] website.

3.3.3 News Data

Another alternative data source selected for the research is news. News articles contain a lot of useful information about the companies or events that might impact them. In total, we collect 0.5 mln news articles that mention selected 50 U.S. and European equities operating in the Apparel and Footwear industry. The news sources are listed below.

- EventRegistry - the majority of the articles are obtained from the EventRegistry [4] (supplied by the project partner Arabesque Asset Management).
- Scraping - in order to complement the EventRegistry dataset, we also scrape 9 influential fashion online news websites, magazines and blogs (e.g. *Vogue*, *Dazed*, etc.). We select 9 fashion websites based on their popularity, monthly readership, variety of topics they cover and general intuition. The full list of scrapers is visualised in Figure 3.1, where the yellow colour denotes the data scrapers built during this research to extract the data.

The collected articles contain a variety of different news categories including business, entertainment, political, financial and general consumer news.

3.4 Data Scrapers and Storage

In order to obtain some datasets, we build 13 scraping tools. The tools are built using Python libraries- *Scrapy* and *Beautiful Soup*. The Figure 3.1 shows a visualization of all datasets used for the research (yellow colour denotes the data scrapers built for this study e.g. *https://www.instagram.com*, *https://www.social- blade.com*, *https://www.hypebeast.com*, *https://www.businessoffashion.com*, etc.). The scrapers for news websites and blogs are used once to scrape entire websites, meanwhile, the Instagram scraper is used daily to extract the daily count of followers.

Most of the data used for this research work comes in the text format, e.g. news articles, Form 10-K reports. For the data storage, we use *MongoDB* database to store the documents in JSON format. Although, the key disadvantage of the system is that it is slow and is not optimized for the document search. Therefore, in addition to the *MongoDB*, we also use *ElasticSearch*- a distributed search engine that indexes

[3]SocialBlade - an online platform that contains historic followers data and information about various user accounts on the platforms such as Instagram, Twitter, YouTube and others (*https://www.socialblade.com*).

[4]EventRegistry is a news aggregator service (*https://www.eventregistry.org*).

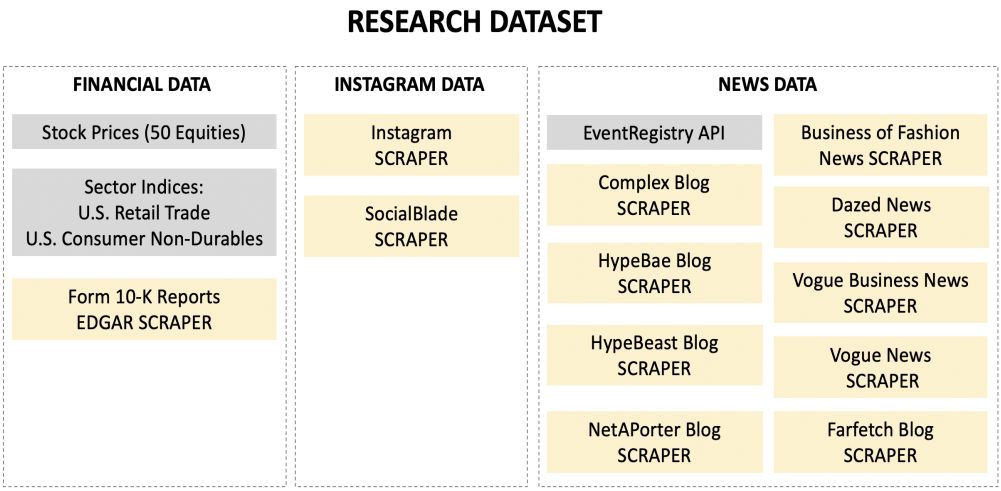


FIGURE 3.1: Types of data used during this research work. In yellow - the data obtained by building scrapers to crawl the websites and extract the relevant information; grey - data obtained from the Bloomberg Terminal or supplied by Arabesque Asset Management.

the documents and provides a fast way to search structured and unstructured text, numeric data within the documents. We use *ElasticSearch* during the experiment 3 (Chapter 6) as a tool to search for specific synonyms within the news articles.

3.5 Conclusion

To summarize, for this research work we use 3 different types of datasets including financial, social media and news data. Some of the financial data and news datasets are supplied by Arabesque Asset Management. The remaining financial data is obtained from the Bloomberg Terminal. In addition, we build 12 scraping tools to scrape the remaining missing datasets such as Form 10-K reports, Instagram followers and 9 online magazines and blogs. We store the data in MongoDB and use ElasticSearch for quick and optimized text search.

Chapter 4

Relationship between Instagram Popularity and Stock Prices

This chapter presents the first experiment which explores the relationship between the changes in a company's popularity on the social media platform Instagram and its stock price, revenue. Here, the popularity is defined as a company's daily total count of followers on its Instagram's profile. The experiment is subdivided into three parts: 1) analysis of the relationship between a company's popularity and its stock price; 2) analysis of the relationship between a company's popularity and its revenue; 3) design of two hypothetical trading strategies to measure whether the information about the followers could improve the returns.

4.1 Problem Overview

Over the last decade, there was an increase in research works that explored alternative data sources, such as social media or news, for investment decision making. Most commonly, the data from social media platforms such as Twitter, Facebook or StockTwits are used (Sohangir, 2018; Zhou, 2018). The studies analyzed consumer or investor sentiment towards companies. Other researchers explored a general public mood and its relationship towards the market movements. Meanwhile, there is a lack of studies that explore the relationship between the company's changes in popularity and its stock price or revenue movements. This is particularly important when analysing direct to consumer trend-driven companies, e.g. fashion brands. Moreover, the data from Instagram, one of the fastest-growing social media platforms, have not been widely used in financial research literature.

Instagram became the key marketing tool to drive sales. It is widely used by consumer companies especially Apparel and Footwear brands as one of the main digital marketing tools to engage with their target audiences and attract new customers. The platform is influencing fashion buying decisions. Therefore, during this experiment, it is assumed that consumer engagement with a company's profile on Instagram, could act as an early identifier of the future financial performance of a company. The research work (O'Connor, 2012) studied the relationship between a brand's popularity on social media platforms (Facebook, Twitter) and stock price.

The study found that a brand's popularity on social media reflects a public's interest in the brand and hence could be associated with a company's financial performance.

This experiment is an improvement of the previous works (O'Connor, 2012). The aim of the study is to understand whether there is a relationship between the changes in a company's popularity on Instagram and its stock price movements or revenues. A company's popularity is represented by its daily total followers count on its Instagram's profile. To our knowledge, there is no other research work that studies the popularity of an ephemeral goods company (e.g. apparel and footwear) on Instagram and its relationship with financial performance. The study is unique and differentiates from the previous works by the following:

- The experiment utilizes followers data on Instagram due to the platform importance and relevance at the time of this research work.
- In addition to the stock prices, we also analyse a company's revenue data.
- The data sample size used during this study covers a period of 3.5 years.

It is assumed that a brand's popularity on Instagram is a measure of its current relevance amongst the consumers. For example, a sudden increase in popularity could be an early identifier of consumers' interest in a brand or intent to purchase its goods. Therefore, during this experiment we assume that company's followers (i.e. perceived brand relevance) can have an impact on its financial performance, future revenues hence can be related to a company's share price changes. In order to test this assumption, the experiment is subdivided into three parts:

1. Popularity and stock prices: measuring a relationship between the changes in a company's followers on Instagram and its stock price movement after time t.
2. Popularity and revenues: analysis of the average change in a company's followers on Instagram and its revenue during a time period t.
3. Instagram trading strategies - two hypothetical trading strategies are designed that execute trades as per following: buy a company's stock if its Instagram followers acceleration decreases to a certain threshold and sell if the acceleration increases above a threshold, otherwise hold the stock. Both strategies are compared with a random strategy, where stocks are bought and sold randomly, to measure whether the information about the changes in followers could help to yield better returns.

This chapter is structured as follows: first, a dataset with its statistical properties is described, second, features are derived by decomposing and transforming the data and finally, we test all three hypotheses defined above.

4.2 Dataset

The dataset used for this experiment includes a set of publicly traded Apparel and Footwear equities and their associated daily Instagram follower counts, daily stock prices and reported revenues for a period of time. In total 11 public companies are selected that mostly trade under one brand name and have a single Instagram profile page associated with a company. For example, $Burberry\ Group\ PLC$ (ticker $LON : BRBY$) has one main consumer-facing brand $Burberry$, whilst $LVMH\ Moet\ Hennessy\ Louis\ Vuitton\ SE$ is a holding company for more than 15 brands (such as $Louis\ Vuitton$, $Dior$, $Fendi$, $Givenchy$, $Kenzo$, etc.) that all have separate Instagram profiles. The companies with multiple brands were excluded from this analysis as it would be difficult to assess the contribution of each brand to the overall financial performance of their holding company. The full list of equities selected for this experiment is shown in Table 4.1.

TABLE 4.1: The list of all stocks used during the experiment 1.

Company Name	Stock Ticker
$Brunello\ Cucinelli\ SpA$	$BIT : BC$
$Burberry\ Group\ PLC$	$LON : BRBY$
$Hermes\ International$	$EPA : RMS$
$Hugo\ Boss\ AG$	$ETR : BOSS$
$Michael\ Kors\ Holdings\ Ltd$	$NYSE : KORS$
$Moncler\ SpA$	$BIT : MONC$
$Mulberry\ Group\ PLC$	$LON : MUL$
$Prada\ SpA$	$1913 : HK$
$Ralph\ Lauren\ Corp$	$NYSE : RL$
$Salvatore\ Ferragamo\ Italia\ SpA$	$BIT : SFER$
$Under\ Armour\ Inc$	$NYSE : UAA$

The data used for this experiment consists of 3 datasets: 1) daily Instagram profile followers counts; 2) daily stock prices; 3) quarterly or semi-annual revenues of the selected companies for a period of 3.5 years on average.

The daily followers' data is collected by scraping the Instagram platform and historic followers data is obtained by scraping SocialBlade[1] website. The periods for which the historic followers' data are available varies slightly based on the brand. The Table 4.2 provides a summary of the total number of data points extracted for each company together with the time periods. In total there are 13,907 data points of followers counts across all companies used for the experiment.

The historic daily stock prices data (closing price) for each company is obtained from the Bloomberg Terminal. The stock prices for the weekends and bank holidays are populated with the price from the previous day in order to match the size of the followers' dataset.

[1]SocialBlade - an online platform that contains historic followers data and information about various user accounts on the platforms such as Instagram, Twitter, YouTube and others (*https://www.socialblade.com*).

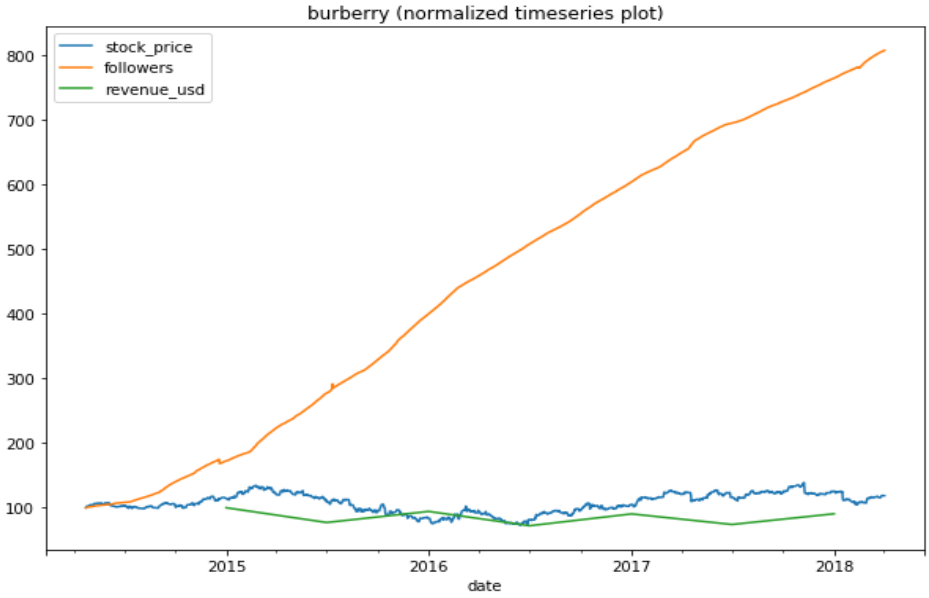


FIGURE 4.1: Burberry: normalized Instagram followers, stock prices and revenues during 2014-2018.

The revenue data for each asset is also obtained from the Bloomberg Terminal. The revenue reporting periods vary based on the company. The Table 4.3 gives an overview of the revenue data collected for each asset, time periods together with the reporting frequency. The total number of revenue data points across all companies is 134. The dataset also includes the revenues reported beyond the periods of which the followers' data is available in order to observe whether the changes in followers have an impact on future revenue growth.

TABLE 4.2: Instagram follower counts for each company.

Company Name	Period	Total Data Points
Brunello Cucinelli SpA	2015-06-16 - 2018-03-28	1017
Burberry Group PLC	2014-04-23 - 2018-04-02	1441
Hermes International	2014-07-04 - 2018-04-02	1369
Hugo Boss AG	2014-09-09 - 2018-04-02	1302
Michael Kors Holdings Ltd	2014-04-30 - 2018-04-02	1434
Moncler SpA	2015-05-05 - 2018-04-02	1064
Mulberry Group PLC	2015-04-14 - 2018-03-28	1080
Prada SpA	2014-07-26 - 2018-04-02	1347
Ralph Lauren Corp	2014-08-01 - 2018-04-02	1341
Salvatore Ferragamo Italia SpA	2015-02-11 - 2018-04-02	1147
Under Armour Inc	2014-07-08 - 2018-04-02	1365

In order to better visualise the trends in the datasets, the Figures 4.1 and 4.2 show normalized time series plots of all three data types (followers, stock prices and revenues) for *Burberry* and *UnderArmour* respectively. Refer to Appendix A for the associated plots of the remaining 9 companies.

TABLE 4.3: Revenue data overview for each company.

Company Name	Reporting Frequency	Period	Total Data Points
Brunello Cucinelli SpA	Quarterly	2015-09-30 - 2018-06-30	12
Burberry Group PLC	Semi-Annual	2015-03-31 - 2018-03-31	7
Hermes International	Quarterly	2014-09-30 - 2018-06-30	16
Hugo Boss AG	Quarterly	2014-12-31 - 2018-06-30	15
Michael Kors Holdings Ltd	Quarterly	2014-09-27 - 2018-06-30	16
Moncler SpA	Quarterly	2015-09-30 - 2018-09-30	13
Mulberry Group PLC	Semi-Annual	2016-03-31 - 2018-03-31	5
Prada SpA	Semi-Annual	2015-01-31 - 2018-06-30	6
Ralph Lauren Corp	Quarterly	2014-12-27 - 2018-09-29	16
Salvatore Ferragamo Italia SpA	Quarterly	2015-06-30 - 2018-06-30	13
Under Armour Inc	Quarterly	2014-12-31 - 2018-09-30	16

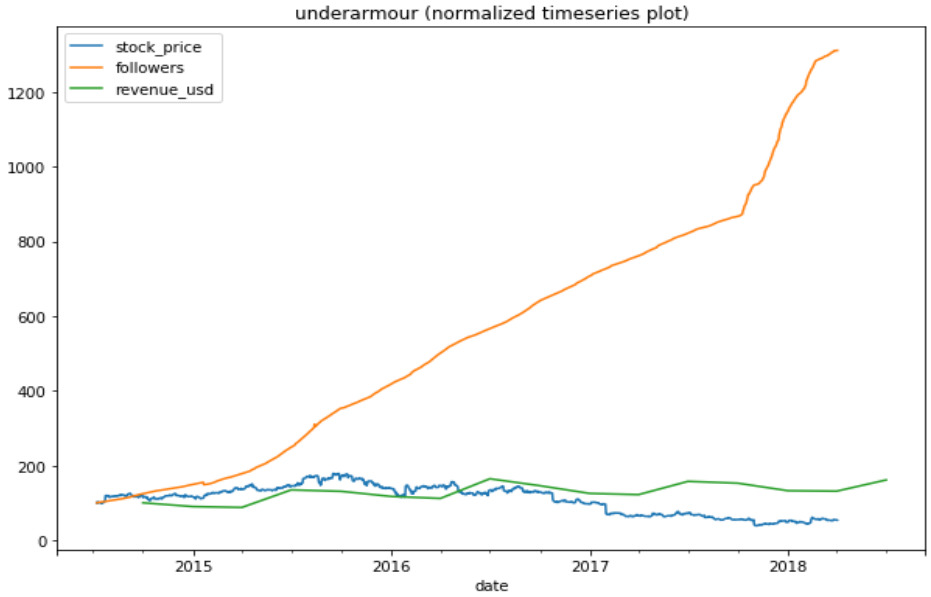


FIGURE 4.2: Under Armour: normalized Instagram followers, stock prices and revenues during 2014-2018.

4.3 Feature Engineering

In order to prepare the data for the experiment, it is decomposed and transformed in order to eliminate the seasonality and trend effects. The new features are created that are used for further analysis. Below we list all equations that are used to transform the data (daily followers count f, daily stock prices p and revenues r, where x'_t denotes a value of f, p or r (or their transforms) at the time step t - days):

1. Logarithmic transformation of f, p and r datasets is defined by the equation 4.1.

$$x'_t = \ln x_t \tag{4.1}$$

2. The first and second order differences (equations 4.2 and 4.3 respectively).

$$\Delta x'_t = x_t - x_{t-1} \tag{4.2}$$

$$\Delta x''_t = \Delta x'_t - \Delta x'_{t-1} \tag{4.3}$$

3. Relative difference x_{rt} (equation 4.4).

$$x_{rt} = \frac{x_t - x_{t-1}}{x_{t-1}} \tag{4.4}$$

4. Division between the two time steps (equation 4.5).

$$x_{dt} = \frac{x_t}{x_{t-1}} \tag{4.5}$$

5. Asset price return (equation 4.6) and logarithmic transform of a return (equation 4.7).

$$R_t = \frac{p_t}{p_{t-1}} \tag{4.6}$$

$$lnR_t = \ln \frac{p_t}{p_{t-1}} \tag{4.7}$$

6. Velocity - the rate of change (equation 4.8), where t_{period} - a time period over which the change occurred.

$$v = \frac{\Delta x'_t}{t_{period}} \tag{4.8}$$

7. Acceleration (equation 4.9):

$$a = \frac{\Delta x''_t}{t_{period}} \tag{4.9}$$

The full list of features derived from the original datasets are summarized in the Table 4.4. In addition to the features listed in the Table 4.4, binary values for all the features (equations 4.1 - 4.9) are derived as per condition in the equation 4.10 and appended to the final dataset.

$$x_{binary} = \begin{cases} 1, \ if \ x \ > 0 \\ 0, \ if \ x \ \leq 0 \end{cases} \tag{4.10}$$

Standardization of the features is also implemented using the equation 4.11. Here, $mean(x)$ is a mean of the values x and $std(x)$ is a standard deviation of values x.

$$x_{t\ scaled} = \frac{x_t - mean(x)}{std(x)} \tag{4.11}$$

TABLE 4.4: Features created using the transformation and decomposition equations 4.1 - 4.9.

Feature Name	Input Data	Equation Used
returns	price	4.6
log_returns	price	4.7
log_PX_LAST	price	4.1
df_PX_LAST	price	4.2
df.df_PX_LAST	df_price	4.2
df.log_PX_LAST	log_price	4.2
rdf_PX_LAST	price	4.4
dv_PX_LAST	price	4.5
df.dv_PX_LAST	dv_price	4.2
df.df.dv_PX_LAST	df.dv_price	4.2
log.dv_PX_LAST	dv_price	4.1
v_PX_LAST	df_PX_LAST, t_{period} = 1 day	4.8
a_PX_LAST	df.df_PX_LAST, t_{period} = 1 day	4.9
log_followers	followers	4.1
df_followers	followers	4.2
df.df_followers	df_followers	4.2
df.log_followers	log_followers	4.2
rdf_followers	followers	4.4
rdf.rdf_followers	rdf_followers	4.4
dv_followers	followers	4.5
df.dv_followers	dv_followers	4.2
df.df.dv_followers	df.dv_followers	4.2
log.dv_followers	dv_followers	4.1
v_followers	df_followers, t_{period} = 1 day	4.8
a_followers	df.df_followers, t_{period} = 1 day	4.9
log_revenue	revenue	4.1
df_revenue	revenue	4.2
df.df_revenue	df_revenue	4.2
df.log_revenue	log_revenue	4.2
rdf_revenue	revenue	4.4
dv_revenue	revenue	4.5
df.dv_revenue	dv_revenue	4.2
df.df.dv_revenue	df.dv_revenue	4.2
log.dv_revenue	dv_revenue	4.1

4.3.1 Features Overview

After creating the features, they are tested for stationarity and normality using statistical tests. Stationary of the features are tested using Augmented Dickey–Fuller (ADF) and Kwiatkowski–Phillips–Schmidt–Shin (KPSS) tests and data normality - Shapiro-Wilk (SW) test. We find that most of the features are stationary although not normally distributed. The tests results for selected features (followers acceleration and returns) are presented in the Tables 4.5, 4.6 and 4.7. The Figures 4.3-4.7 show the frequency distributions of selected features for *Burberry* that are used for the further analysis. The feature distributions plots of the remaining 10 companies

are shown in Appendix A.

TABLE 4.5: Selected features stationarity tests results: Augmented Dickey–Fuller (ADF) test ($\alpha = 0.05$).

Feature	ADF test statistic	p-value	Result
Burberry followers acceleration	-8.397	0.0	Stationary
Burberry returns	-15.04	0.0	Stationary
Hugo Boss followers acceleration	-11.18	0.0	Stationary
Hugo Boss returns	-14.46	0.0	Stationary
Brunello Cucinelli followers acceleration	-8.17	0.0	Stationary
Brunello Cucinelli returns	-3.92	0.0	Stationary
Ferragamo followers acceleration	-8.81	0.0	Stationary
Ferragamo returns	-9.83	0.0	Stationary
Hermes followers acceleration	-6.42	0.0	Stationary
Hermes returns	-3.73	0.0	Stationary
Michael Kors followers acceleration	-8.38	0.0	Stationary
Michael Kors returns	-16.50	0.0	Stationary
Moncler followers acceleration	-9.69	0.0	Stationary
Moncler returns	-12.44	0.0	Stationary
Mulberry followers acceleration	-7.81	0.0	Stationary
Mulberry returns	-8.64	0.0	Stationary
Ralph Lauren followers acceleration	-9.23	0.0	Stationary
Ralph Lauren returns	-13.86	0.0	Stationary
Under Armour followers acceleration	-9.23	0.0	Stationary
Under Armour returns	-13.86	0.0	Stationary

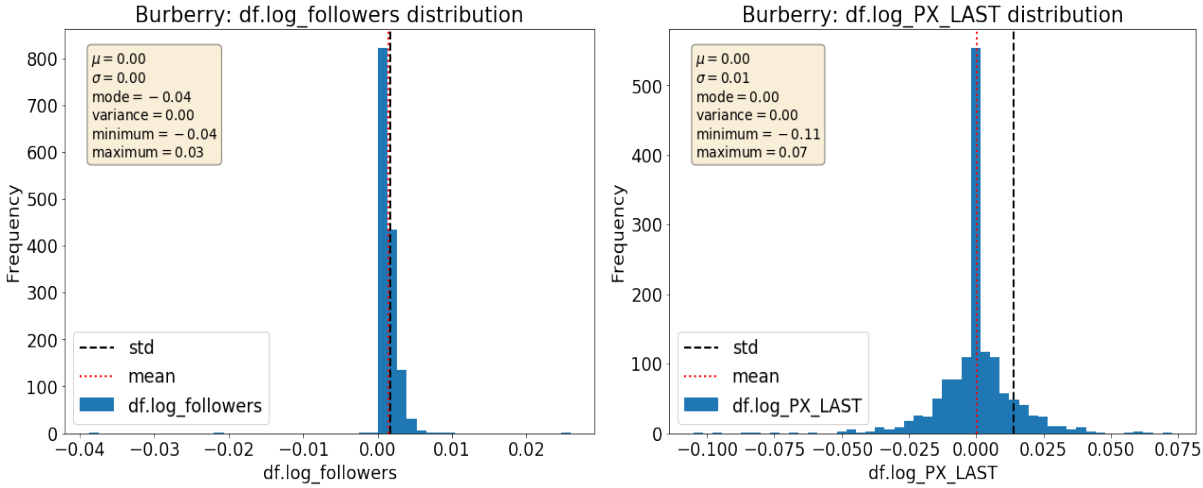
TABLE 4.6: Selected features stationarity tests results: Kwiatkowski–Phillips–Schmidt–Shin (KPSS) test ($\alpha = 0.05$).

Feature	KPSS test statistic	p-value	Result
Burberry followers acceleration	0.06	0.1	Stationary
Burberry returns	0.41	0.07	Stationary
Hugo Boss followers acceleration	0.10	0.1	Stationary
Hugo Boss returns	0.39	0.07	Stationary
Brunello Cucinelli followers acceleration	0.12	0.1	Stationary
Brunello Cucinelli returns	0.21	0.1	Stationary
Ferragamo followers acceleration	0.04	0.1	Stationary
Ferragamo returns	0.12	0.1	Stationary
Hermes followers acceleration	0.08	0.1	Stationary
Hermes returns	0.16	0.1	Stationary
Michael Kors followers acceleration	0.06	0.1	Stationary
Michael Kors returns	0.57	0.0	Not stationary
Moncler followers acceleration	0.09	0.1	Stationary
Moncler returns	0.38	0.08	Stationary
Mulberry followers acceleration	0.05	0.1	Stationary
Mulberry returns	0.04	0.1	Stationary
Ralph Lauren followers acceleration	0.11	0.1	Stationary
Ralph Lauren returns	0.41	0.07	Stationary
Under Armour followers acceleration	0.05	0.1	Stationary
Under Armour returns	0.2	0.1	Stationary

TABLE 4.7: Selected features normality tests results: Shapiro-Wilk (SW) test ($\alpha = 0.05$).

Feature	SW test statistic	p-value	Result
Burberry followers acceleration	0.25	0.0	Not normal
Burberry returns	0.95	0.0	Not normal
Hugo Boss followers acceleration	0.53	0.0	Not normal
Hugo Boss returns	0.95	0.0	Not normal
Brunello Cucinelli followers acceleration	0.90	0.0	Not normal
Brunello Cucinelli returns	0.94	0.0	Not normal
Ferragamo followers acceleration	0.85	0.0	Not normal
Ferragamo returns	0.98	0.08	Normal
Hermes followers acceleration	0.81	0.0	Not normal
Hermes returns	0.94	0.0	Not normal
Michael Kors followers acceleration	0.27	0.0	Not normal
Michael Kors returns	0.88	0.0	Not normal
Moncler followers acceleration	0.67	0.0	Not normal
Moncler returns	0.97	0.0	Not normal
Mulberry followers acceleration	0.25	0.0	Not normal
Mulberry returns	0.85	0.0	Not normal
Ralph Lauren followers acceleration	0.59	0.0	Not normal
Ralph Lauren returns	0.98	0.0	Not normal
Under Armour followers acceleration	0.48	0.0	Not normal
Under Armour returns	0.97	0.0	Not normal

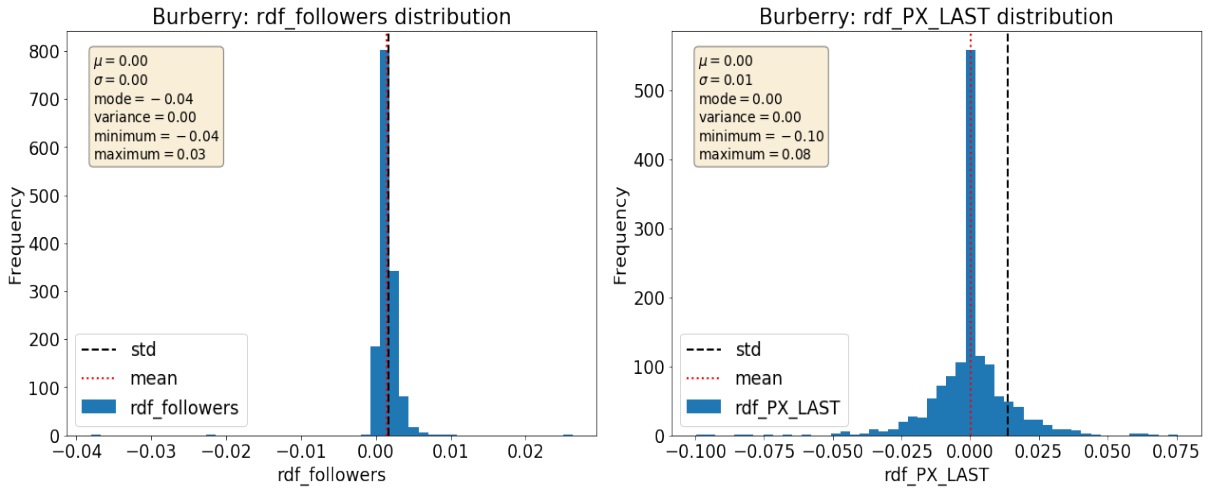
- The Figure 4.3 shows the frequency distributions of a change in logarithmic transforms of followers (A) and stock prices (B) for *Burberry*.



(A) (B)

FIGURE 4.3: Burberry: shows the frequency distributions of a change in logarithmic transforms of followers (A) and stock prices (B) (2014-2018).

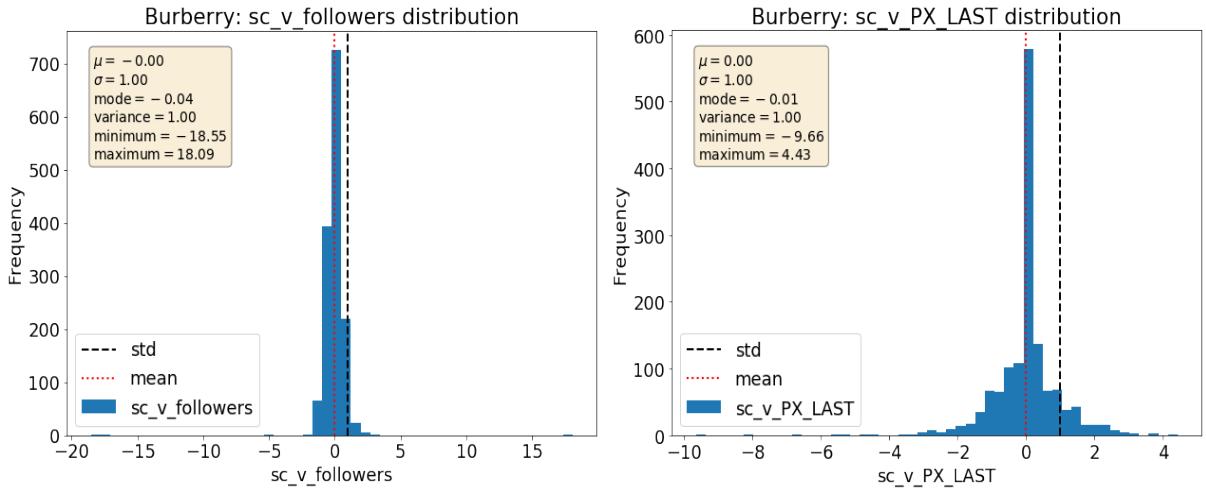
- The Figure 4.4 shows the frequency distributions of a relative change in followers (A) and stock prices (B) for *Burberry*.



(A) (B)

FIGURE 4.4: Burberry: shows the frequency distributions of a relative change in followers (A) and stock prices (B) (2014-2018).

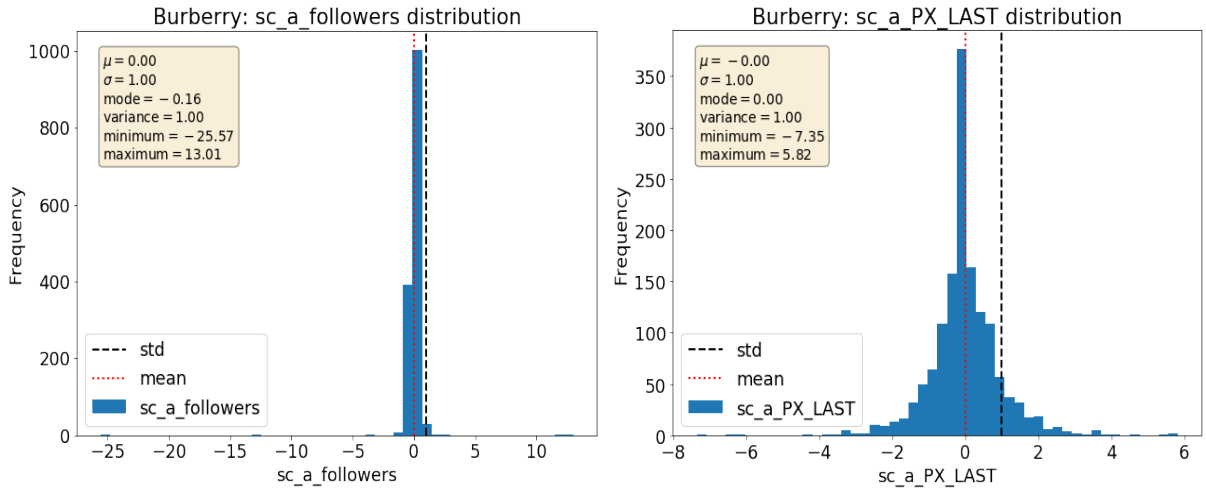
- The Figure 4.5 shows the frequency distributions of followers velocity (A) and stock prices velocity (B) for *Burberry*.



(A) (B)

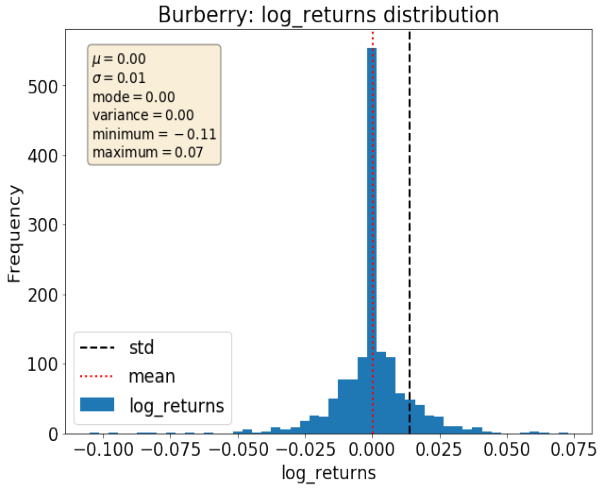
FIGURE 4.5: Burberry: shows the frequency distributions of followers velocity (A) and stock prices velocity (B) (2014-2018).

- The Figure 4.6 shows the frequency distributions of followers acceleration (A) and stock prices acceleration (B) for *Burberry*.
- The Figure 4.7 shows the frequency distributions of the logarithmic transform of the returns for *Burberry*.



(A) (B)

FIGURE 4.6: Burberry: shows the frequency distributions of followers acceleration (A) and stock prices acceleration (B) (2014-2018).



(A)

FIGURE 4.7: Burberry: shows the frequency distributions of the logarithmic transform of the returns (2014-2018).

4.4 Popularity and Stock Prices

The first part of the experiment aims to test whether there is any relationship between the changes in a company's Instagram followers counts and its stock price movements. Specifically, the aim is to observe if the following scenario happens: when there is a significant change in a company's followers counts on the Instagram, after time t a change happens in its stock price. As an example, the Figures 4.8 and 4.9 show the two time series of the changes in followers and stock prices for $Burberry$ and $UnderArmour$, the goal is to quantify the link between these two time series.

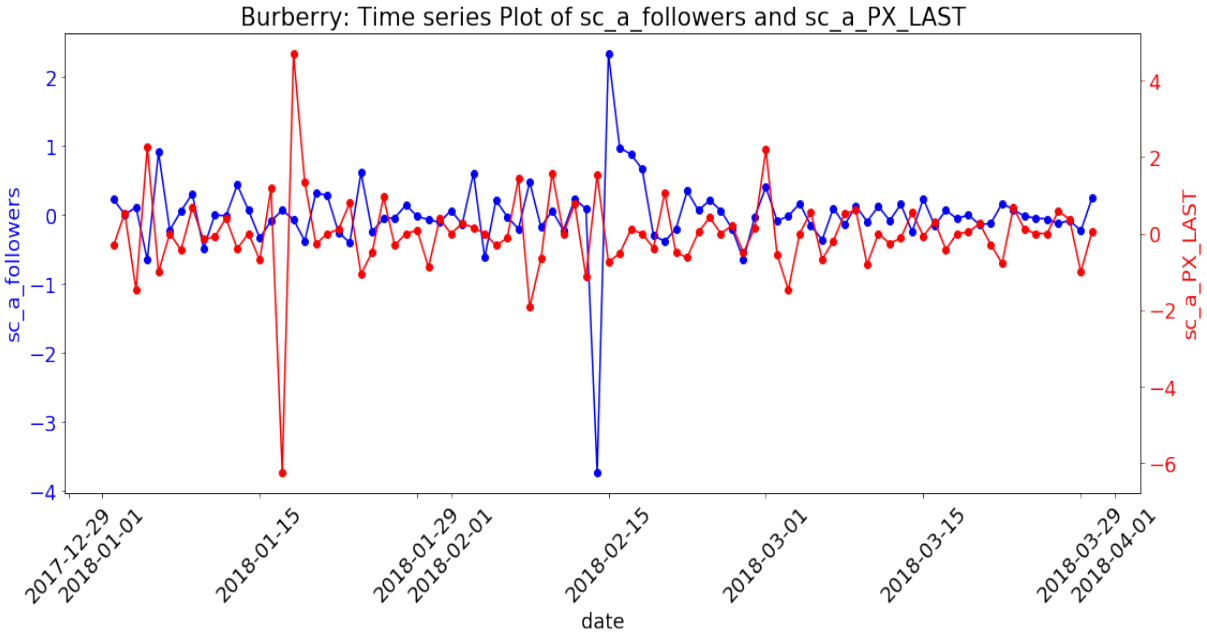


FIGURE 4.8: Burberry: acceleration the of stock prices and followers (January - March 2018).

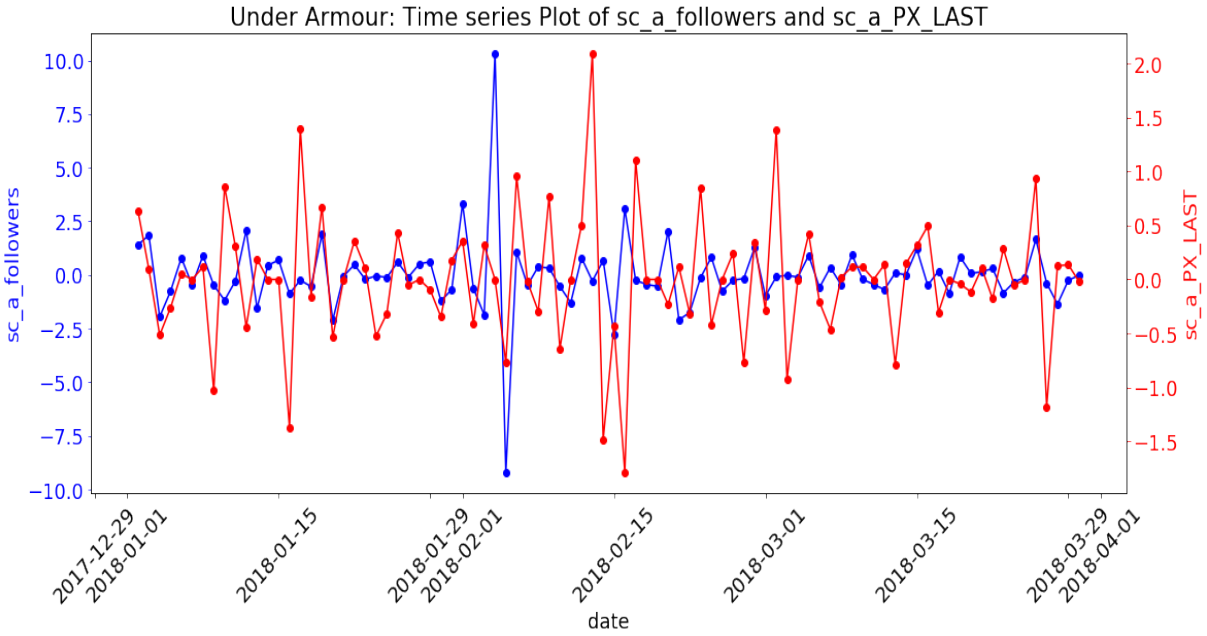


FIGURE 4.9: Under Armour: acceleration of the stock prices and followers (January - March 2018).

4.4.1 Experiment Design and Implementation

In order to analyze the relationship between the followers and stock prices, a cross-correlation analysis is implemented with the time lag of 0 to +/- 25 days. The full list of features used for this analysis is listed in the Table 4.8. The cross-correlation analysis is performed between the features with continuous values and corresponding binary values.

TABLE 4.8: Features used for analysing the relationship between the changes in a company's popularity and its stock price.

Feature Name	Description
df.log_followers	change in logarithmic transform of followers count
rdf_followers	relative change of followers count
dv_followers	division of followers count between two time steps
sc_v_followers	followers velocity (scaled)
sc_a_followers	followers acceleration (scaled)
log_returns	logarithmic transform of returns
df.log_PX_LAST	change in logarithmic transform of stock price
rdf_PX_LAST	relative change of stock price
dv_PX_LAST	division of stock prices between two time steps
sc_v_PX_LAST	stock price velocity (scaled)
sc_a_PX_LAST	stock price acceleration (scaled)

For the continuous values, cross-correlations between the two time series (followers and stock prices) with the time lags l (in days) ($l \in [-25, 25]$) are calculated using Pearson correlation coefficient r as per equation 4.12, where X and Y is a pair of sample features in the two different time series and N - total number of pairs. The significance level sl for the correlation coefficient is defined by the equation 4.13.

$$r = \frac{\sum XY - \frac{(\sum X)(\sum Y)}{N}}{\sqrt{(\sum X^2 - \frac{(\sum X)^2)}{N})(\sum Y^2 - \frac{(\sum Y)^2)}{N})}} \tag{4.12}$$

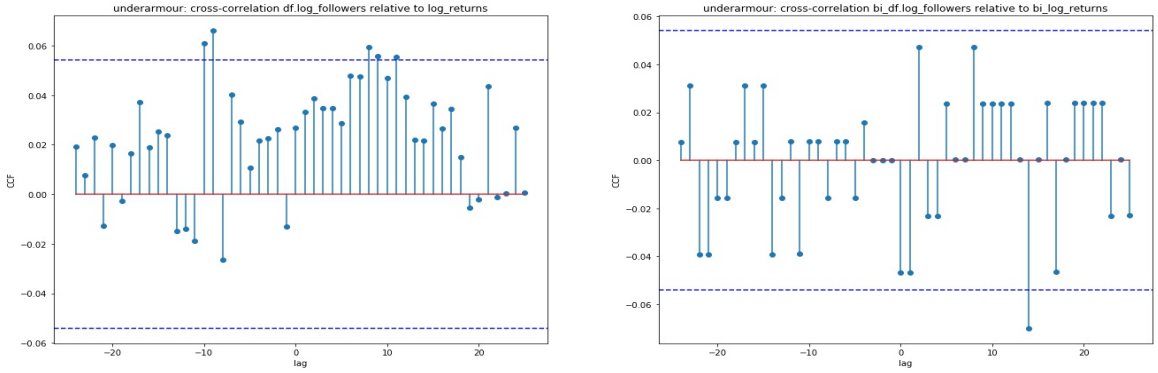
$$sl = \frac{2}{\sqrt{N}} \tag{4.13}$$

For the binary values, cross-correlations are estimated using Matthews correlation coefficient (MCC) which is defined by the equation 4.14, where TP is true positives, TN - true negatives, FP - false positives, FN - false negatives.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP \times FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{4.14}$$

Hypotheses The following hypotheses are defined and the cross-correlations analysed by plotting the CCF plots for each hypothesis across all 11 companies.

1. The changes in a logarithmic transformation of a company's followers count ($df.log\_followers$) is positively correlated with the changes in its logarithmic

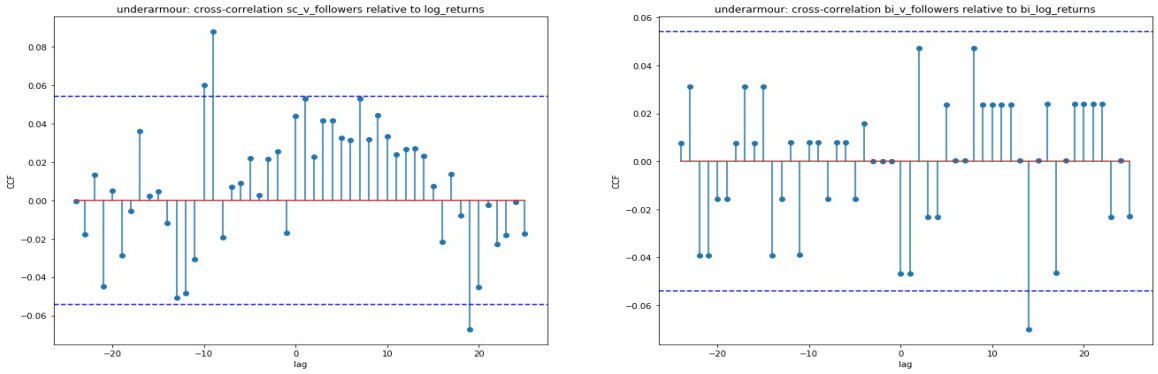


(A) (B)

FIGURE 4.10: Under Armour: cross-correlation of a change in logarithmic transform of followers ($df.log\_followers$) relative to a logarithm of returns ($log\_returns$) (A) and the corresponding binary version (B).

transformation of the returns ($log\_returns$) with a negative lag l. The Figure 4.10 shows the corresponding CCF plots for $Under\ Armour$.

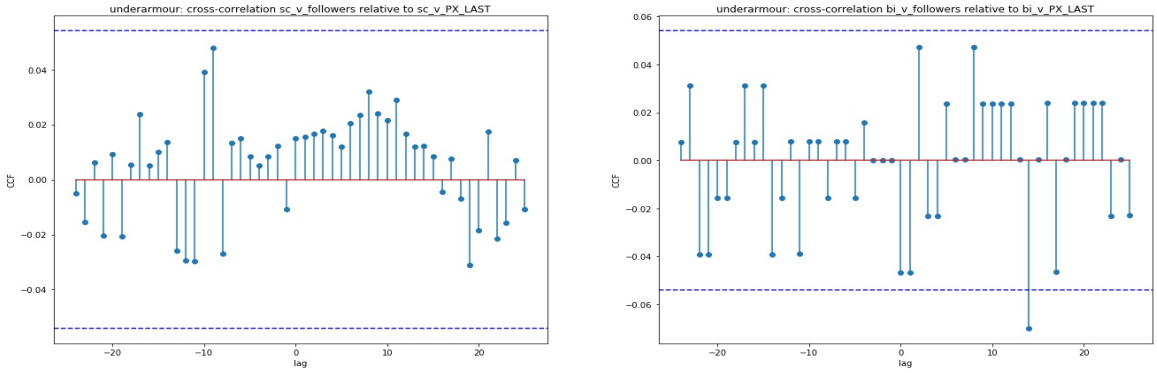
2. The changes in a company's followers velocity ($sc\_v\_followers$) is positively correlated with the changes in its logarithmic transformation of the returns ($log\_returns$) with a negative lag l. The Figure 4.11 shows the corresponding CCF plots for $Under\ Armour$.



(A) (B)

FIGURE 4.11: Under Armour: cross-correlation of a change in followers velocity ($sc\_v\_followers$) relative to a logarithm of returns ($log\_returns$) (A) and the corresponding binary version (B).

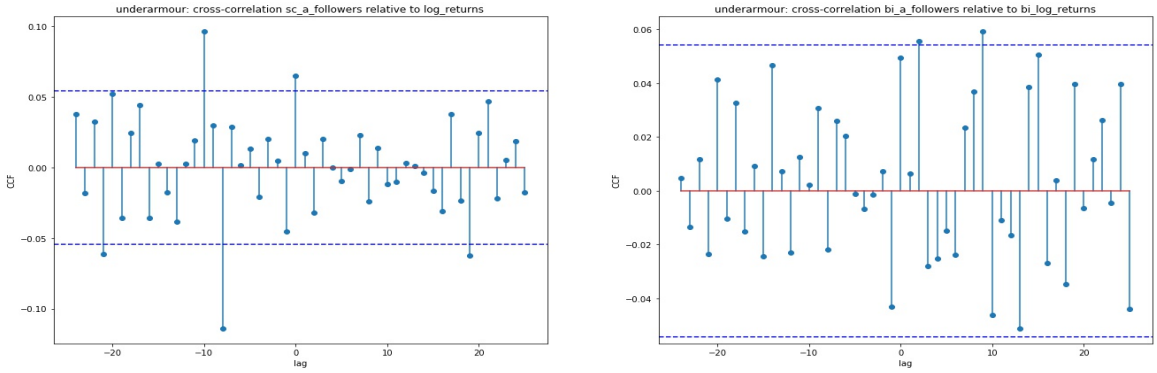
3. The changes in a company's followers velocity ($sc\_v\_followers$) is positively correlated with the changes in its stock price velocity ($sc\_v\_PX\_LAST$) with a negative lag l. The Figure 4.12 shows the CCF plots for $Under\ Armour$.



(A) (B)

FIGURE 4.12: Under Armour: cross-correlation of a change in followers velocity ($sc\_v\_followers$) relative to a velocity of stock prices ($sc\_v\_PX\_LAST$) (A) and the corresponding binary version (B).

4. The changes in a company's followers acceleration ($sc\_a\_followers$) is positively correlated with the changes in its logarithmic transformation of the returns ($log\_returns$) with a negative lag l. The Figure 4.13 shows the corresponding CCF plots for $Under\ Armour$.

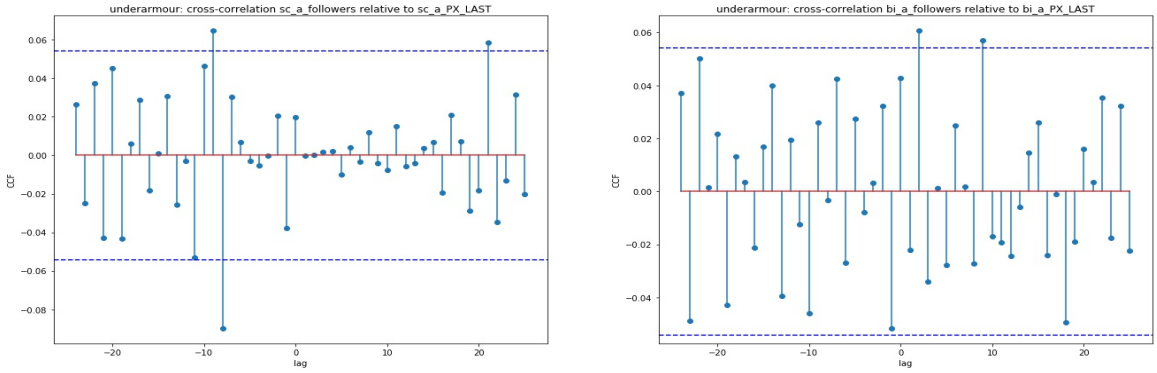


(A) (B)

FIGURE 4.13: Under Armour: cross-correlation of a change in followers acceleration ($sc\_a\_followers$) relative to a logarithm of returns ($log\_returns$) (A) and the corresponding binary version (B).

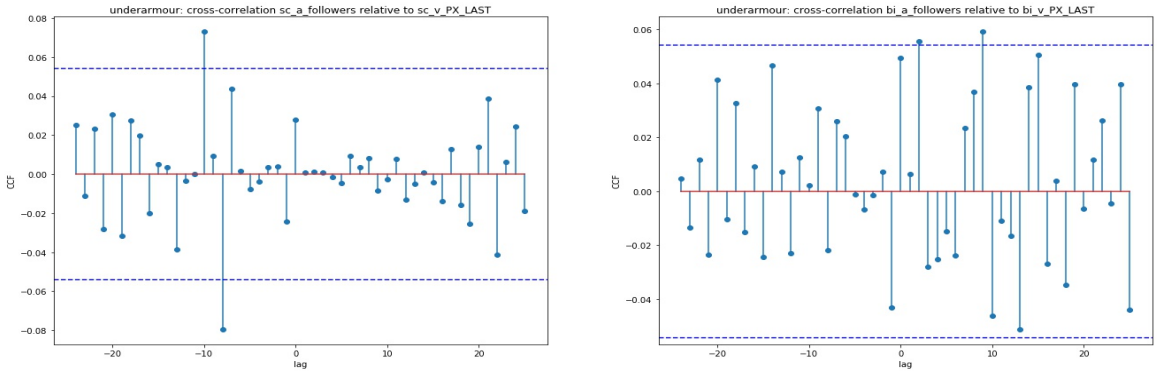
5. The changes in a company's followers acceleration ($sc\_a\_followers$) is positively correlated with the changes in its stock price acceleration ($sc\_a\_PX\_LAST$) with a negative lag l. The Figure 4.14 shows the CCF plots for $Under\ Armour$.

6. The changes in a company's followers acceleration ($sc\_a\_followers$) is positively correlated with the changes in its stock price velocity ($sc\_v\_PX\_LAST$) with a negative lag l. The Figure 4.15 shows the corresponding CCF plots for $Under\ Armour$.



(A) (B)

FIGURE 4.14: Under Armour: cross-correlation of a change in followers acceleration ($sc\_a\_followers$) relative to an acceleration of stock price ($sc\_a\_PX\_LAST$) (A) and the corresponding binary version (B).

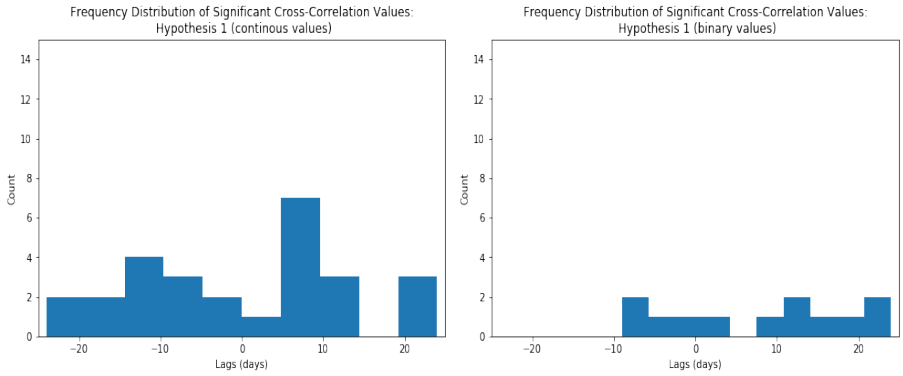


(A) (B)

FIGURE 4.15: Under Armour: cross-correlation of a change in followers acceleration ($sc\_a\_followers$) relative to a velocity of stock price ($sc\_v\_PX\_LAST$) (A) and the corresponding binary version (B).

We analyse the significant cross-correlations (values that are above the significance level, a blue dotted line, see the Figure 4.14 as an example) and their occurrence frequencies. As a benchmark, in total there are 561 cross-correlation values (across +/- 25 days and 11 companies, significant cross-correlations and non-significant cross-correlations). For each hypothesis, the frequencies of significant correlations and their lags are displayed in the Figures 4.16 - 4.21.

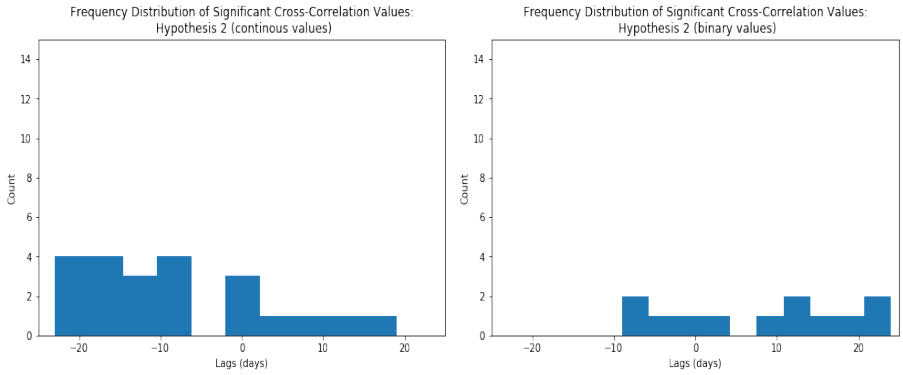
- The Figure 4.16 shows the frequencies of significant cross-correlation values across the time lags for Hypothesis 1 with continuous (A) and binary (B) values across all companies in the dataset.
- The Figure 4.17 shows the frequencies of significant cross-correlation values across the time lags for Hypothesis 2 with continuous (A) and binary (B) values



(A) (B)

FIGURE 4.16: The frequencies of significant cross-correlation values across the time lags for Hypothesis 1 with continuous (A) and binary (B) values across all companies in the dataset.

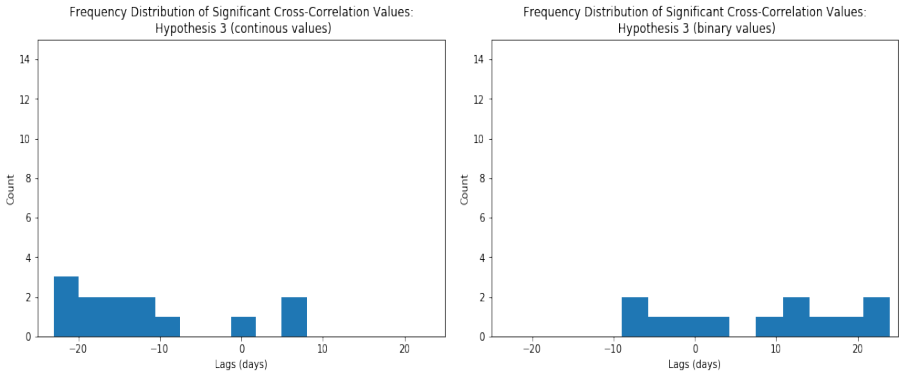
across all companies in the dataset.



(A) (B)

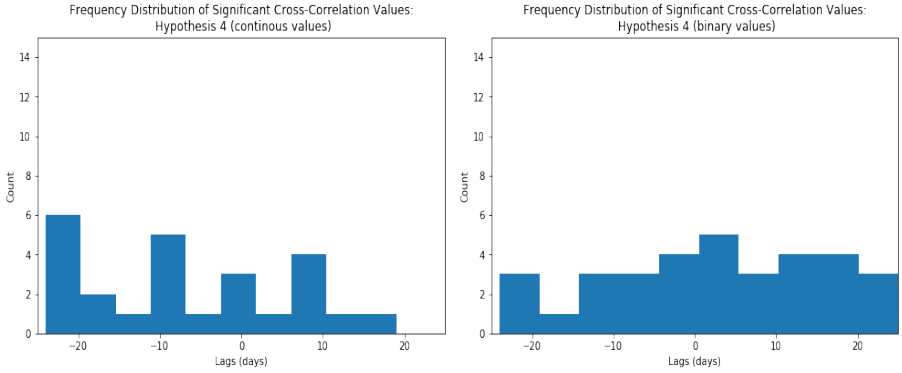
FIGURE 4.17: The frequencies of significant cross-correlation values across the time lags for Hypothesis 2 with continuous (A) and binary (B) values across all companies in the dataset.

- The Figure 4.18 shows the frequencies of significant cross-correlation values across the time lags for Hypothesis 3 with continuous (A) and binary (B) values across all companies in the dataset.
- The Figure 4.19 shows the frequencies of significant cross-correlation values across the time lags for Hypothesis 4 with continuous (A) and binary (B) values across all companies in the dataset.
- The Figure 4.20 shows the frequencies of significant cross-correlation values across the time lags for Hypothesis 5 with continuous (A) and binary (B) values across all companies in the dataset.



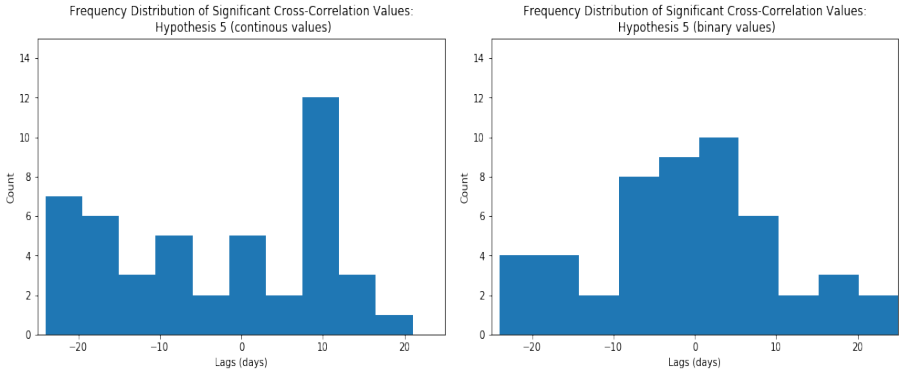
(A) (B)

FIGURE 4.18: The frequencies of significant cross-correlation values across the time lags for Hypothesis 3 with continuous (A) and binary (B) values across all companies in the dataset.



(A) (B)

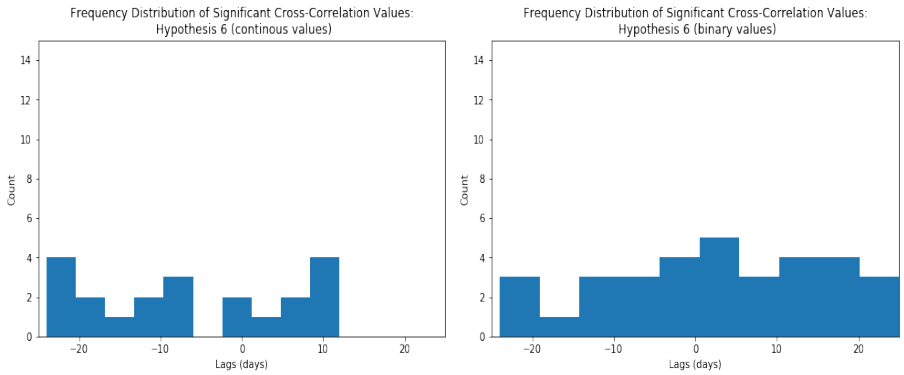
FIGURE 4.19: The frequencies of significant cross-correlation values across the time lags for Hypothesis 4 with continuous (A) and binary (B) values across all companies in the dataset.



(A) (B)

FIGURE 4.20: The frequencies of significant cross-correlation values across the time lags for Hypothesis 5 with continuous (A) and binary (B) values across all companies in the dataset.

- The Figure 4.21 shows the frequencies of significant cross-correlation values across the time lags for Hypothesis 6 with continuous (A) and binary (B) values across all companies in the dataset.



(A) (B)

FIGURE 4.21: The frequencies of significant cross-correlation values across the time lags for Hypothesis 6 with continuous (A) and binary (B) values across all companies in the dataset.

In order to compare all the hypothesis and the total significant cross-correlations in each hypothesis across all the companies, the frequencies of the total significant cross-correlations are plotted in the Figure 4.22.

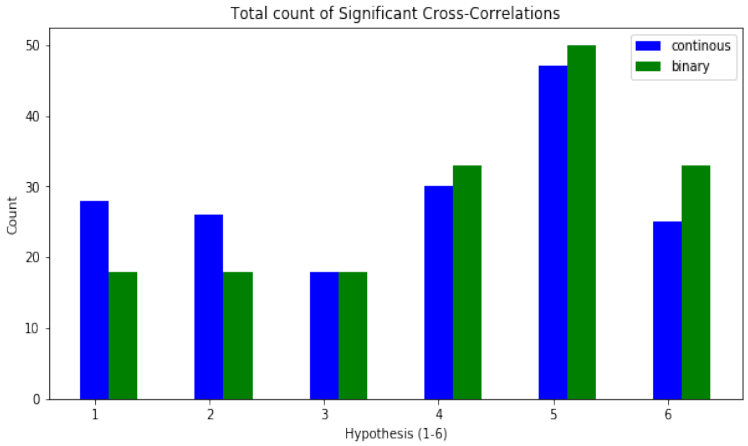


FIGURE 4.22: Comparison of the significant cross-correlation values in each Hypothesis 1-6.

4.4.2 Experiment Results

When considering CCF plots (cross-correlations) for each individual company, most of them do not display any significant trend, as both followers and stock prices datasets are extremely noisy. The cross-correlations coefficients are varying between the different time lags with no apparent pattern, for example, see the Figure 4.23 that displays $Hermes$ the cross-correlation analysis between the followers' acceleration and the acceleration of its stock price. Here, the significant cross-correlation

coefficients varies across the positive and negative time lags without any apparent trend.

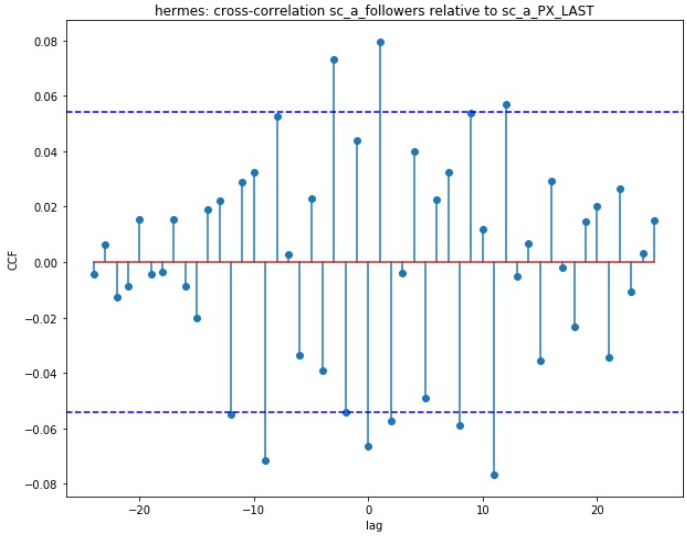
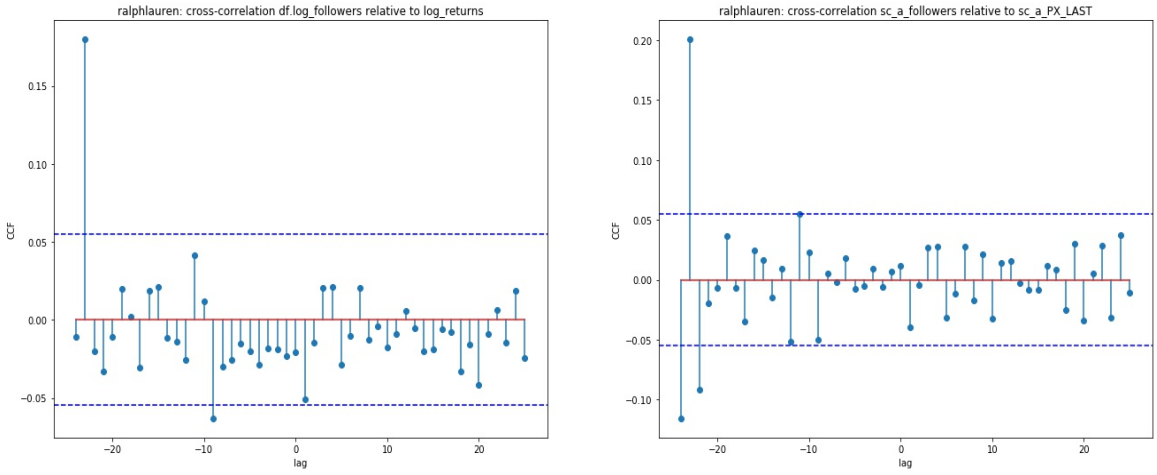


FIGURE 4.23: Hermes: cross-correlation between the acceleration of followers and the acceleration of stock prices where the blue line represents the significance level.

Although, a trend could be observed in the CCF plots for two US-based companies $Michael\ Kors$ and $Ralph\ Lauren$. For $Ralph\ Lauren$, the significant cross-correlation between the change in logarithmic transform of followers and returns is 0.18 at the time lag of -24 days (Figure 4.24 (A)) and the significant cross-correlations between the acceleration of followers and the acceleration of stock prices are 0.2 and -0.12 at the time lags of -24 days and -23 days respectively (Figure 4.24 (B)). This shows that potentially the changes in the followers and stock prices for $RalphLauren$ are correlated with a time lag of -23/-24 days, where a change in followers happens and after 23/24 days there is a corresponding change in the stock price.

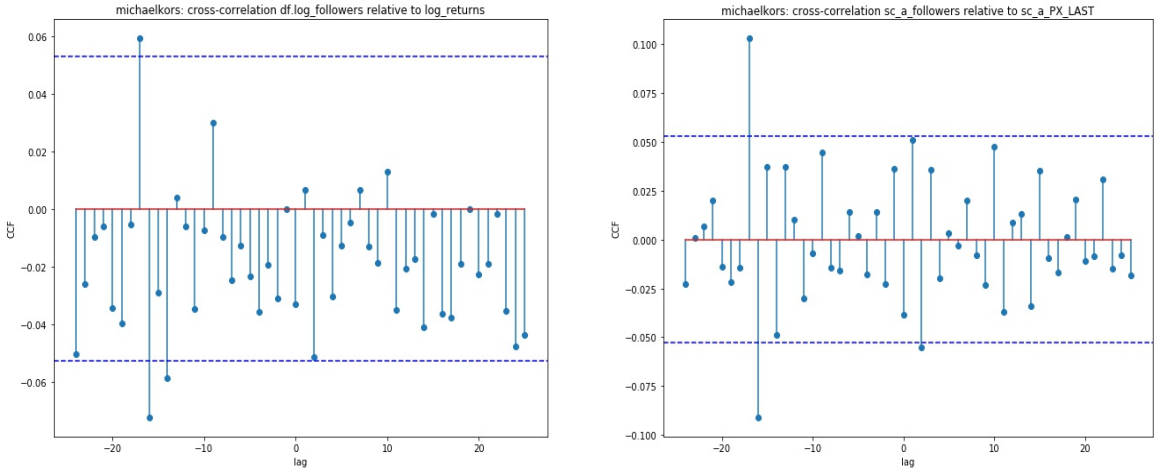
A similar trend can be observed in the CCF plots for $Michael\ Kors$, the significant cross-correlation between the change of logarithmic transform of the followers and returns is 0.06 at the time lag of -17 days and -0.07 at the time lag of -16 days (Figure 4.25 (A)). In addition, the significant cross-correlations between the acceleration of followers and the acceleration of stock prices are 0.1 and -0.095 at the time lags of -18 days and -17 days respectively (Figure 4.25 (B)).

Considering all the frequency distribution plots of significant cross-correlation coefficients ($r_{significant} \in [-1, -0.05) \cup (0.05, 1]$) across all companies (Figure 4.16 - 4.17), it can be observed that none of the hypothesis show any significant trend. According, to the Figure 4.22, the Hypothesis 5, which states that the changes in company's followers' acceleration is positively correlated with the changes in its stock price acceleration with a negative lag l, has the highest count of significant cross-correlation values (48 out of 561 for the continuous case).



(A) (B)

FIGURE 4.24: Ralph Lauren: (A) cross-correlation between the change in logarithmic transform of the followers and returns; (B) cross-correlations between the acceleration of followers and the acceleration of stock prices. The significant cross-correlations are observed at the time lags of -24 days and -23 days in both (A) and (B).



(A) (B)

FIGURE 4.25: Michael Kors: (A) cross-correlation between the change in logarithmic transform of the followers and returns; (B) cross-correlations between the acceleration of followers and the acceleration of stock prices. The significant cross-correlations are observed at the time lags from -18 days to -16 days for both (A) and (B).

A further analysis of the acceleration features could be implemented by eliminating the noise and cross-correlating only the periods of high volatility (events). For example, select the periods during which the follower acceleration is higher than *normal* (event periods) and cross-correlate them with the volatile periods of stock

price changes. This would help to eliminate the noise in both time series and show if events on Instagram have an impact on the stock prices. Moreover, further analysis with new data could be done with $Ralph\ Lauren$ and $Michael\ Kors$ follower counts and stock prices as the cross-correlation analysis shows relatively high cross-correlation between the changes in followers and stock prices (refer to the Figures 4.24 and 4.25).

Relationship between Followers Acceleration and Change in Stock Price

In addition, we further analyze the relationship between the mean acceleration of followers and the change in stock prices during a period of 1 year. For each asset, we calculate a mean yearly followers acceleration for every year between 2015 and 2018. Then, we also calculate the percentage change in the stock price during the same time periods. The Figure 4.26 shows a plot between the yearly normalized mean followers accelerations and the corresponding change in stock prices for all 11 assets. There is a positive relationship between the two values. We find that the yearly mean followers' acceleration is positively correlated with the change in stock price with a correlation coefficient of 0.48 and p-value - 0.006 which is a statistically significant correlation. This shows that if a company is consistently gaining Instagram followers during a year it is more likely that its stock price will increase during this period. A consistent increase in a company's popularity (e.g. mean followers acceleration) could be assumed to be reflected in its revenues (more sales), therefore an increase in the asset value.

4.5 Popularity and Revenues

The second part of the experiment aims to analyze whether there is a relationship between the average change in company's Instagram followers during the time period t and its revenue over the same period t or $t+1$. It is assumed that there is a positive correlation between the two values. All 11 companies, analyzed in this experiment, report their revenues every 3 or 6 months. On the day of a company's revenue report its share price experiences positive or negative change based on whether a company managed to meet the revenue expectations or not. As the changes in company's revenue are closely related to its stock price movements, the goal of this analysis is to measure whether a change in a company's revenue is related to the average change in the number of its Instagram followers over the revenue reporting period.

4.5.1 Experiment Design and Implementation

The dataset containing the daily followers' count, acceleration, velocity and revenues for all 11 companies are used for this experiment. For each company, the

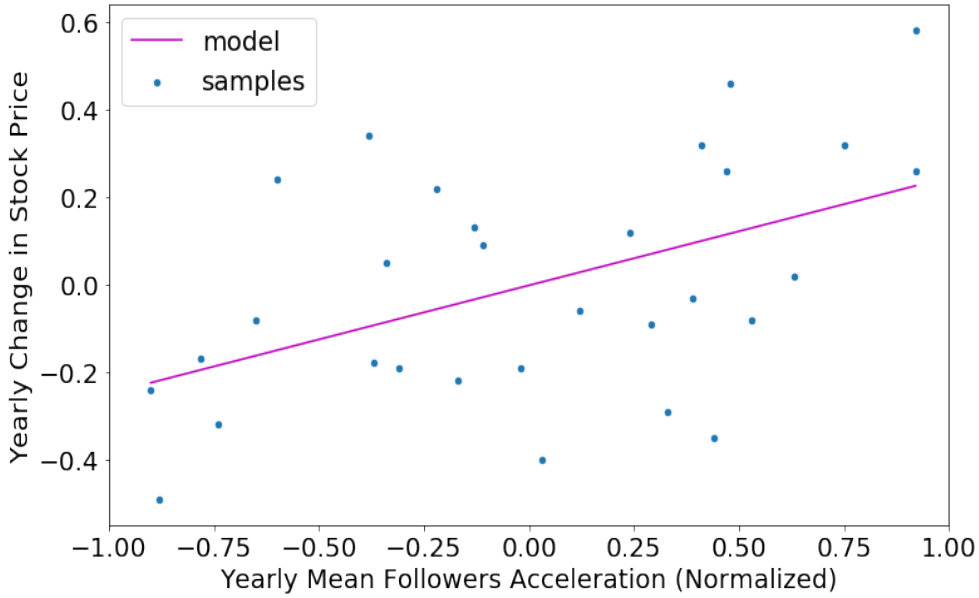


FIGURE 4.26: The relationship between the yearly mean followers acceleration and the change in stock price. Each data point represents the relation between the normalized mean acceleration and the change in stock price of a company during one year period. The Pearson correlation coefficient between the two values is 0.48 and $p - value$ = 0.006 (with α = 0.05) which shows a statistically significant positive correlation. The linear regression function is fitted to visualise the relationship between the two variables (included as a guide): $f(x) = -0.002 + 0.25x$. MSE = 0.05, R^2 = 0.24.

followers' acceleration and velocity are averaged across the revenue reporting period and then used for further analysis. The cross-correlations between the following (binary) values are measured for each company: 1) the change in revenue and the average followers' acceleration; 2) the change in revenue and the average followers' velocity; 3) the change in revenue and the average relative change in followers. The Figures 4.27 - 4.29 show the example plots for $Burberry$.

4.5.2 Experiment Results

From the visual observation of time series plots for all 11 companies, some similarities between the changes in revenue and the changes in followers could be observed. For example, $Mulberry$ exhibits a similar trend between the changes in revenues and the average relative followers count in the Figure 4.30, this similarity might be due to seasonality. The average cross-correlations across all companies between the following features are:

- change in revenue and average followers acceleration: 0.04.
- change in revenue and average followers velocity: -0.009.
- change in revenue and average relative change in followers: 0.12

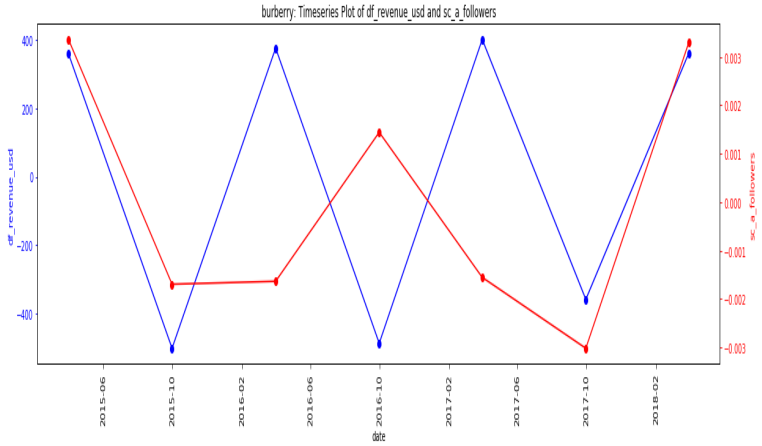


FIGURE 4.27: Burberry: time series plot of the changes in revenue and average acceleration of followers over the revenue reporting period.

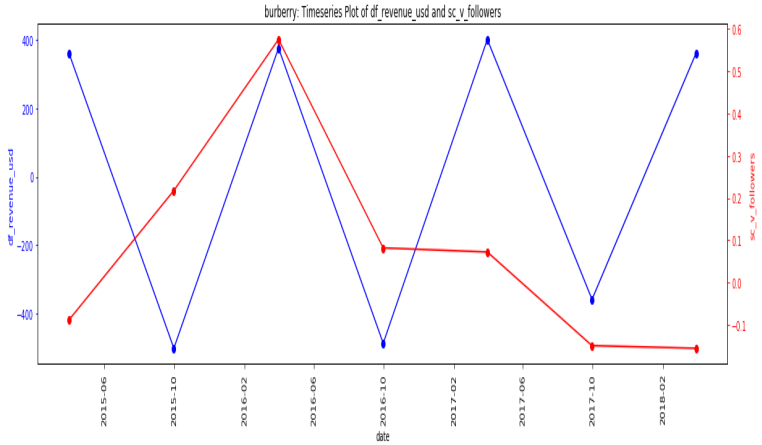


FIGURE 4.28: Burberry: time series plot of the changes in revenue and average velocity of followers over the revenue reporting period.

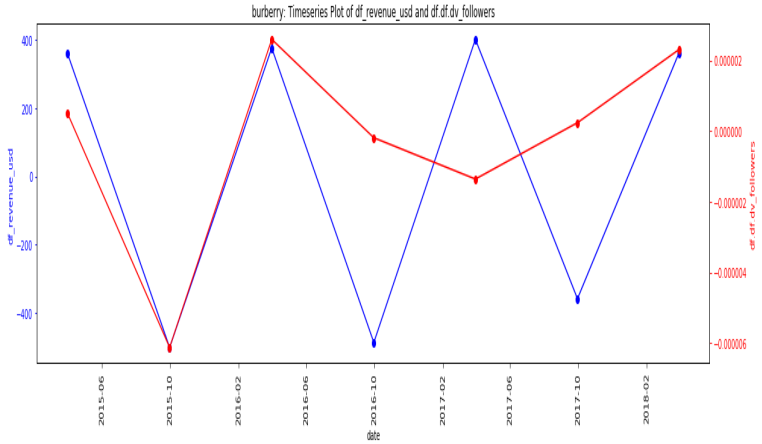


FIGURE 4.29: Burberry: time series plot of the changes in revenue and average relative difference in followers over revenue the reporting period.

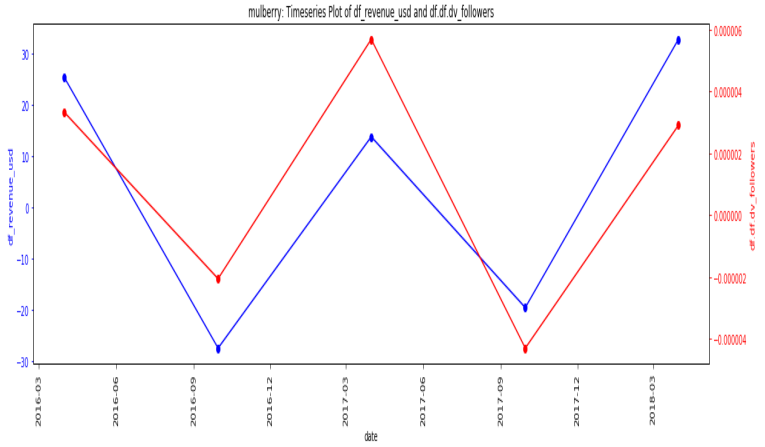


FIGURE 4.30: Mulberry: time series plot of the change in revenue and average relative difference in followers over the revenue reporting period.

To summarise, the analysis does not show any significant relationship between the values across all companies, although further analysis with new data could be done analyzing specific assets such as $Mulberry$ that exhibits more apparent trend between the followers and revenue.

4.6 Instagram Trading Strategies

Similarly, as in the research work by Preis, 2013, two hypothetical investment strategies are designed to test whether the information about the followers could improve the investment returns. The strategies are defined as per following (the fees associated with the execution of the trades are not incorporated within the estimate of the return):

Instagram Strategy 1: in this strategy we sell a stock at the closing price $p(t+1)$ on the first trading day of the week $(t+1)$ if the average acceleration a_{avrg} of the followers during the previous week t increases above the threshold th_s and close the sell position by buying the stock back at the closing price $p(t+2)$ on the first trading day of the following week $(t+2)$. We buy the stock at the closing price $p(t+1)$ on the first trading day of the week $(t+1)$ if the average acceleration a_{avrg} during the week t decreases below the threshold th_b and close the buy position by selling the stock at the closing price $p(t+2)$ on the first trading day of the following week $(t+2)$. If the a_{avrg} falls within the threshold range ($th_b < a_{avrg} < th_s$), no trades are executed. The Figure 4.31 shows a visual representation of the *Instagram Strategy 1* and how the trades are executed. The returns on the sell and buy positions are defined by the equations 4.15 and 4.16 respectively, and the combined return is defined by the equation 4.17.

$$return_{sell} = \frac{p(t+1)}{p(t+2)},\ if\ a_{avrg}(t)\ > th_s \tag{4.15}$$

$$return_{buy} = \frac{p(t+2)}{p(t+1)},\ if\ a_{avrg}(t)\ < th_b \tag{4.16}$$

$$return_{comb} = return_{sell} + return_{buy} \tag{4.17}$$

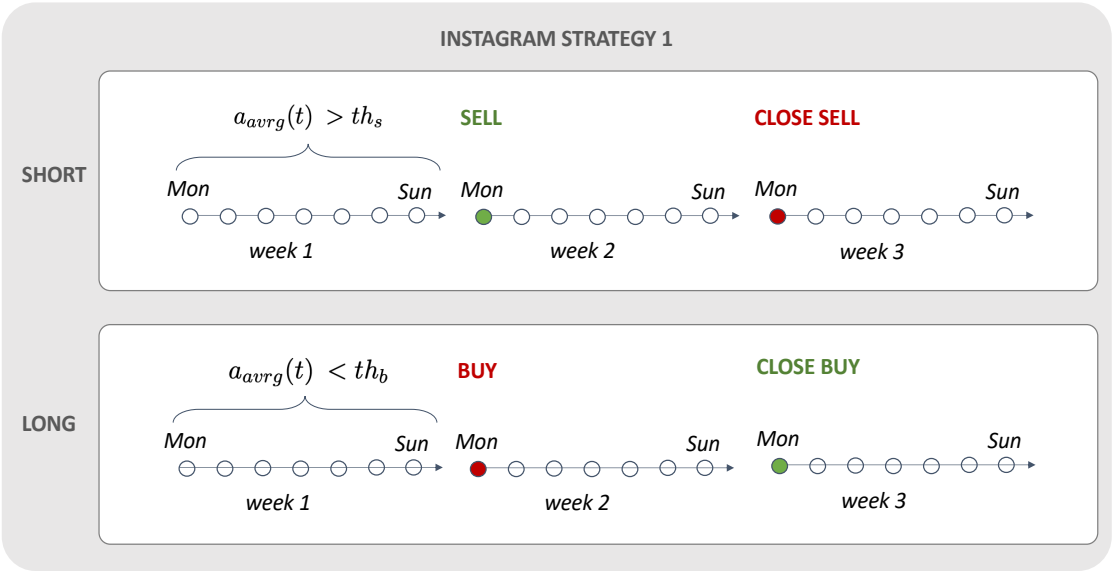


FIGURE 4.31: Instagram Strategy 1: short (sell) a stock at the closing price $p(t+1)$ on the first trading day of the week $(t+1)$ if the average acceleration a_{avrg} of the followers during the previous week t increases above the threshold th_s and close the short position by buying the stock back at the closing price $p(t+2)$ on the first trading day of the following week $(t+2)$. Long (buy) a stock at the closing price $p(t+1)$ on the first trading day of the week $(t+1)$ if the average acceleration a_{avrg} during the week t decreases below the threshold th_b and close the long position by selling the stock at the closing price $p(t+2)$ on the first trading day of the following week $(t+2)$.

Instagram Strategy 2: we sell a stock at the closing price $p(t+1)$ on the first trading day of the week $(t+1)$ if the average acceleration a_{avrg} of the followers during the previous week t increases above the threshold th_s and close the sell position by buying the stock back at the price $p(t+z)$ when the a_{avrg} drops below the threshold th_b, where z is the number of weeks after which the threshold th_b is reached. We buy the stock at the closing price $p(t+1)$ on the first trading day of the week $(t+1)$ if the average acceleration a_{avrg} during the previous week t decreases below the threshold th_b and close the buy position by selling the stock at the closing price $p(t+z)$ when the a_{avrg} increases above the threshold th_s. Similarly as in the Strategy

1, if the a_{avrg} falls within the threshold range ($th_b < a_{avrg} < th_s$), no trades are executed. The returns on the sell and buy positions are defined by the equations 4.18 and 4.19, the combined return is defined as per equation 4.17. The Figure 4.32 shows a visual representation of the *Instagram Strategy 2* and how the trades are executed.

$$return_{sell} = \frac{p(t+1)}{p(t+z)},\ if\ a_{avrg}(t)\ > th_s,\ p(t+z)\ at\ a_{avrg}(t+z)\ \leq th_b \tag{4.18}$$

$$return_{buy} = \frac{p(t+z)}{p(t+1)},\ if\ a_{avrg}(t)\ \leq th_b,\ p(t+z)\ at\ a_{avrg}(t+z)\ > th_s \tag{4.19}$$

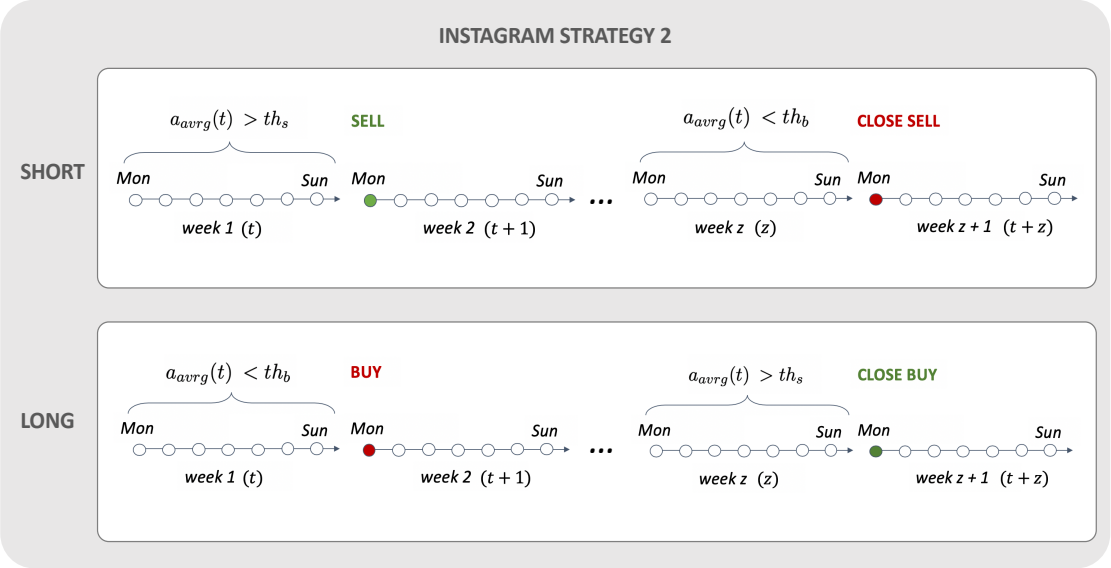


FIGURE 4.32: Instagram Strategy 2: short (sell) a stock at the closing price $p(t+1)$ on the first trading day of the week $(t+1)$ if the average acceleration a_{avrg} of the followers during the previous week t increases above the threshold th_s and close the short (sell) position by buying the stock back at the price $p(t+z)$ when the a_{avrg} drops below the threshold th_b, where z is the number of weeks after which the threshold th_b is reached. Long (buy) a stock at the closing price $p(t+1)$ on the first trading day of the week $(t+1)$ if the average acceleration a_{avrg} during the previous week t decreases below the threshold th_b and close the long position by selling the stock at the closing price $p(t+z)$ when the a_{avrg} increases above the threshold th_s.

In order to analyze the returns generated by the above trading strategies, the returns are compared with the *random* returns generated by a random trading strategy where stocks are bought and sold randomly. Below is a description of the *Random Strategy* that is used in parallel with the *Instagram Strategy 1* and *Instagram Strategy 2*

strategies in order to measure whether the returns generated by the Instagram strategies are significantly different from the returns generated by the *Random Strategy*.

Random Strategy : in this random strategy we sell a stock at the closing price $p(t)$ on the first trading day of the week (t) which is chosen randomly and close the sell position by buying the stock back at the closing price $p(t+1)$ on the first trading day of the following week $(t+1)$. We buy a stock at the closing price $p(t)$ on the first trading day of a random week (t) (chosen randomly) and close the buy position by selling the stock at the closing price $p(t+1)$ on the first trading day of the following week $(t+1)$. The combined return generated by this strategy is defined as per equation 4.17

4.6.1 Experiment Design and Implementation

The aim of the experiment is to measure whether the information on the changes in Instagram followers can help to generate better returns in comparison with a random trading strategy. Below is an overview of all hypotheses that are tested during this experiment (Wilcoxon signed-rank test is used to test the hypotheses):

- **Hypothesis 1** : the Null Hypothesis (H_0) - there is no significant difference between the distributions of returns ($return_{comb}$) generated by the *Random Strategy* and *Instagram Strategy 1* for each company. The opposite hypothesis - H_1.
- **Hypothesis 2** the Null Hypothesis (H_0) - there is no significant difference between the distributions of returns ($return_{comb}$) generated by the *Random Strategy* and *Instagram Strategy 2* for each company. The opposite hypothesis - H_1.
- **Hypothesis 3** the Null Hypothesis (H_0) - there is no significant difference between the distributions of returns ($return_{comb}$) generated by the *Random Strategy* and *Instagram Strategy 1* for a portfolio consisting of all 11 companies. The opposite hypothesis - H_1.
- **Hypothesis 4** the Null Hypothesis (H_0) - there is no significant difference between the distributions of returns ($return_{comb}$) generated by the *Random Strategy* and *Instagram Strategy 2* for a portfolio consisting of all 11 companies. The opposite hypothesis - H_1.

For each hypothesis, we also perform a grid search across different buy and sell thresholds (th_b and th_s) in order to find the values that maximise the returns.

4.6.2 Experiment Results

After testing the above hypotheses using Wilcoxon signed-rank test, we conclude the following.

TABLE 4.9: Hypothesis 1 testing results for Burberry. No significant difference found between the returns generated by the *Random Strategy* and the *Instagram Strategy 1*.

	Instagram Strategy 1				Random Strategy					Hypothesis 1 Testing		
	Followers Acceleration Thresholds		Returns Distribution		Returns Distribution		Wilcoxon: single sample Normal Distribution Testing H0: returns from Random Strategy are distributed normally (alpha=0.05)			Wilcoxon: two sample H0: returns from Instagram Strategy 1 and Random Strategy are from the same distribution (alpha=0.05)		
Company	Buy	Sell	Mean	Std. Dev.	Mean	Std. Dev.	Test Statistics	p-value	Test Result	Test Statistics (W)	p-value	Test Result
Burberry	-2586.3	209.7	0.0	0.04	0.0	0.04	368.0	0.572	fail to reject H0	175	0.990	fail to reject H0

Hypothesis 1: There is no significant difference between the returns generated by the *Random Strategy* and the *Instagram Strategy 1* across all companies. As an example, see the Figure 4.33 that shows the density distributions of the returns generated by both strategies for $Burberry$. The Hypothesis 1 testing results for Burberry are presented in the Table 4.9.

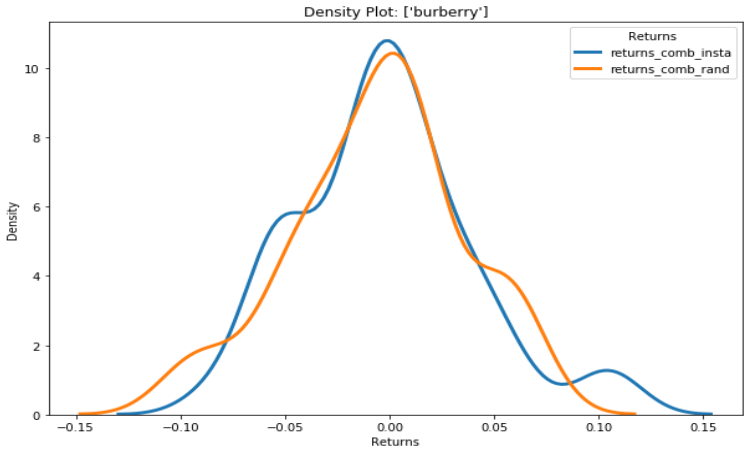


FIGURE 4.33: Hypothesis 1 - Burberry: the density distribution of the returns generated by the *Instagram Strategy 1* (blue) and the *Random Strategy* (orange) with no significant difference. Wilcoxon test statistics=175.000, p=0.990, fails to reject H0: both samples are from the same distribution. The returns are displayed using a kernel density estimate (KDE) with a Gaussian kernel and a bandwidth estimated using Silverman's rule of thumb.

Hypothesis 2 When considering the hypothesis tests independently, the statistically significant differences between the returns generated by the *Random Strategy* and the *Instagram Strategy 2* are found for the following companies: $Brunello\ Cucinelli$, $Prada$, $Ralph\ Lauren$, $Under\ Armour$ (assuming $\alpha = 0.05$). This might happen due to a chance, therefore, we apply Bonferroni correction for multiple hypothesis testing and get the new alpha value: $\alpha = 0.05/11 = 0.0045$, where 11 is a number of assets used for the hypothesis testing. We find that only $Ralph\ Lauren$ returns generated by the *Instagram Strategy 2* are significantly different from the *Random Strategy* returns. The summary of the tests results can be found in the Table 4.10. The Figure 4.34 shows the returns generated by the *Instagram Strategy 2* and the *Random*

TABLE 4.10: Hypothesis 2 testing results: individual asset returns generated by the *Random Strategy* and the *Instagram Strategy 2*.

	Instagram Strategy 2				Random Strategy					Hypothesis 1 Testing		
	Followers Acceleration Thresholds		Returns Distribution		Returns Distribution		Wilcoxon: single sample Normal Distribution Testing H0: returns from Random Strategy are distributed normally (alpha=0.0045)			Wilcoxon: two sample H0: returns from Instagram Strategy 1 and Random Strategy are from the same distribution (alpha=0.0045)		
Company	Buy	Sell	Mean	Std. Dev.	Mean	Std. Dev.	Test Statistics	p-value	Test Result	Test Statistics	p-value	Test Result
Brunello Cucinelli	-2.13	1.38	0.002	0.04	0.0	0.03	2820.5	0.15	fail to reject H0	1038.0	0.03	fail to reject H0
Prada	-43.02	16.74	0.01	0.07	0.0	0.05	6199.0	0.50	fail to reject H0	2425.0	0.032	fail to reject H0
Ralph Lauren	-264.36	0.54	0.04	0.12	0.0	0.04	2715.0	0.15	fail to reject H0	1725.0	0.002	reject H0
Under Armour	-340.05	40.57	0.18	0.45	0.0	0.07	1594.0	0.90	fail to reject H0	849.0	0.02	fail to reject H0

Strategy for $Ralph\ Lauren$. Here, it can be observed that the mean returns generated by the *Instagram Strategy 2* are higher than the returns generated by the *Random Strategy*. This shows that in the case of $Ralph\ Lauren$ the followers' data could potentially be used as one of the trading signals or features when designing trading strategies. Even though the signal is weak, the popularity data could still contribute to the overall performance of a trading strategy when used in conjunction with other information.

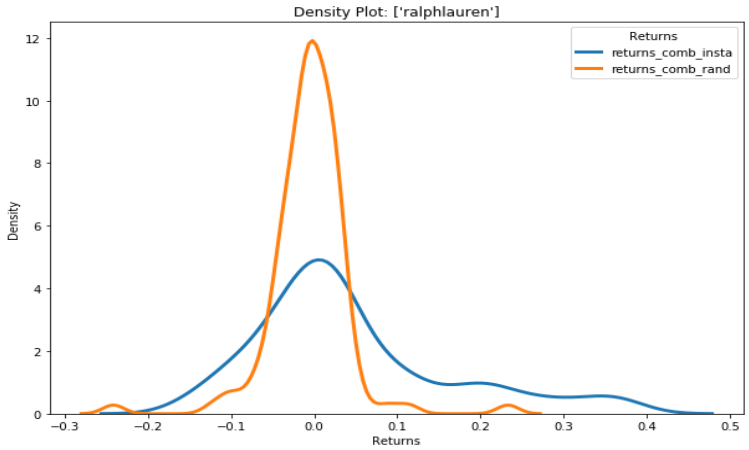


FIGURE 4.34: Hypothesis 2 - Ralph Lauren: the density distribution of the returns generated by the *Instagram Strategy 2* (blue) and the *Random Strategy* (orange) with statistically significant difference. Wilcoxon test statistics=1725.0, p=0.002 (with $\alpha = 0.0045$), reject H0: both samples are from the different distributions. The returns are displayed using a kernel density estimate (KDE) with a Gaussian kernel and a bandwidth estimated using Silverman's rule of thumb.

Hypothesis 3 There is no significant difference between the portfolio (with 11 assets) returns generated by the *Random Strategy* and the *Instagram Strategy 1*. The density distributions of the returns are presented in the Figure 4.35.

Hypothesis 4 There is no significant differences between the portfolio (with 11 assets) returns generated by the *Random Strategy* and the *Instagram Strategy 2*. The density distributions of the returns are presented in the Figure 4.36.

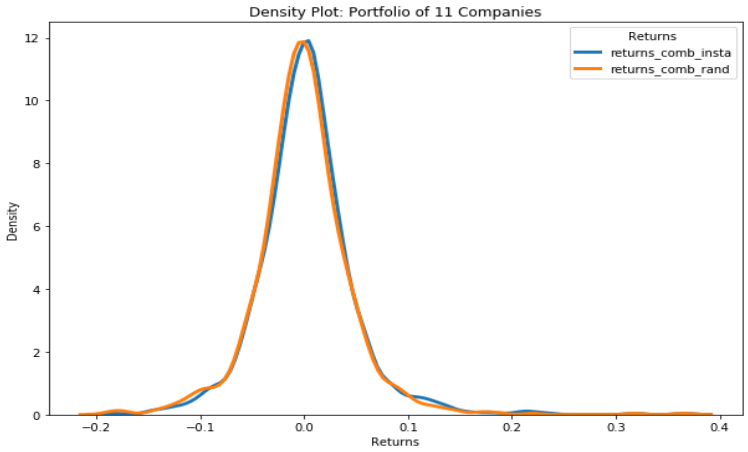


FIGURE 4.35: Hypothesis 3 - the portfolio of 11 companies: the density distribution of the returns generated by the *Instagram Strategy 1* (blue) and the *Random Strategy* (orange), there is no statistically significant difference between the two distributions. Wilcoxon test statistics=76315.0, p=0.1, fail to reject H0: both samples are from the same distribution. The returns are displayed using a kernel density estimate (KDE) with a Gaussian kernel and a bandwidth estimated using Silverman's rule of thumb.

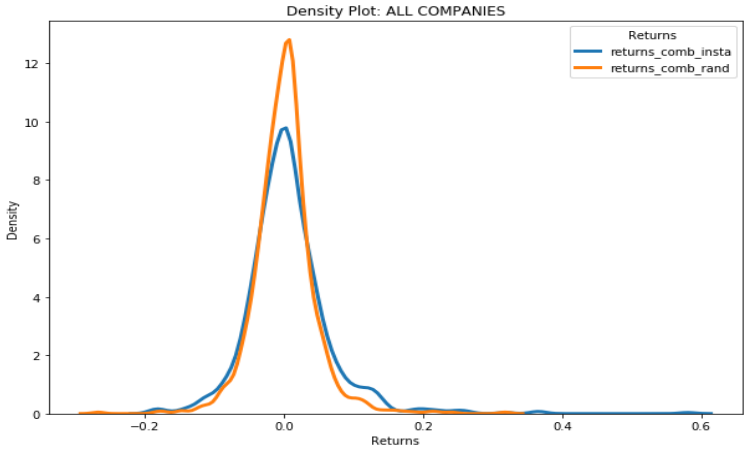


FIGURE 4.36: Hypothesis 4 - the portfolio of 11 companies: the density distribution of the returns generated by the *Instagram Strategy 1* (blue) and the *Random Strategy* (orange), there is no statistically significant difference between the two distributions. Wilcoxon test statistics=109432.0, p=0.07, fail to reject H0: both samples are from the same distribution. The returns are displayed using a kernel density estimate (KDE) with a Gaussian kernel and a bandwidth estimated using Silverman's rule of thumb.

4.7 Conclusion

To summarize, the experiment results suggest that the popularity of a fashion company on Instagram could provide insights into the future financial performance of a company. Below we summarise the key insights from the experiment:

- The certain assets demonstrate a statistically significant cross-correlation between the changes in the followers and returns. We find that *Michael Kors* and *Ralph Lauren* datasets show a positive statistically significant cross-correlation between the following time series: 1) the change in the log transform of followers and the log transform of returns; 2) the acceleration of followers and the acceleration of stock prices (the change of returns). The cross-correlation between the followers and returns is significant with the time lag between 16 to 24 days, i.e. according to the analysis, a change in followers happens and after 16-24 days it is *reflected* in the stock price. For the remaining 9 companies, there are no significant cross-correlations found.

- There is a statistically significant positive correlation (correlation coefficient = 0.48, p-value = 0.006) between the yearly mean followers' acceleration and the change in the stock price over the same period of time (refer to the Figure 4.26). This shows that if a fashion company continuously gains followers at a high rate during a period of 1 year, its stock price is likely to increase and visa versa.

- One of the Instagram trading strategies, *Instagram Strategy 2*, shows statistically significant difference between the returns generated by the *Random Strategy* and *Instagram Strategy 2* for *Ralph Lauren* asset. Here, the mean returns generated by the *Instagram Strategy 2* are higher than the returns generated by the *Random Strategy*. As the *Instagram Strategy 2* executes the trades (buy or sell) only when a certain followers acceleration threshold is reached, this shows that the relationship between the Instagram and stock market could be related only at some points in time and not continuously. This means that only certain events (i.e. an abnormal increase in followers acceleration volatility) on Instagram could have an impact on the asset prices. An increase in the followers' volatility might demonstrate a sudden hike in interest or decrease in the relevancy of a company among the consumers signalling a potential impact on its financial future.

The experiments presented in this chapter show that a fashion company's popularity data on Instagram could potentially be used as one of the trading signals or features when designing investment strategies. The signals derived from the changes in followers on Instagram might be very weak but could still contribute to the overall performance of the strategy when used in conjunction with other information about the asset. Thus, when making investment decisions, a company's popularity could be used as one of the alternative data points to model the future financial performance of a company.

Chapter 5

Dynamic Density-based News Clustering

This chapter outlines the second experiment during which the unsupervised framework is built for automatic company's news clustering. The main research problem presented in this section is how important events reported in the news could be identified without any prior knowledge about the event types, a number of events or reporting frequency. The framework utilizes a density-based clustering algorithm DBSCAN to cluster the news articles without a need to pre-define the parameters such as cluster density or size. The parameters are selected automatically using Gaussian Mixture Model method and Information Entropy. The experiment is done in collaboration with Arabesque Asset Management.

5.1 Problem Overview

News is one of the most important information sources that help investors to make their trading decisions. High impact, unique events such as changes in a company's management structure, fraud, illegal activities, shop closures, product launches, employee scandals etc., most often have an impact on a company's financial performance. Therefore, it is important to track these events in order to quickly react and mitigate the risks exposed to portfolio stocks. There are thousands of news sources globally and it is impossible for a human to consume all of the information. Consequently, investors, service providers, academics are now researching and building systems to automatically *read* the news. This became a rapidly growing research area. Although, the majority of research work is aimed at predicting company's stock price movements based on the news articles (Ding, 2014; Akita, 2016; Ding, 2015; Huynh, 2017; Ding, 2016; Liu, 2018a; Merello, 2018; Li, 2018). Meanwhile, there is a lack of research done in analysing the properties of industry-specific company events (e.g. Apparel/Footwear) and how they can be automatically identified in the news articles. The key problem with the news articles is that they are extremely noisy and contain a lot of irrelevant information. At the same time, the event properties are dynamic and depend on a company. For example, the volume/popularity of the articles reporting an event vary across different companies within the industry; the similarity between the articles that are talking about the same event is inconsistent

and therefore it is problematic to find the boundary between the noise and relevant event article. Here we define noise as irrelevant information (news articles). Thus, there is a need for better domain-specific text analytics techniques to separate the useful news information from the noise.

In this experiment, instead of modelling how the news articles impact the stock prices, the aim is to analyse the characteristics of the news articles by designing the unsupervised framework that takes the company-related news articles and extracts events without any prior knowledge about the events such as a number of events, their type, frequency or popularity. The main goal of the framework is to cluster the incoming company's articles daily and extract the *events* (if any) out of the noise with no human input (no predefined parameters). The framework uses density-based clustering technique DBSCAN where the parameters are dynamically determined from the data using Gaussian Mixture Model and Information Entropy methods. In this study we aim to answer the following questions:

1. Which most commonly used distance metric (Cosine Similarity, Euclidean Distance and Manhattan Distance) measures the similarity between the news articles the best.

2. How dynamically determine the separation boundary between the event articles and the noise without the predefined parameters.

3. How to design the overall framework for the unsupervised news event clustering.

This chapter is structured as follows: first, we present the dataset used for this research, second, we describe the methodology used during the experiment, then we present the framework (algorithm) for clustering the news articles and discuss the results.

5.2 Dataset

The dataset used for this experiment is comprised of English news articles that mention 40 different U.S. and European companies operating in the Apparel and Footwear industry. The total amount of articles is 177,309 published during the period between April 2017 and August 2018. Each article has an associated list of companies mentioned in the article. The articles are obtained from the news aggregator service EventRegistry[1] and by scraping online fashion blogs and magazines (see more details about the data sources in Chapter 3). The Figure 5.1 shows the top 25 assets with the most news articles. It can be observed that 50% of the dataset is comprised of articles about the sportswear companies Nike and Adidas. As an article representation, we choose to use its headline only.

[1]EventRegistry - a news aggregator service (*https://www.eventregistry.org*).

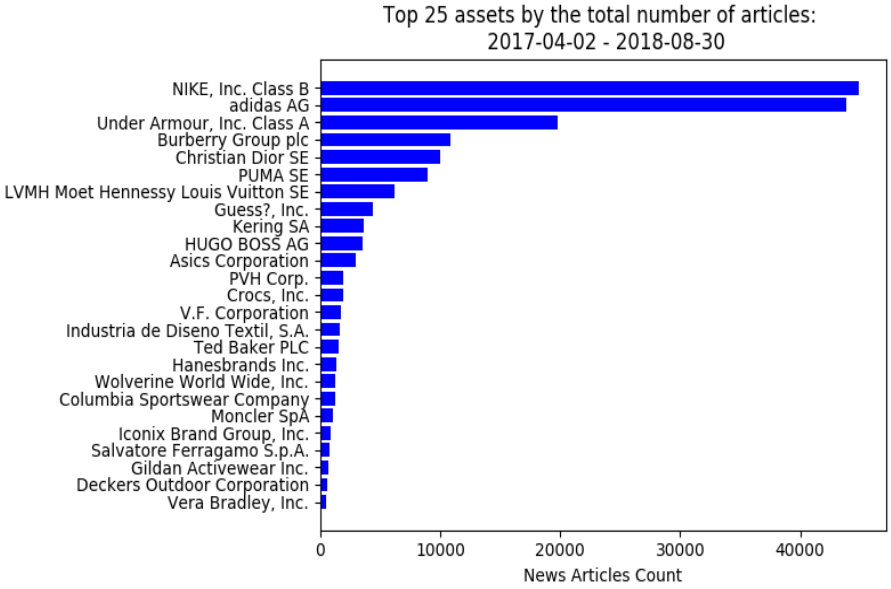


FIGURE 5.1: The top 25 assets with the most news articles. The majority of dataset is comprised of the news articles mentioning sportswear companies: Nike and Adidas.

In addition, we also create a labelled dummy dataset for analysing the properties of the event-related articles and noise. For the dummy dataset, we select 1,000 articles released during a 3-day period that mention company Nike and label them with 0 (noise article) and 1 (event article) labels. In total we identified 100 articles that mention 4 different events, the rest of 900 articles are nosy irrelevant information. The Figure 5.2 shows an example of 5 articles that mention 1 event (in orange) together with noise (blue).

5.3 Methodology

The main objective of this experiment is to design a framework for unsupervised news articles clustering that does not require predefined input parameters. The aim is to cluster the articles daily and detect popular events reported in the news. During this study, DBSCAN clustering algorithm (Ester, 1996) is used to identify event clusters within the data. The density-based algorithm requires two input parameters: 1) eps - radius, i.e. distance between the points and 2) $minPts$ - a minimum number of points within the cluster. The parameters are pre-defined by the user and requires some prior knowledge about the dataset. The main issue with such an approach is that the parameters are fixed and not dynamically updated once the new data comes in (daily flow of news articles). Therefore, in this experiment, the aim is to design a method that automatically derives these parameters from the dataset and can dynamically adjust them after a period of time (e.g. a day) when a new batch of articles comes in. To design the framework the following methodology is implemented:

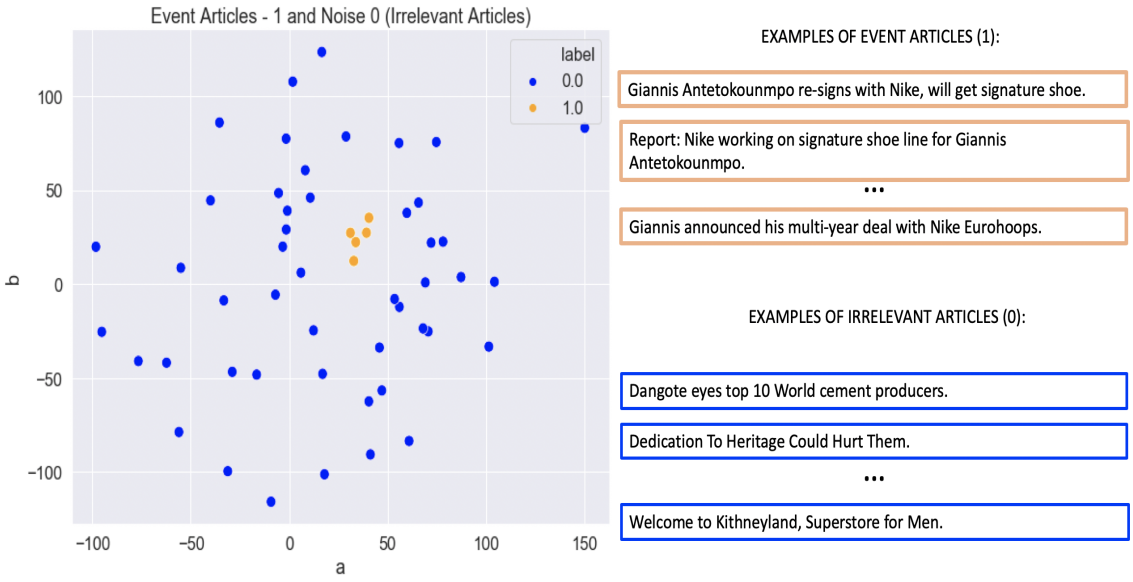


FIGURE 5.2: An example of the dummy dataset - the Figure on the left shows a cluster with the event articles (orange) and noise articles (blue). On the right, examples of event articles and noisy irrelevant information.

1. Article title embeddings - titles of the news articles are embedded using the Word2Vec representations (Mikolov, 2013).

2. Similarity metric selection - three different vector distance metrics - Cosine Similarity, Euclidean Distance and Manhattan Distance are analysed to find which metric represents the similarity between the titles the best.

3. Dynamic determination of the inter-cluster distances range - Gaussian Mixture Model is used to find the distance between the noise and events vectors.

4. DBSCAN parameters search - using the distance range determined by Gaussian Mixture Model, the best DBSCAN parameters (eps and $minPts$) are found for DBSCAN that maximize the inter-cluster Information Entropy.

Below we introduce the concepts of the Word2Vec embeddings, similarity metrics, Gaussian Mixture Model, Information Entropy and describe each methodology step in detail.

Word2Vec

Prior to the study (Mikolov, 2013), the words were mainly represented as indices in vocabulary e.g. using the one-hot encoding. Such an approach does not capture any similarity between the words and is not scalable. According to the language distributional structure, the words that appear in the similar locations have similar semantics (Harris, 1981) therefore based on this observation the word context in the corpora can be used to train word vector representations. The research

study (Mikolov, 2013) introduced an unsupervised Neural Network based method to compute continuous vector representations for words trained on a large text corpora. Two models for training the embeddings were proposed: 1) Continuous Bag-of-Words model (CBOW) with an objective to predict a word using a context of 8 words that surround it (4 words before and 4 after); 2) Continuous Skip-gram model (Skip-gram) is an opposite model to CBOW, it takes a single word and predicts other words within a range from a given word. Both models contain a Neural Network architecture with a single fully-connected hidden layer and output the probabilities of the target words using softmax. The Neural Network weights are adjusted during the training to minimize the loss function. After the model is trained the weights of a hidden layer are used as word embeddings.

Vector Similarity Metrics

There are a number of different distance metrics used to calculate a similarity between the two objects (e.g. vector representations of two different news articles). Many clustering algorithms (e.g. DBSCAN, K-Means, etc.) use similarity metrics to measure pair-wise distances between the samples and identify clusters within the data. Below is an overview of the three most commonly used similarity metrics in Natural Language Processing.

Euclidean Distance is one of the most commonly used distance metrics to find a proximity between two vectors. The equation 5.1 defines the Euclidean Distance between the two vectors $\mathbf{x}$ and $\mathbf{y}$ (where $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$).

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{5.1}$$

Manhattan Distance is another distance metric where the distance between two objects is a sum of absolute differences of their Cartesian coordinates. The Manhattan Distance between the two vectors $\mathbf{x}$ and $\mathbf{y}$ (where $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$) is defined as per equation 5.2.

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} |x_i - y_i| \tag{5.2}$$

Cosine Similarity defines an angle between two vectors measuring how close they are. The cosine similarity between the two vectors $\mathbf{x}$ and $\mathbf{y}$ (where $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$) is defined by the equation 5.3.

$$similarity(\mathbf{x}, \mathbf{y}) = cosine(\theta) = \frac{\mathbf{x} \cdot \mathbf{y}}{|\mathbf{x}||\mathbf{y}|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}} \tag{5.3}$$

Gaussian Mixture Model

Gaussian Mixture Model (GMM) is a probabilistic model that assumes that the data points within the dataset are generated from a mixture of *K* multivariate Gaussian distributions, where each distribution has a set of parameters - mean $\boldsymbol{\mu}_k$ vector and covariance matrix $\boldsymbol{\Sigma}_k$ (Murphy, 2012). The Gaussian Mixture Model is defined as per equation 5.4, where π_k - is mixing weights, $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$.

$$p(x_i|\theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{5.4}$$

In this experiment we use Gaussian Mixture Model for clustering, where we assume that the distances between the articles fall into one of the two groups (clusters, $K = 2$) - either event cluster articles or noise articles, in order to find the upper bound for inter-cluster distances.

The parameters of the distributions are determined by the Expectation Maximization (EM) algorithm - iterative method to compute maximum likelihood estimates of model parameters to fit the data with latent variables. Expectation Maximization algorithm has two main steps: 1) *E-step* - using the given dataset the latent values are estimated; 2) *M-step* - using all the data points (given dataset and estimated values from the *E-step*) the statistical model parameters are updated (Murphy, 2012).

Gaussian Mixture Model is fitted using EM as per following:

1. For each data point (x_i) in the dataset compute the posterior probability $p(z_i = k|x_i, \theta)$ that x_i belongs to the cluster k using Bayes rule as defined in the equation 5.5.

$$r_{ik} = p(z_i = k|x_i, \theta) = \frac{p(z_i = k|\theta)p(x_i|z_i = k, \theta)}{\sum_{k'=1}^{K} p(z_i = k'|\theta)p(x_i|z_i = k', \theta)} \tag{5.5}$$

2. Update the model parameters π_k, μ_k and Σ_k. The mixing weights π_k is defined as per equation 5.6, where N - total data points. Model mean μ_k and covariance Σ_k are updates as per equations 5.7 and 5.8 respectively.

$$\pi_k = \frac{\sum_{i=1}^{N} r_{ik}}{N} \tag{5.6}$$

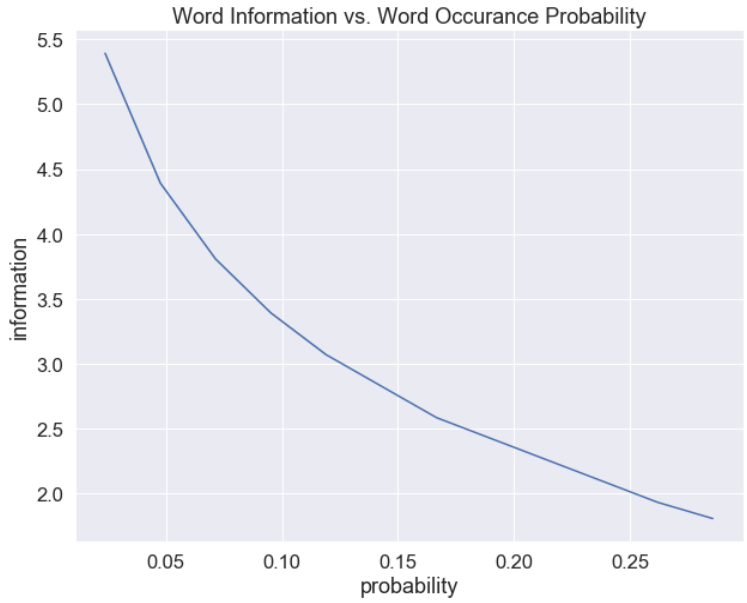


FIGURE 5.3: The relationship between the information content in words (from the news articles corpus) and their occurrence probabilities.

$$\mu_k^{new} = \frac{\sum_{i=1}^{N} r_{ik} x_i}{\sum_{i=1}^{N} r_{ik}} \tag{5.7}$$

$$\Sigma_k = \frac{\sum_{i=1}^{N} r_{ik}(x_i - \mu_k^{new})(x_i - \mu_k^{new})^T}{\sum_{i=1}^{N} r_{ik}} \tag{5.8}$$

Information Entropy

Information Theory is an area of mathematics concerned with quantifying information content in the data for communication (Murphy, 2012). The amount of information contained within the random variables, events or distributions can be measured using probabilities. The main idea behind quantifying information is evaluating how much *surprise* an event contains. The events with a lower probability contain more surprise hence more information whilst events that occur frequently (highly probable events) contain less surprise and low information. The information content for an event e is defined as per equation 5.9 where $P(e)$ is a probability of an event e. As an example, the Figure 5.3 shows how the information content in words (from the news articles corpus) is related to their occurrence probabilities, it can be observed that the words with higher probability contain less information (less *surprise*).

$$H(e) = -log_2(P(e)) \tag{5.9}$$

Information Entropy is an information content for a random variable (x). It is defined as per equation 5.10, where x is a random variable with K states, $P(k)$-

event probability in the state k. A random variable would have the largest Entropy if all the events have an equal likelihood.

$$E(x) = -\sum_{k=1}^{K} P(k) \times log_2(P(k)) \tag{5.10}$$

5.3.1 Article Title Embeddings

The research study (Li, 2015) found that summarized articles predict the stock price movement better than a full article, therefore for this study we choose to use an article headline as a representation of an article instead of a full article. The headlines are preprocessed using common natural language preprocessing techniques: 1) cleaned from symbols and non-English letters; 2) all words converted to lowercase; 3) English stop words removed; 4) words are stemmed. In addition, company names are masked in all of the titles in order to eliminate model bias towards any specific organization. Then, the titles are embedded using the Word2Vec pre-trained vectors (Mikolov, 2013). Here, each word in the title is represented by the corresponding pre-trained Word2Vec vector and the final title vector ($\mathbf{v}_{title}$) is an average of all word vectors in the title (refer to the equation 5.11 where n - number of words in the title, $\mathbf{w_i}$ - i^{th} word vector in the title, $\mathbf{w_i} \in \mathbb{R}^m$, for the pre-trained Word2Vec vectors $m = 300$). Then, all the headlines in the dataset have a corresponding vector representation $\mathbf{v}_{title}$ where $\mathbf{v}_{title} \in \mathbb{R}^{300}$.

$$\mathbf{v}_{title} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{w_i} \tag{5.11}$$

5.3.2 Similarity Metric Selection

The clustering algorithm, DBSCAN, uses a distance metric to find pair-wise proximities between the data points. We compare different distance metrics to select the best metric that measures the distances between the articles the best. The best performing metric is used as a DBSCAN distance metric in the final clustering framework. To find the best performing metric, we use the labelled dummy dataset, cluster the articles using three different versions of DBSCAN: 1) DBSCAN with Euclidean distance metric; 2) DBSCAN with Manhattan distance metric; 3) DBSCAN with Cosine Similarity metric, and find which metric yields the most accurate clustering results. The below steps are implemented to find the best distance metric:

- First, we analyse 4 event clusters within the dummy dataset. The aim is to find how the distances between the articles differ within the cluster (inter-cluster pairwise distances) and out of the cluster (proximities between the articles in the cluster and the noise articles). We calculate pairwise distances between all articles within the clusters together with their proximities to the noise articles. As a result, two distinct distributions of the proximities can be observed, the

Figure 5.4 shows Cosine Similarities between the inter-cluster articles (orange line) together with their Cosine Similarities to noise articles, cluster-noise (blue line). The distributions are plotted using the kernel density estimate (KDE), with a Gaussian kernel and bandwidth calculated with Silverman's rule of thumb (Silverman, 2018) as defined per equation 5.12, where σ- standard deviation and n- number of samples.

$$h = (\frac{4\sigma^5}{3n})^{\frac{1}{5}} \tag{5.12}$$

The distribution mean of the inter-cluster pairwise cosine similarities between the articles is 0.16 and the standard deviation - 0.06. Meanwhile, the distribution mean of the cluster-noise cosines similarities is 0.31 and the standard deviation - 0.09. There is a statistically significant difference between the inter-cluster proximities and cluster-noise distributions according to Wilcoxon signed-rank test: V = 12393.0, p = 0.0, α = 0.05. This shows that the articles that belong to event clusters in the dummy dataset have the average cosine similarity between each other of 0.16.

The Figures 5.5 and 5.6 show corresponding distributions for Euclidean and Manhattan distance metrics. The results for all three different distance metrics are summarized in the Table 5.1.

It can be observed that the inter-cluster distances are significantly different from the cluster-noise articles proximities.

In addition, there is a variation between the inter-cluster Cosine Similarities across different clusters. For example, the mean Cosine Similarity between the articles within the cluster 1 is 0.2, while cluster 2 and 3 - 0.15 and 0.18 respectively. The Figure 5.7 demonstrates the distributions of inter-cluster Cosine Similarities within the clusters 1,2 and 3.

Metric	Inter-Cluster μ, σ	Cluster-Noise μ, σ	Wilcoxon Test $\alpha = 0.05$
Cosine Similarity	$\mu = 0.16$, $\sigma = 0.06$	$\mu = 0.31$, $\sigma = 0.09$	$V = 12393.0$, $p = 0.0$
Euclidean Distance	$\mu = 1.87$, $\sigma = 0.42$	$\mu = 2.61$, $\sigma = 0.39$	$V = 14315.0$, $p = 0.0$
Manhattan Distance	$\mu = 25.75$, $\sigma = 4.85$	$\mu = 35.32$, $\sigma = 5.18$	$V = 14978.0$, $p = 0.0$

TABLE 5.1: Comparison of inter-cluster and cluster-noise articles proximity distributions generated by different distance metrics.

- Next, we cluster the articles in the dummy dataset using DBSCAN with all three different distance metrics. Then, we evaluate the clustering results to find the proximity metric that generates the most accurate results. As an example, the Figures 5.8 and 5.9 show *true cluster labels* (A) and *DBSCAN clustered labels* (B) generated by DBSCAN with Cosine Similarity distance metric. To compare the clustering outputs from all three different DBSCAN clustering methods (DBSCAN with Cosine Similarity, DBSCAN with Euclidean distance

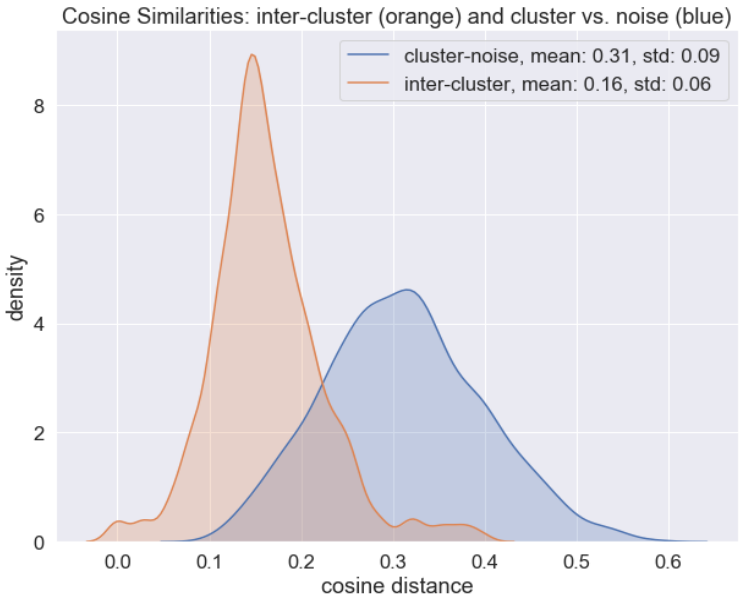


FIGURE 5.4: The distributions of Cosine Similarities between the inter-cluster articles (orange) and cluster articles vs. noise articles (cluster-noise) (blue). Both distributions are plotted using KDE with a Gaussian kernel and bandwidth calculated with Silverman's rule of thumb.

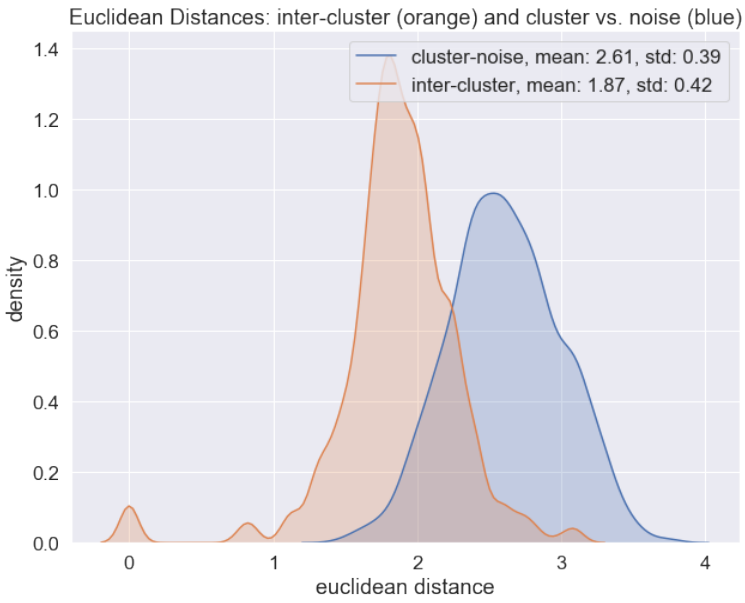


FIGURE 5.5: The distributions of Euclidean Distances between the inter-cluster articles (orange) and cluster articles vs. noise articles (cluster-noise) (blue). Both distributions are plotted using KDE with a Gaussian kernel and bandwidth calculated with Silverman's rule of thumb.

and DBSCAN with Manhattan distance), we use the following metrics: accuracy, precision, recall and F1 metrics. The evaluation results are presented in the Table 5.2. It can be observed that Cosine Similarity yields the best clustering results. Therefore, we choose Cosine Similarity as a proximity metric for

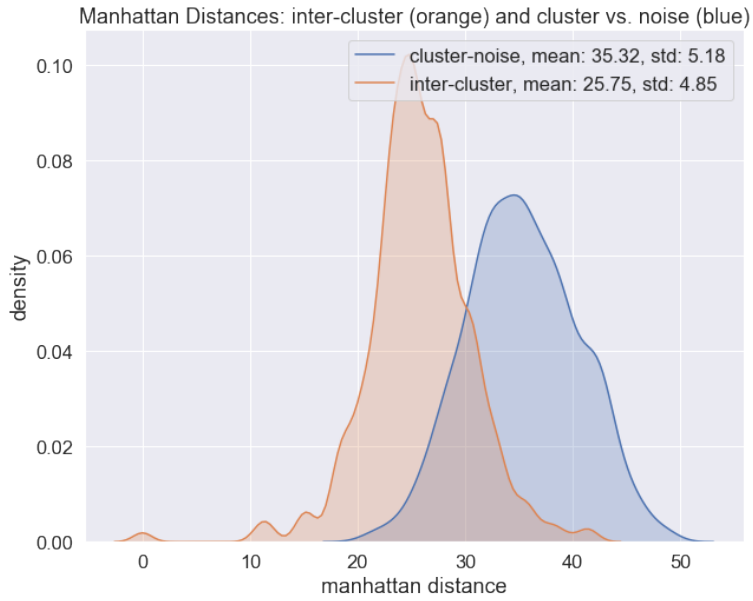


FIGURE 5.6: The distributions of Manhattan Distances between the inter-cluster articles (orange) and cluster articles vs. noise articles (cluster-noise) (blue). Both distributions are plotted using KDE with a Gaussian kernel and bandwidth calculated with Silverman's rule of thumb.

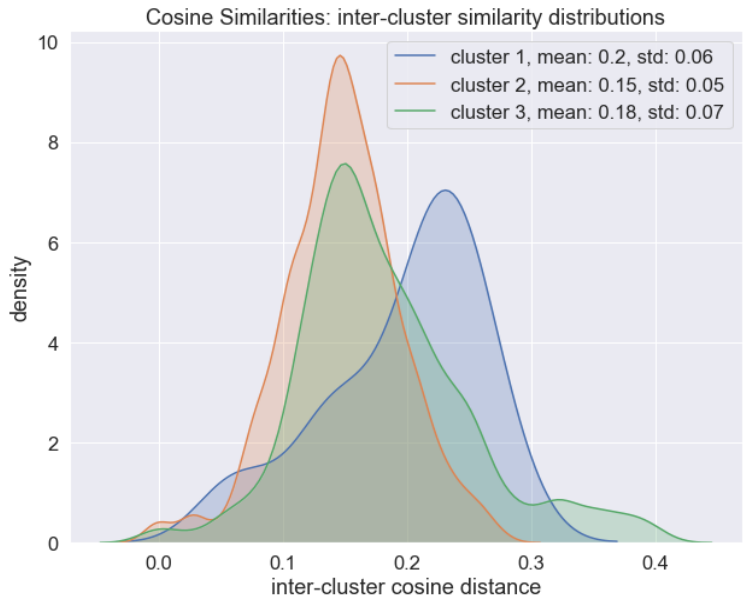
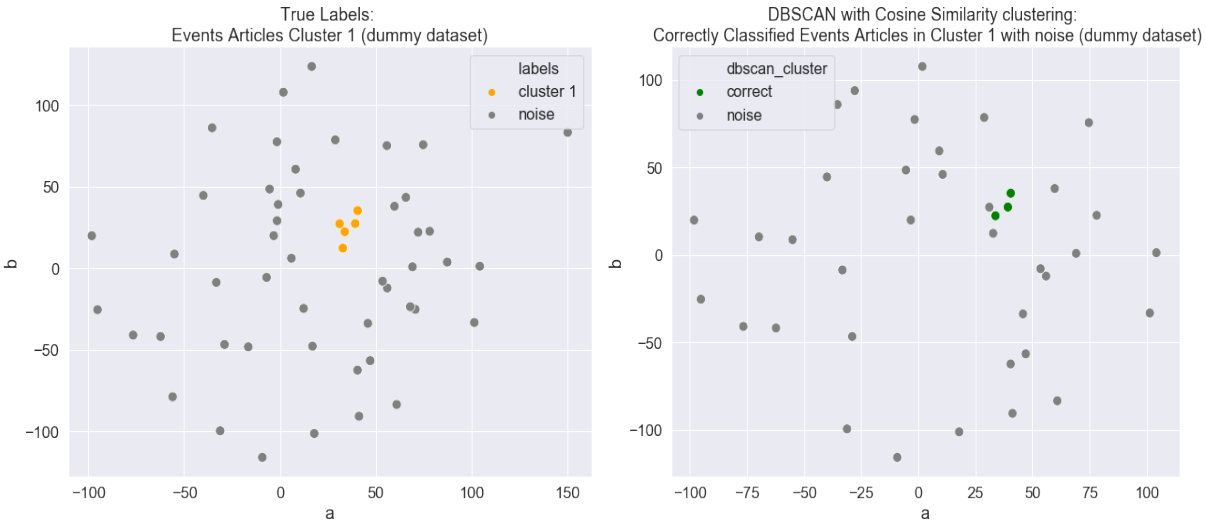


FIGURE 5.7: The distributions of inter-cluster Cosine Distances between the articles within the events clusters 1-3. The distributions are plotted using KDE with a Gaussian kernel and bandwidth calculated with Silverman's rule of thumb.

further modelling.

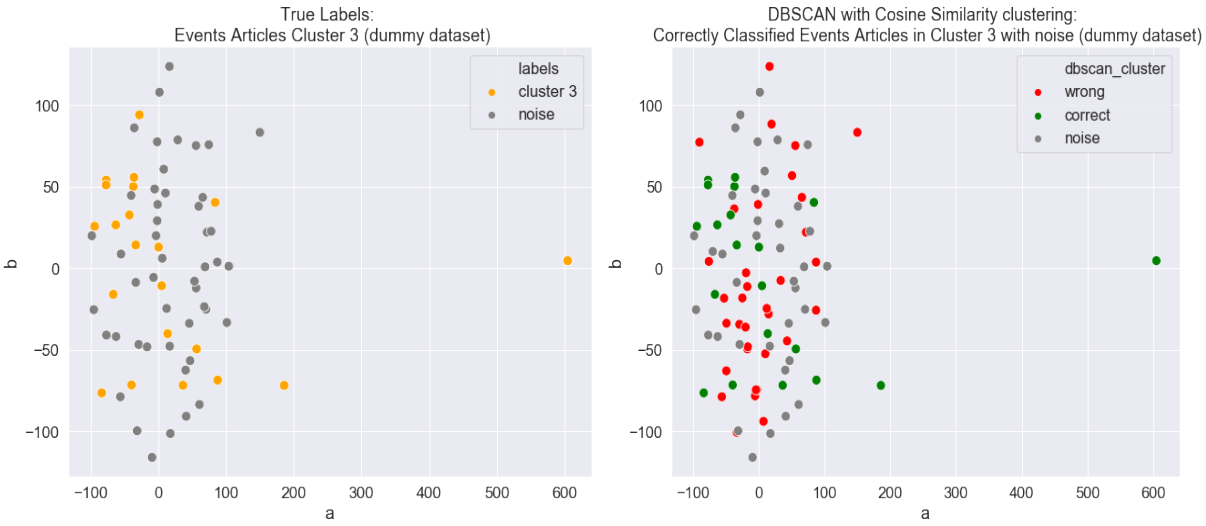
Metric	Accuracy	Precision	Recall	F1
DBSCAN - Cosine Similarity	0.57	0.45	0.45	0.41
DBSCAN - Euclidean Distance	0.55	0.21	0.28	0.22
DBSCAN - Manhattan Distance	0.53	0.21	0.28	0.21

TABLE 5.2: Clustering performance evaluation using different distance metrics for the DBSCAN clustering algorithm.



(A) (B)

FIGURE 5.8: (A) - shows the true cluster 1 labels with noise; (B) - shows cluster 1 labels assigned by DBSCAN with Cosine Similarity metric algorithm.



(A) (B)

FIGURE 5.9: (A) - shows the true cluster 3 labels with noise; (B) - cluster 3 labels assigned by DBSCAN with Cosine Similarity metric algorithm.

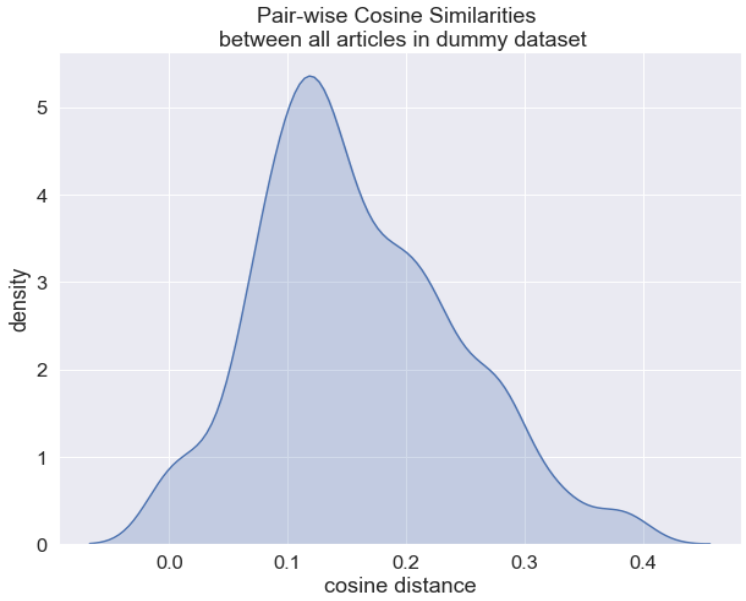


FIGURE 5.10: The distribution of all pair-wise Cosine Similarities between all the articles in the dummy dataset. The distributions are plotted using KDE with a Gaussian kernel and bandwidth calculated with Silverman's rule of thumb.

5.3.3 Dynamic Determination of Inter-Cluster Distances Range

After selecting the best distance metric that is used to measure the proximities between the articles in the dataset, we further investigate the properties of the event articles (within the clusters) and the noise articles using the dummy dataset. The main objective is to find a method that could dynamically determine the distance between the cluster articles (i.e. density of the event clusters) and the noise solely from the data without the need to manually set the density parameter (*eps*) for DBSCAN clustering model.

As observed earlier (see the Figures 5.4, 5.5, 5.6), the data contains two distinct distributions of proximities: 1) distances within the event clusters (inter-cluster proximities); 2) distances between the articles outside the clusters (proximities between cluster articles and noise articles or between noise articles only). Therefore, in order to dynamically estimate the *eps* parameter for DBSCAN algorithm (which defines the inter-cluster distances between the event articles) directly from the data, we use Gaussian Mixture Model. We implement the below steps:

- First, we calculate the pair-wise cosine similarities between all the articles in the dummy dataset. The Figure 5.10 shows the distribution of the similarities between the articles. The distribution is plotted using the kernel density estimate (KDE), with a Gaussian kernel and bandwidth calculated with Silverman's rule of thumb. The distribution mean $\mu = 0.16$ and standard deviation $\sigma = 0.08$.

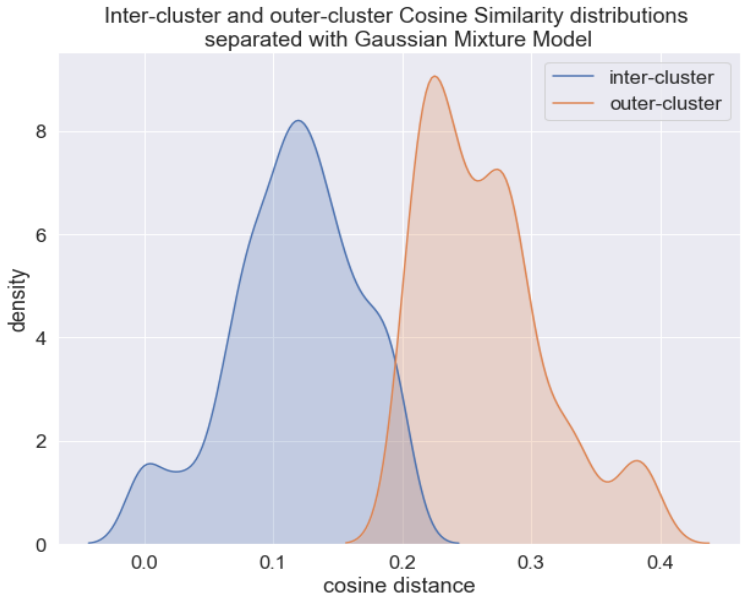


FIGURE 5.11: Resulting two distributions: 1) inter-cluster proximities (blue); 2) outside cluster proximities: cluster-noise or noise-noise proximity (orange); after applying Gaussian Mixture Model to separate the distribution of cosine similarities as presented in the Figure 5.10. The distributions are plotted using KDE with a Gaussian kernel and bandwidth calculated with Silverman's rule of thumb.

- We assume that the Cosine Similarities between the articles in the dataset are drawn from the two normal distributions: 1) inter-cluster proximities 2) outside cluster proximities: cluster-noise or noise-noise proximity. Based on this assumption, we use Gaussian Mixture Model to separate the Cosine Similarities between all the articles (as shown in the Figure 5.10) into two distributions. The resulting two distributions are shown in the Figure 5.11. The first inter-cluster proximities distribution (blue) has a mean of $\mu = 0.12$ and the standard deviation of $\sigma = 0.08$. This similar to the earlier results of the mean inter-cluster cosine similarities of 0.16 (refer to the Figure 5.4). The second distribution of outside cluster proximities (orange) has a mean of $\mu = 0.26$ and the standard deviation of $\sigma = 0.04$. There is a statistically significant difference between these two distributions according to Wilcoxon signed-rank test: V = 486.6, p = 0.0, α = 0.05. This shows that the Gaussian Mixture Model could be a good way to assess if the dataset contains two distributions and estimate the range of inter-cluster distances.

- Next, using the properties of the inter-cluster distribution, we derive an *eps* range that will be subsequently used for the DBSCAN parameter search. The lower and upper bounds of the *eps* range are defined as per equations 5.13, 5.14 where μ and σ are the mean and standard deviation of the distribution (Figure 5.11 (blue)). The lower bound is defined as 1 standard deviation from the mean as we do not want clusters to be very small containing only very similar articles. Finally, for this example, the eps range that will be used for DBSCAN

is $eps \in [0.12 - 0.08, 0.12 + 1.5 \times 0.08]$. The Table 5.3 provides a summary of how the eps range derived using Gaussian Mixture Model changes if the dataset contains: a) event articles clusters together noise articles; b) only noise articles. It can be observed that the average *eps* value increases significantly for the data that contains only noise articles which shows that in DBSCAN would not be able to find dense clusters in the noise only dataset.

$$eps_{lower} = \mu - \sigma \tag{5.13}$$

$$eps_{upper} = \mu + 1.5\sigma \tag{5.14}$$

Data	μ	σ	eps range
Event Clusters with Noise	0.12	0.08	$eps \in [0.04, 0.24]$
Noise only	0.19	0.04	$eps \in [0.15, 0.25]$

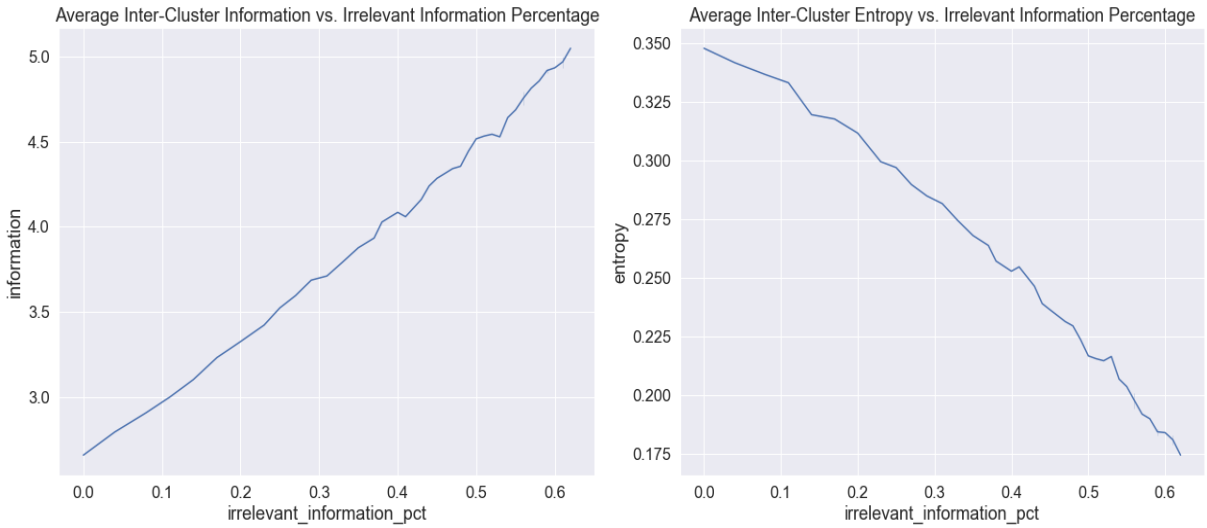
TABLE 5.3: A summary of how eps range derived using Gausian Mixture Model changes if the dataset contains: a) event articles clusters together noise articles; b) only noise articles.

5.3.4 DBSCAN Parameters Search

After finding the *eps* range for the given dataset, the DBSCAN parameters search is implemented. The goal is to cluster the articles and achieve *pure* event clusters that do not contain irrelevant information (non-event related articles that do not belong to the cluster). We use Information content and Information Entropy metrics to evaluate the *purity* of event clusters. Unlike in thermodynamics where by introducing more noise i.e. increasing temperature of a system, the Entropy increases due to an increase in velocity (random activity) of particles, hereby introducing more irrelevant information (non-event articles) into a cluster (i.e. a system), the Entropy decreases. This is due to the fact that word frequencies of irrelevant information (non-event news articles) in a cluster are low in comparison to frequencies of words describing an event. The Figure 5.12 shows that by introducing more irrelevant articles into an event cluster (a system), the average Information content increases (A) whilst the Entropy decreases (B). Therefore, in this experiment, we assume that *pure* event clusters would have high Information Entropy. Hence, the objective is to find the optimum DBSCAN parameter configuration (*eps* and *minPts*) that maximises the average inter-cluster Information Entropy.

The average cluster Information content and Entropy are calculated using the following steps:

1. A cluster corpus consists of all article titles in the cluster. A vocabulary (V) of words is generated from the corpus. We count the number of times each word in the vocabulary occurs across the corpus - n^{th} word frequencies (f_{w_n}).



(A) (B)

FIGURE 5.12: (A) - shows the relationship between the average inter-cluster Information content and the inter-cluster irrelevant information (non-event related articles) percentage; (B) - shows the relationship between the average inter-cluster Entropy and the inter-cluster irrelevant information (non-event related articles) percentage.

2. The probability of each word (P_{w_i}) in the corpus is defined as per below equation 5.15, where f_{w_n} - is a total occurrence of a word w_n across the corpus, N_{corpus} - is a total number of words in the corpus.

$$P_{w_n} = \frac{f_{w_n}}{N_{corpus}} \tag{5.15}$$

3. The normalized Information content (H_{t_k}) for an article title (t_k) in the cluster is calculated using the equation 5.16, where $P_{w_{nj}}$ is a probability of the j^{th} word (w_{nj}) in the title and J - total words in the title.

$$H_{t_k} = -\frac{1}{J}\sum_{j=1}^{J} log_2(P_{w_{nj}}) \tag{5.16}$$

4. The average Information content for a cluster is defined as per equation 5.17, where K - total number of titles in the cluster)

$$H_{cluster} = \frac{1}{K} H_{t_k} \tag{5.17}$$

5. The normalized Entropy (E_{t_k}) for an article title (t_k) in the cluster is calculated using the equation 5.18.

$$E_{t_k} = -\frac{1}{J}\sum_{j=1}^{J} P_{w_{nj}} \times log_2(P_{w_{nj}}) \tag{5.18}$$

6. The average Entropy for a cluster is defined as per equation 5.19.

$$E_{cluster} = \frac{1}{K} E_{t_k} \tag{5.19}$$

After defining how the average Information content and Entropy metrics are calculated for the cluster, we describe the process of the DBSCAN parameter search using the dummy dataset.

First, the parameter grid is defined over which the search is done - a range of eps values $eps \in [0.04, 0.24]$ as found by Gaussian Mixture Model and minimum points in the cluster range $minPts \in [4, 20]$. Second, for each parameter configuration, the articles are clustered using DBSCAN and the following properties of the clustered data are measured:

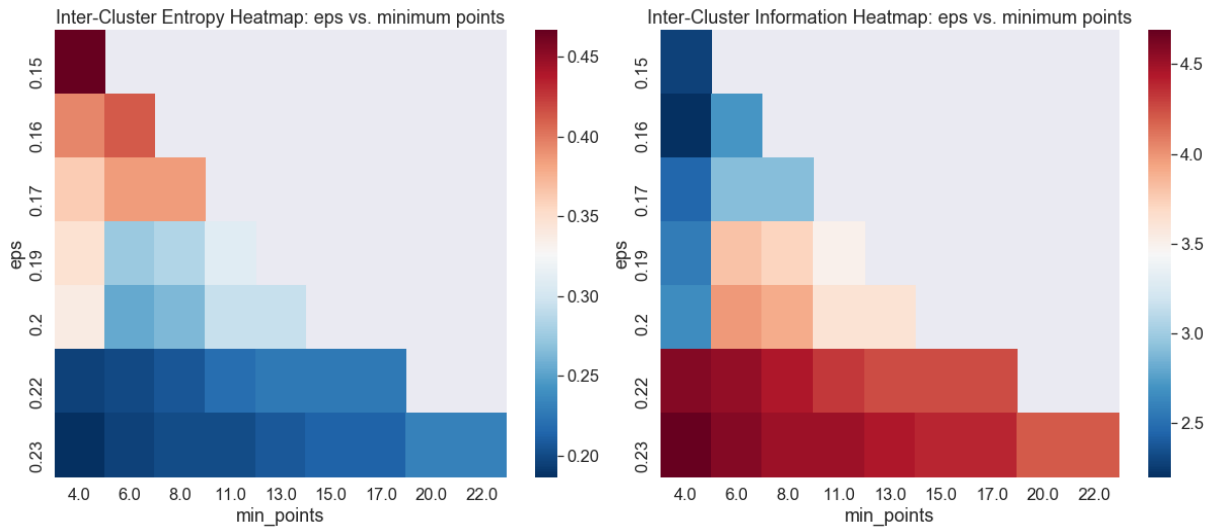
- Accuracy.
- Average inter-cluster Entropy.
- Average inter-cluster Information content.
- Average inter-cluster article correlation.
- Average maximum proximity between the articles in the cluster.
- Average cluster size (number of articles in the cluster).
- Clustered articles ratio (articles in the cluster out of total articles in the dataset).

The objective of the parameter search is to find the eps and $minPts$ values that maximize the average inter-cluster Entropy. For the dummy dataset, it can be observed from the Figure 5.13 that the maximum Entropy is achieved at $eps = 0.15$ and $minPts = 4$ and minimum Information content - $eps = 0.16$ and $minPts = 4$. Meanwhile the highest accuracy is achieved at $eps = 0.15$ and $minPts = 4$ according to the Figure 5.14. Therefore, in this case the optimum DBSCAN parameters for clustering the dummy dataset are $eps = 0.15$ and $minPts = 4$.

In addition, we analyse how the cluster sizes (Figure 5.15 A), maximum proximity between the articles in the cluster (Figure 5.15 B) and ratio of number of articles within the cluster and noise (irrelevant information) (Figure 5.16) change with different values of eps. As expected, all three properties increase with the increasing eps due to more noise being introduced within the clusters.

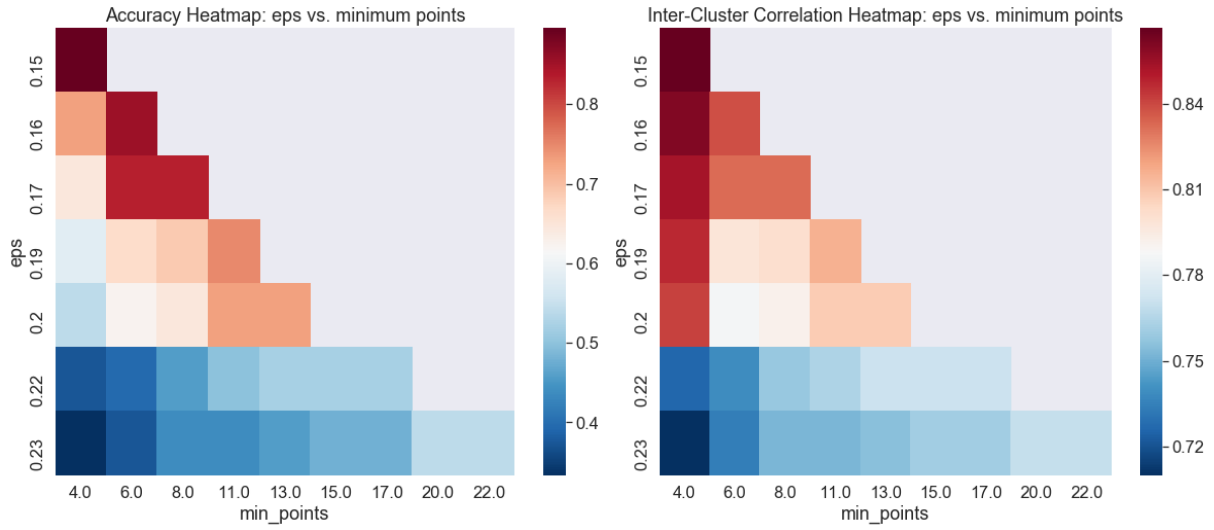
5.4 Dynamic Density-based News Clustering

In this section, the final framework for the news events clustering is presented. The Dynamic Density-based News Clustering framework utilizes a density-based clustering method - DBSCAN where the cluster density (eps) and size ($minPts$) parameters are dynamically determined from the dataset. The framework is defined in the



(A) (B)

FIGURE 5.13: (A) Inter-cluster Entropy heatmap, the maximum Entropy is achieved at $eps = 0.15$ and $minPts = 4$; (B) - Inter-cluster Information content, the minimum values achieved at $eps = 0.16$ and $minPts = 4$.



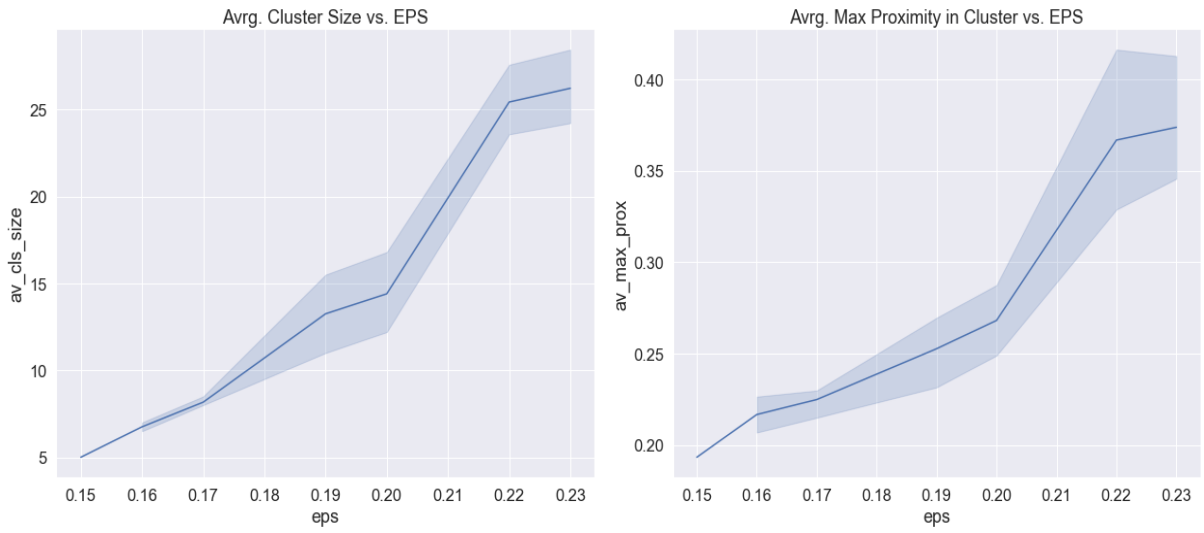
(A) (B)

FIGURE 5.14: (A) Clustering accuracy heatmap, the maximum accuracy is achieved at $eps = 0.15$ and $minPts = 4$; (B) - Inter-cluster pairwise correlation between the articles, the maximum values achieved at $eps = 0.15$ and $minPts = 4$.

Algorithm 1. Below is a description of the Dynamic Density-based News Clustering framework (Algorithm 1):

1. The news headlines about a company published during a period (e.g. a day) are used as an input (X_{period}).

2. The headlines are cleaned and then embedded using Word2Vec embeddings.



(A) (B)

FIGURE 5.15: (A) - shows the relationship between the average cluster size and eps; (B) - shows the relationship between the average maximum proximity between the articles in the cluster and eps.

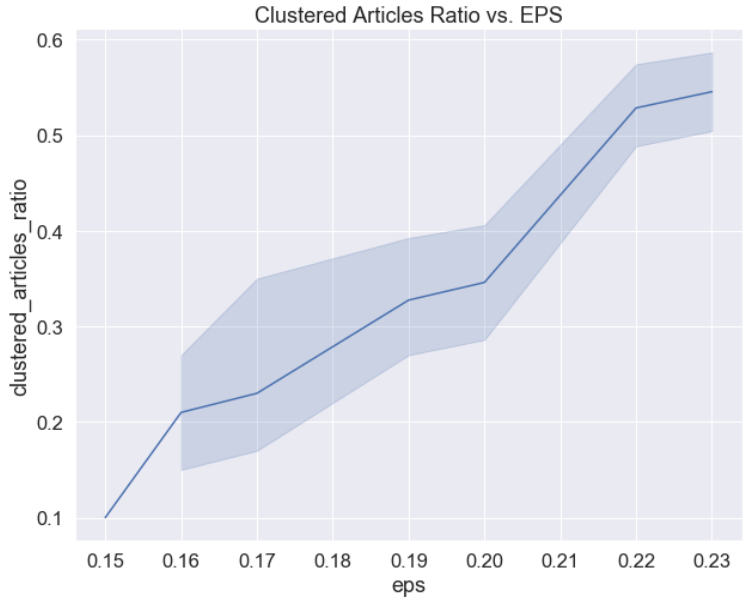


FIGURE 5.16: The relationship between a ratio of clustered articles to total articles in the dataset and eps.

3. The pair-wise cosine similarities are calculated between all the headlines. For each title, the top 5 most similar titles are selected and used for further analysis to determine the eps_{range}.

4. Using Gaussian Mixture Model, the pair-wise similarities between the data points are separated into two distributions that are assumed to be - inter-cluster similarities and noise. The eps_{range} is estimated from the inter-cluster similarity distribution (see the equations 5.13, 5.14).

5. The eps_{range} (cluster density parameters) and $minPts_{range} \in [4, 20]$ (cluster size parameters) ranges are used for the DBSCAN parameter grid search. Here, with each combination of the parameters, all the headlines (X_{period}) are clustered and the average inter-cluster Entropy is measured. The optimum set of parameters (eps', $minPts'$) are selected by finding the parameter configuration that generated clusters with the maximum inter-cluster Entropy.

6. Finally, the titles are clustered with DBSCAN using the optimum parameters (eps', $minPts'$) and the clusters of the events are returned.

Algorithm 1 : Dynamic Density-based News Clustering

Input: All company's news headlines published during a period (X_{period})
Output: Events headlines (X_{events}) (if any)

procedure NEWSEVENTS(X_{period})
  for $all\ x\ in\ X_{daily}$ **do** ▷ Create embeddings.
    $Word2Vec(x)$
  $CosineSimilarity(X'_{daily}) \rightarrow sim_{top5}$ ▷ Top 5 similar headlines.
  $GaussianMixture(sim) \rightarrow eps_{range}$ ▷ Get eps range.
  $minPts_{range} \in [4, 20]$ ▷ Predefined cluster size.
  for eps in eps_{range} **do** ▷ Parameter grid search.
    for $minPts$ in $minPts_{range}$ **do**
      $DBSCAN(eps, minPts) \rightarrow (Entropy, eps, minPts)$
      $ParamList.append((Entropy, eps, minPts))$
  $MAX(ParamList) \rightarrow (eps', minPts')$ ▷ Get final parameters.
  $DBSCAN(eps', minPts') \rightarrow X_{events}$ ▷ Final clustering.

5.5 Results

Finally, all the news articles in the dataset (177,309) are clustered using the Dynamic Density-based News Clustering framework (Algorithm 1). In addition, to analyse the performance of the framework, the articles are also clustered using DBSCAN with non-dynamic parameters (without the grid search) where $eps = 0.12$ and $minPts = 8$ for all cases. The Figure 5.17 shows Nike's events timeline generated using the proposed dynamic clustering framework. In total there are 80 events associated with Nike found during the period between July 2017 and July 2018. In contrast, the Figure 5.18 shows the events found in the same dataset without the grid search, with the static parameter values - $eps = 0.12$ and $minPts = 8$. This method extracts only 20 events during the same period.

The Figures 5.19 and 5.20 show a summary of all events found across different companies in the dataset using both methods: 1) proposed Dynamic Density-based News Clustering framework (Algorithm 1) - Figure 5.19; and 2) DBSCAN with non-dynamic parameters ($eps = 0.12$ and $minPts = 8$) - Figure 5.20. The Dynamic Density-based News Clustering framework with varying DBSCAN parameters (eps, $minPts$) optimized for maximum inter-cluster Entropy, is able to identify

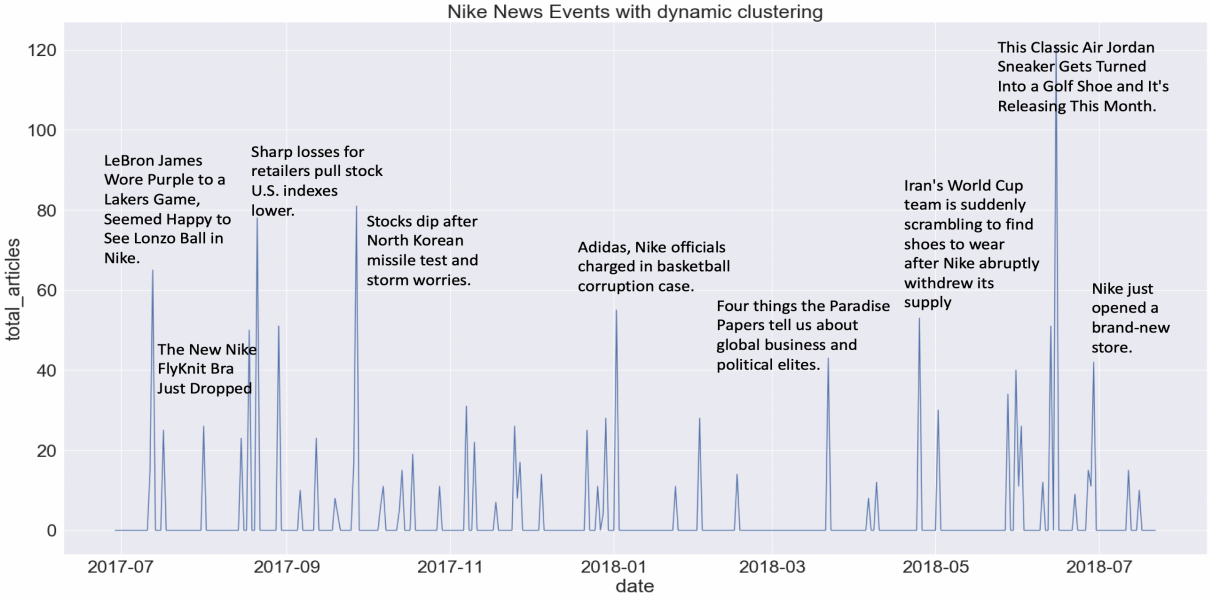


FIGURE 5.17: Nike news events generated using Dynamic Density-based News Clustering framework (Algorithm 1). The framework identified 80 events in total during 1 year period.

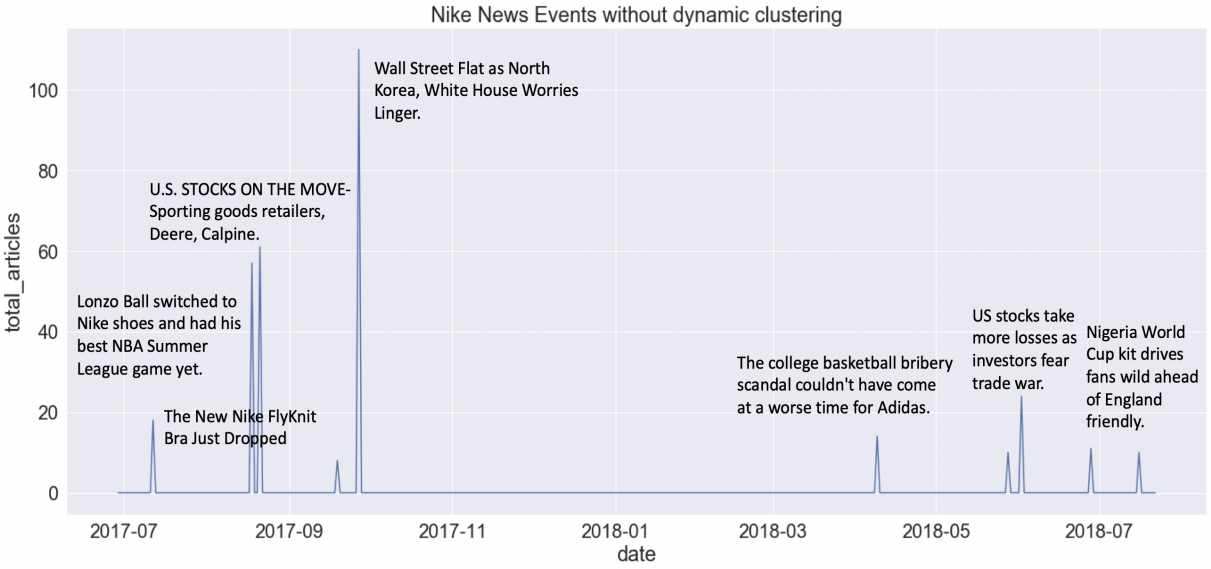


FIGURE 5.18: Nike news events generated using DBSCAN clustering with non-dynamic parameters ($eps = 0.12$ and $minPts = 8$). The framework identified only 15 events in total during 1 year period.

more events with clusters of various sizes and densities. This is particularly important when clustering news articles about different companies. For example, more popular companies with larger market capitalization (i.e. Nike, Adidas, Burberry, LVMH, etc.) tend to be mentioned in the news more often. Consequently, their articles have different volume levels and frequencies and therefore different DBSCAN parameters - cluster densities and sizes (eps, $minPts$). Hence, the parameters should be dynamic and tuned accordingly to the company. On the other hand, there might

be still some events not identified by the method. The only way to measure this is by reading all the news articles in the dataset.

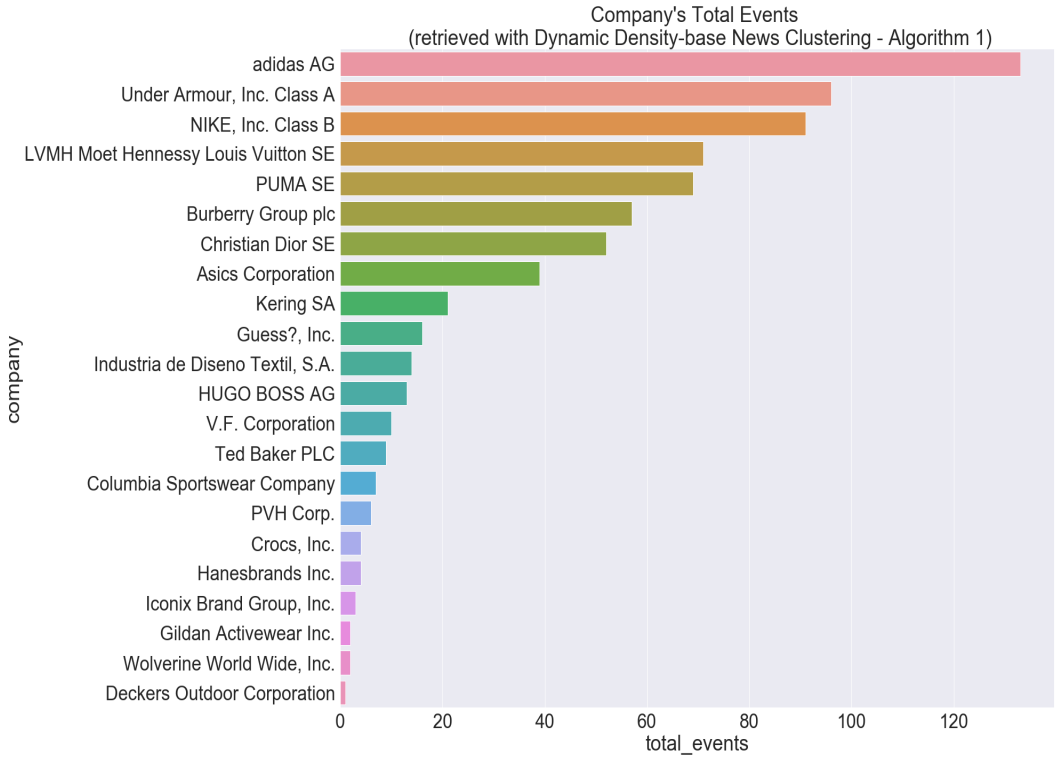


FIGURE 5.19: Companies' news events generated using the Dynamic Density-based News Clustering framework (Algorithm 1), during the period between April 2017 and August 2018.

In addition, from the observations we find that the upper bound for the inter-cluster events Cosine Similarity is 0.2, i.e. most commonly the articles with the pair-wise cosine similarities above 0.2 are noise. Also, the lower bound of the average inter-cluster Entropy is 0.35, as the clusters that have lower average Entropy than 0.35 usually contain noise articles.

5.6 Conclusion

In this chapter, we analyse the characteristics of the company-related events reported in the news. The aim is to design the unsupervised framework that is able to extract events from the noisy news data without the predefined model parameters or expert knowledge. The Dynamic Density-based News Clustering Framework (see Algorithm 1) is proposed as a method to cluster the company's articles. The framework incorporates DBSCAN clustering model where its parameters (eps and $minPts$) are determined dynamically using a grid search method and finding a set of values that maximises the inter-cluster Entropy. We show that the proposed framework is better at finding events within the data than using static DBSCAN parameters for all companies or time periods. The framework is able to extract more events from the

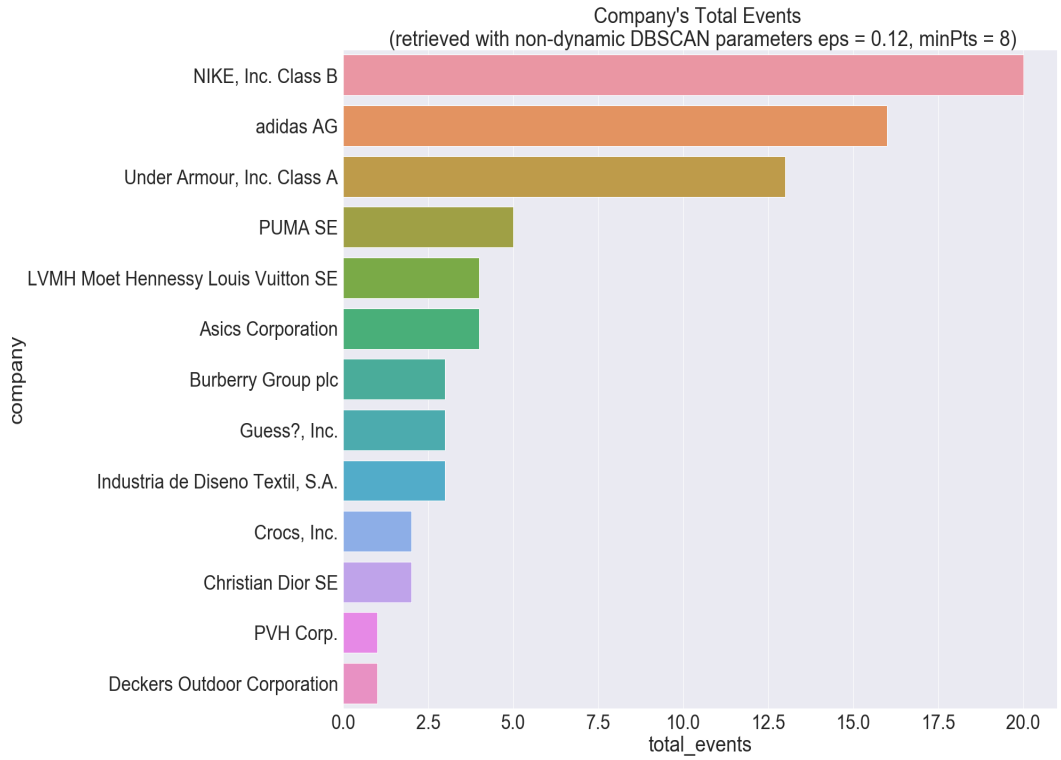


FIGURE 5.20: Companies' news events generated using DBSCAN clustering with non-dynamic parameters ($eps = 0.12$ and $minPts = 8$), during the period between April 2017 and August 2018.

data as the sizes and densities of event clusters are dynamic - they are dependant on a company and a time period.

Chapter 6

ALGA: Automatic Logic Gate Annotator for Event Detection

This chapter discusses an original automatic data labelling framework called ALGA - Automatic Logic Gate Annotator, proposed by us. The framework helps to create large amounts of annotated data for training domain-specific financial news events detection classifiers at scale. ALGA's framework implements a rules-based approach to annotate the training dataset. This allows an easier transferability to other domains and better interpretability of models trained on automatically labelled data. Unlike traditional data annotation methods, ALGA helps to filter the relevant news articles from the noise better. Hence, the models are able to achieve state-of-the-art performance in domain-specific financial events detection. Using the training data generated by ALGA's framework, we build the Apparel/Footwear industry news events detection model focused on identifying events that are likely to impact a company's stock price. The experiment is done in collaboration with Arabesque Asset Management.

6.1 Problem Overview

A company's stock price is mostly driven by its fundamentals and the release of new information. The market-related news might change investors' expectations and impact the asset price (Tan, 2019). The news can either be related to the macro events such as economic, political, sector-specific events or company-specific events that are directly or indirectly (e.g. news about company's suppliers, competitors or clients) linked to a company. These news-worthy events are shown to be associated with a company's stock price changes (Ding, 2014). Therefore, an ability to automatically identify which news contain financially relevant events and predict their impact may give an additional insight into a company's future financial performance and potentially yield better investment returns.

There are two main research directions that analyze news impact on stock prices: event and sentiment-driven (Tetlock, 2007). Most of the research work in this area is focused on building predictive models that use news sentiment or events mentioned in the articles to forecast stock price movement (Ding, 2014; Akita, 2016; Ding, 2015; Huynh, 2017; Ding, 2016; Liu, 2018a; Merello, 2018; Li, 2018). Another research

area is aimed at building models that detect financial events in the news (Han, 2018; Yang, 2018). In this experiment, we focus on the event detection problem and present a framework for identifying financially relevant events in the news (events that are likely to have a future impact on a company's stock price).

We argue that in order to build better event detection models they have to be tuned to the domain-specific language as the event importance across different business sectors varies. For example, the news articles about *'successful clinical trials'* would have more impact on the companies operating in the Pharmaceutical sector whereas news about *'store closures'* would be more significant for the companies in Retail Trade or Consumer Goods sectors. However, there is a lack of research in domain-specific event identification methods mainly due to the absence of annotated corpus for training supervised Machine Learning algorithms (Han, 2018). In the previous studies, a training corpus for the Supervised Learning financial event detection task was usually created by either human annotators or labelling articles based on the stock price movements. Both methods have associated challenges. Annotators introduce human-bias, data is time-consuming to label and therefore expensive to obtain. Meanwhile, labelling articles according to the stock price movements alone is ambiguous due to the fact that both price data and news articles are extremely noisy. For example, if 100 articles about a company are released across multiple news sources during a single day and meantime the company's share price dips significantly, using the traditional approach adopted by most of the studies, all 160 articles would be labelled as *'event'* articles even though only 60 articles are talking about a financially relevant event and the remaining articles are irrelevant - noise. The Figure 6.1 shows an example of such scenario. The challenge remains to create a labelled training dataset for the domain-specific event detection task at scale cheaply and build classifiers that can distinguish between the financially significant news and noise.

This experiment aims to fill in the current research gaps by the following:

- Design a new framework for the systematic news articles labelling that is able to differentiate relevant articles from the noise. Unlike studies (Han, 2018; Yang, 2018), we focus on English news articles.
- As we take a systematic approach to create the training dataset, the output from the final model can be easier interpretable.
- We design a domain-specific news event classifier with a focus on a single financial industry.
- Implementation of the state-of-the-art BERT encoder to embed the news articles. The embeddings should be able to generalise better for unseen words and domain-specific language (fine-tuned BERT).

The main objective of this work is to build a domain-specific event classifier that is able to filter through the noisy articles and detect the news stories that are likely

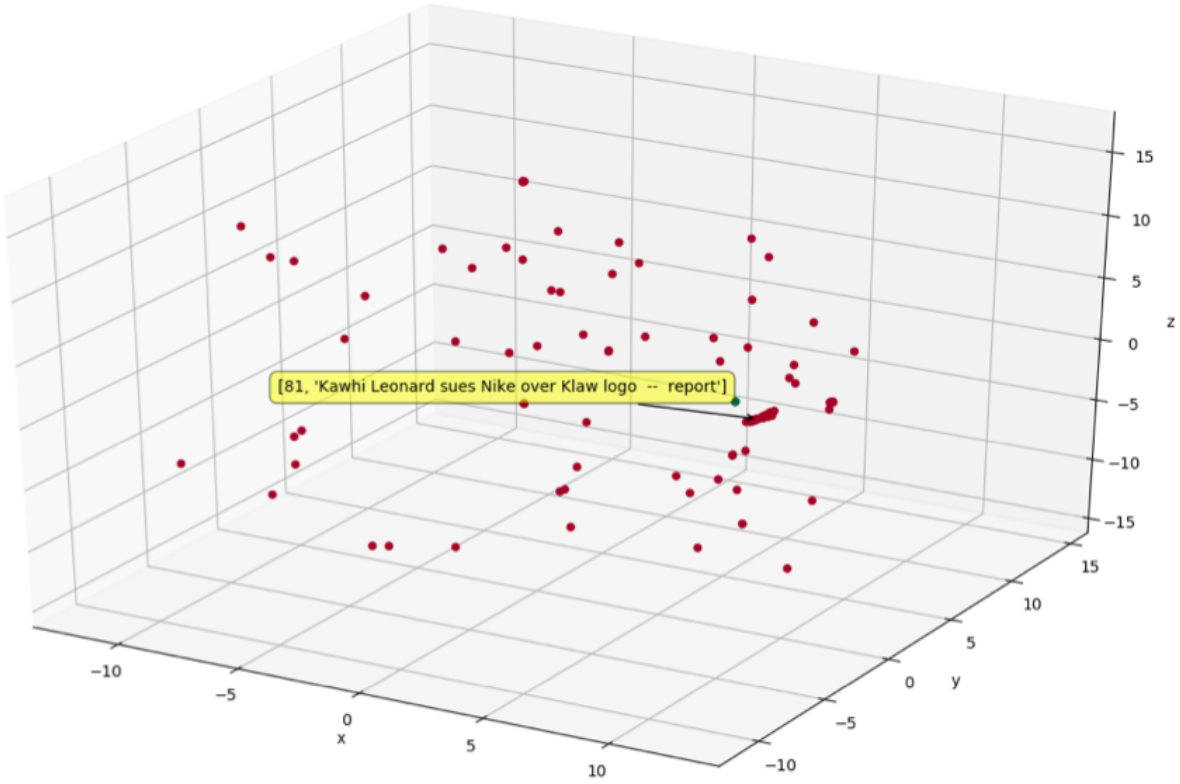


FIGURE 6.1: An example of noisy news data. Traditionally the training datasets for news event detection models are created either by human annotators or labelling articles based on the stock price movements. Both methods have the following challenges: 1) annotators introduce human-bias, data is time-consuming to label, expensive and not scalable; 2) labelling articles according to the stock price movements is ambiguous as returns and news articles are noisy. The plot shows 160 news articles about Nike (sportswear company) released on the 9th July 2019. The financially relevant news event that moved Nike's stock price on that day was associated with a lawsuit over Nike ('Kawhi Leonard sues Nike over Klaw logo.'), the remaining news articles are irrelevant information. The axes in the plot correspond to the three principal components of a news article vector. Originally each article is represented by 768-dimensional vector and PCA is used to reduce the vector dimensions for visualization purpose only.

to impact a company's stock price. Therefore, to solve the problems above, first, we design the Automatic Logic Gate Annotator (ALGA) framework that uses a systematic approach to label the training data, second, we use the labelled data to train the event detection classifier. We design a domain-specific event detection model targeted at Apparel/Footwear industry news. ALGA's framework contains modular combinations of logic gates that help to filter the noisy data and automatically separate articles into *event* and *non-event* news. Specifically, the logic gates impose conditions on an article that have to be met in order for an article to be annotated with an *event* label otherwise it is assigned with a *non-event* label. The conditions are

combinations of the following article features: 1) topic popularity; 2) daily volume of articles; 3) relevancy; 4) is an article part of a cluster i.e. are there more similar articles; 5) does an article contain event synonyms from the pre-defined dictionary; and the features derived from company's stock prices and its sector index: 1) stock price volatility; 2) does the difference between sector index and stock price returns exceeding a pre-defined threshold. For this research work we use headlines as representations of the articles and we embed them using the following methods: BERT encoder (Devlin, 2018), Doc2Vec (Le, 2014) and TF-IDF (Sparck Jones, 1972). The Figure 6.2 shows the overall architecture of ALGA's framework. By implementing the automatic labelling techniques, we can achieve state-of-the-art results in domain-specific financial event detection task.

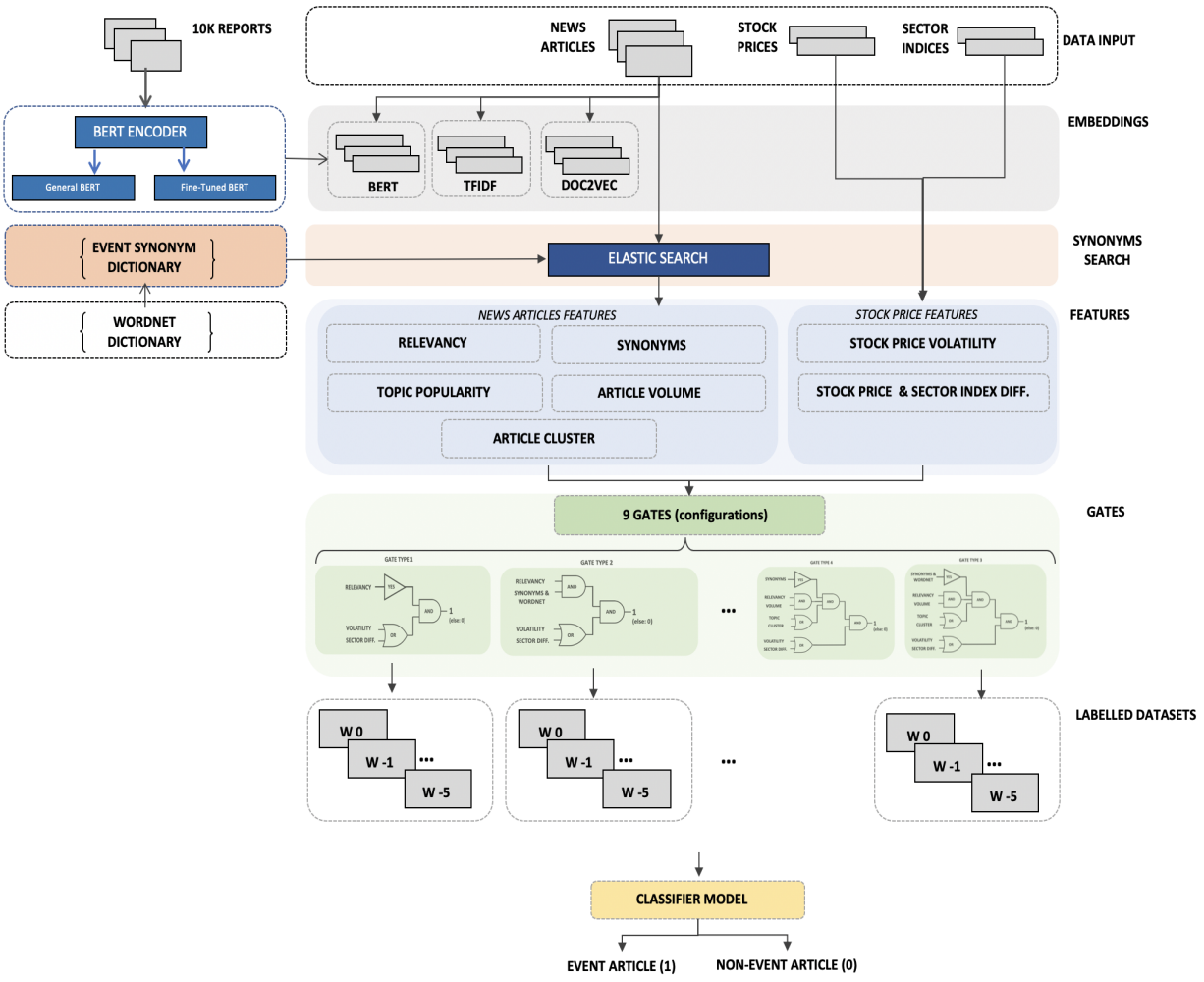


FIGURE 6.2: ALGA: Automatic Logic Gate Annotator Framework. We propose a scalable and transferable systematic method to create a large amount of labelled data for the domain-specific event detection modelling. ALGA's framework contains modular combinations of logic gates that help to filter the noisy data and automatically separate articles into *event* and *non-event* news.

To summarize, the contributions of this research work are:

- We present ALGA - Automatic Logic Gate Annotator framework that can automatically generate large amounts of annotated data at scale by separating news articles into financially relevant events and noise.
- Using the training data generated by the ALGA's framework, we train the financial news event detection classifier and show the state-of-the-art results in domain-specific event detection problem;
- Behaviour analysis of the different text embedding methods: BERT, Doc2Vec, TF-IDF, we compare the behaviour of fine-tuned BERT embeddings (tuned on financial text corpus - 10K reports) against the general BERT.
- ALGA's framework demonstrates scalability, transferability. The behaviour of the classifier trained using such dataset can be easier interpretable.

The content of this chapter is structured as follows. First, we give an overview of the data and methodology used during this study in Section 2. Sections 3 and 4 present the architecture of ALGA's framework and an overview of the event classifier model. The research results are described in Section 6. Section 7 is a conclusion of the work where we describe how the model can be used in practice and discuss areas for further work.

6.2 Dataset

In order to build a domain-specific event detection model for this study, we choose publicly listed U.S. companies that belong to a single industry - Apparel/Footwear. The datasets used during this study include:

- Stock prices of 42 U.S. companies operating in Apparel/Footwear industry across 2 sectors - Retail Trade and Consumer Non-Durables.
- U.S. Retail Trade and Consumer Non-Durables sector indices.
- Business related news articles that mention the 42 companies. The articles are obtained from the EventRegistry[1] and by scraping online fashion news, blogs and magazines (see more details about the datasets in Chapter 3). In total, we obtain 488,000 articles.
- Form 10-K reports, in total 2,947 reports. We use the reports to fine-tune BERT encoder on the financial text corpus.

All the above data is collected for the period between January 2014 and September 2019.

[1]EventRegistry - a news aggregator service (*https://www.eventregistry.org*).

6.3 Methodology

In this section, we give an overview of the methodology implemented during this study. The main motivation behind the research work is to build a model that classifies the news articles into *financially relevant* and *noise*. Here we define a *financially relevant* article as a news event that is likely to have an impact on a company's stock price. For example, we aim to identify the articles mentioning such events as *CEO departure*, *bribery scandal*, *factory disaster*, etc. and capture their relationship with a company's returns. Our objective is to build a classifier that takes news articles as an input and outputs binary value 1 - *financially relevant event*, 0 - *noise*. The key steps of the method are:

1. *ALGA: Automatic Logic Gate Annotator* - we create a rules-based approach to annotate the news articles using Automatic Logic Gate Annotator (ALGA) framework. As part of the framework, we also create a domain-specific event dictionary that contains synonyms associated with the important events within the Apparel/Footwear industry.

2. *Domain Specific Event Classifier* - after creating the labelled training dataset, we train logistic regression classifier that is able to distinguish the event articles from the noisy irrelevant information.

3. *Results* - we evaluate the performance of the classifier and analyse the behaviour of different article embeddings.

Below we describe the article representations (embeddings), evaluation metrics together with an overview of the algorithms used for the article topic modelling (Latent Dirichlet Allocation) and clustering (DBSCAN).

6.3.1 Article Representations

Similarly, as in the second experiment presented in Chapter 4, we use an article title as a representation of the entire article (Li, 2015). We encode the headlines using three different methods: BERT, Doc2Vec and TF-IDF.

BERT

Words used in the financial text such as *bull* or *bear* are not associated with animals in the same way as pre-trained general language word embeddings are (c.f. Mikolov, 2013) (Merello, 2018). Hence, it is important to introduce a domain-specific language representation into the model. To solve this problem, we utilize the state-of-the-art pre-trained language representation model BERT (Bidirectional Encoder Representations from Transformers) (Devlin, 2018). Unlike the previous standard unidirectional language models (e.g. OpenAI GPT (Radford, 2018)), BERT is a multi-layer bidirectional Transformer encoder that utilizes the original implementation of

the Transformer architecture and bidirectional self-attention (Vaswani, 2017). The most recent advancement in transfer learning from language models is to pre-train a model on a language model objective and then to fine-tune the model on the task-specific text corpus (Devlin, 2018). Therefore, prior to using the BERT encoder, we fine-tune the model using the financial text corpus - Form 10K reports. We assume that the fine-tuned encoder should be able to generalize better for domain-specific language in this case finance. Two types of BERT encoders are used for further analysis - the general BERT (originally trained on the general language corpus) and the fine-tuned BERT. We sum the last 4 layers from the encoder output across all the input words (i.e. an article headline) and use the result as the representation for each title.

Doc2Vec

is an Unsupervised Learning algorithm for learning text representations of varied length (i.e. sentences, paragraphs or documents) (Le, 2014). The framework is a Neural Network based model motivated by the method used to learn word embeddings (Mikolov, 2013). The objective is to learn the document vector representations by predicting the surrounding words in the contexts that are sampled from the documents. We train the Doc2Vec (Le, 2014) representations for the news headlines using the corpus of all titles in the news dataset.

TF-IDF

is a statistical measure that evaluates the importance of each word in the document collection (Sparck Jones, 1972). It is a simple and commonly used method to represent the text in a vector format. The main weakness of the TF-IDF method is that the words order in the sentences and the word semantics are ignored. We use the TF-IDF to encode all the article titles in the dataset.

6.3.2 Evaluation Metrics

To evaluate the classification models, we use the following metrics: accuracy, F1-score and ROC curve together with AUC measure.

Accuracy

The classification accuracy is defined as per equation (6.1).

$$Accuracy(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} 1(\hat{y}_i = y_1) \tag{6.1}$$

F1-Score

is defined by the equation (6.2). It is interpreted as a weighted average of the Precision (6.3) and Recall (6.4), where both metrics are defined as per below.

The classification Precision is an evaluation metric which describes the ability of a classifier not to assign the positive label to the negative sample. Another metric - classification Recall is defined as the ability of a classifier to find all the positive samples. Here TP is true positives, FP - false positives, FN - false negative, TN - true negatives.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{6.2}$$

$$Precision = \frac{TP}{TP + FP} \tag{6.3}$$

$$Recall = \frac{TP}{TP + FN} \tag{6.4}$$

AUC - ROC Curve

The most common metric to evaluate the performance of a binary classifier is the ROC curve (Receiver Operating Characteristics) together with associated AUC (Area Under the Curve) metric. The ROC is a probability curve and AUC is an area under this curve (a measure of separability). They show how well the model is able to distinguish between the different classes. The higher AUC value - the better model performance. The ROC curve is a plot of the True Positive Rate (Recall) against the False Positive Rate (FPR) (see the equation 6.5).

$$FPR = \frac{FP}{TN + FP} \tag{6.5}$$

6.3.3 Latent Dirichlet Allocation

The Latent Dirichlet Allocation (LDA) is a generative probabilistic model (Blei, 2003) often used for topic extraction. Here, the choice of Dirichlet distribution is motivated by Bayesian inference. The method assumes that there are K topics ($\beta_{1:k}$) across the document corpus where each topic is defined as a distribution over a fixed vocabulary (β_k). In addition, it is also assumed that a document exhibits multiple topics in different proportions and each word in a document is drawn from one of the topics where the topic is chosen from the per-document distribution over the topics (Blei, 2012). The main objective is to infer the hidden topic structure from the observed documents. Using a joint probability distribution over the observed and hidden variables, the posterior distribution is computed for the structure of the hidden topics given the observed words in the documents. The LDA is defined as the joint

distribution (6.6) where θ_d is the topic proportions for document d and $\theta_{d,k}$ - topic k proportion in the document d, the topic assignment for the document d is z_d, where $z_{d,n}$ is the topic assignment for the n^{th} in d. The observed words in the document d - w_d and $w_{d,n}$ the n^{th} word in d (word from the corpus vocabulary). The posterior is defined as per equation (6.7).

$$p(\beta_{1:k}, \theta_{1:D}, z_{1:D}, w_{1:D}) = \prod_{i=1}^{K} p(\beta_i) \prod_{d=1}^{D} p(\theta_d) \left(\prod_{n=1}^{N} p(z_{d,n}|\theta_d) p(w_{d,n}|\beta_{1:K}, z_{d,n}) \right) \tag{6.6}$$

$$p(\beta_{1:k}, \theta_{1:D}, z_{1:D}|w_{1:D}) = \frac{p(\beta_{1:k}, \theta_{1:D}, z_{1:D}, w_{1:D})}{p(w_{1:D})} \tag{6.7}$$

6.3.4 DBSCAN

The Density Based Spatial Clustering of Applications with Noise (DBSCAN) is a density-based clustering algorithm designed to identify clusters of various shapes (not just convex-shaped) without a need of the domain knowledge to pre-define the number of clusters as an input parameter (c.f. K-Means clustering algorithm) (Ester, 1996). The algorithm finds clusters by separating the high-density areas from the noise - low-density areas based on two input parameters: 1) eps - radius and 2) $minPts$ - a minimum number of points within the cluster, a threshold. A number of data points within a predefined radius (eps) is counted to calculate a point's density within the dataset (Heidari, 2019). Based on the density value, every point in the dataset is classified into a core point, a border point or a noise point.

6.3.5 Fashion Finance Dictionary

Similarly, how the authors of the work (Merello, 2018) used L. McDonald's financial domain dictionary Loughran, 2011, we build a domain-specific event dictionary - *Fashion Finance Dictionary*, with a focus on the apparel and footwear industry events. In order to build this dictionary, LDA and DBSCAN are used to retrieve the topics across the articles and cluster the articles. We implement the article clustering using BERT embeddings. Using the cluster analysis and topics we manually select the event-related synonyms across the news articles. In total, we identify 17 event topics together with a dictionary of 320 synonyms related to these events. The Figure 6.3 presents a snapshot of the events synonyms from the dictionary.

6.4 ALGA: Automatic Logic Gate Annotator Architecture

This section describes Automatic Logic Gate Annotator architecture in more detail and shows how the training data is created. The full framework is represented in

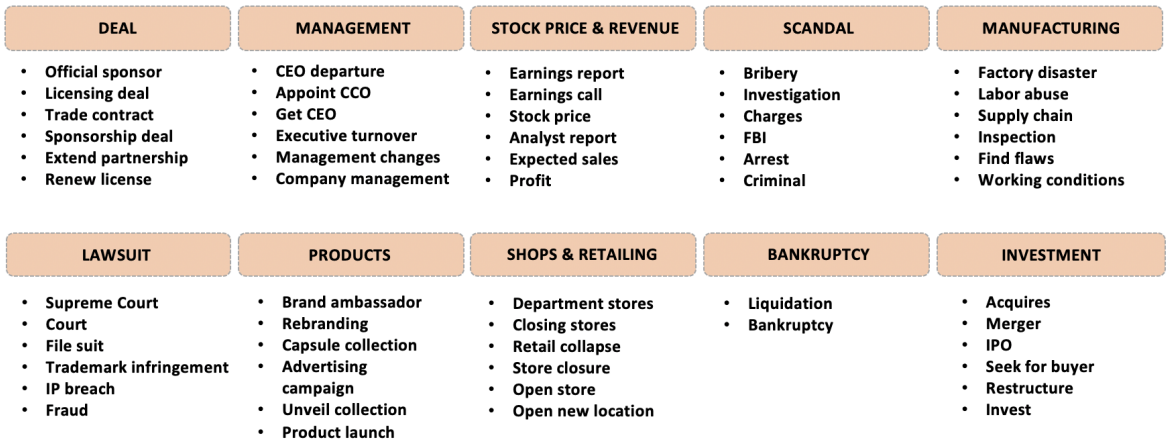


FIGURE 6.3: *Fashion Finance Dictionary* - a domain-specific event synonyms dictionary for the Apparel/Footwear industry. As part of the study, we design a domain-specific event synonyms dictionary for the Apparel and Footwear industry. We extract the topics (LDA), cluster the articles (DBSCAN) and use the analysis to manually select the event-related synonyms across the news articles. We identify 17 event topics together with a dictionary of 320 synonyms related to these events. The dictionary is used as part of ALGA's data labelling framework.

the Figure 6.2.

Data Input

For the data labelling task, the framework takes 3 types of data - news articles, stock prices of the 42 assets mentioned in the news articles and 2 sector indices (US Retail Trade and US Consumer Non-durables). The news articles are cleaned using common text preprocessing techniques (Feuerriegel, 2016).

Embeddings

the news headlines are embedded using 4 different types of embedding methods: fine-tuned BERT encoder, general BERT encoder (original version), Doc2Vec and TF-IDF. Two sets of the article headlines are embedded - the original headlines and headlines with the masked company names. In total, there are 8 variations of the embedded news titles.

Synonyms Search

we upload the news articles into the ElasticSearch (text search engine) in order to identify which article headlines contain words from our predefined event dictionary. We search for exact word match as defined in the dictionary and also their synonym words utilizing WordNet English synonyms dictionary (Miller, 1995). This allows

us to find the articles that mention events related to our 17 event topics from the dictionary.

Features

Using the news articles, their corresponding embeddings and financial data, the following features are created:

- *Relevancy* - we consider the article to be relevant if a company's name is mentioned in the headline and in the main article body.
- *Synonyms* - identify whether the article headline contains words from the event synonyms dictionary or similar words (WordNet).
- *Topic popularity* - LDA algorithm is used to extract topics from all the articles. An article is considered to contain a popular topic if it talks about a *trending* topic on that day. We define *trending* as per the following - if there are 50 articles about a company published during one day and 35 articles are talking about the same topic, all articles within this group are considered as *trending*.
- *Article volume* - we count a total number of articles published about a company each day. If the volume of articles during one day exceeds a pre-defined threshold - all the articles on that day are considered as volume articles.
- *Article cluster* - DBSCAN is used to cluster the articles if an article falls within a cluster it is referred as a cluster article which means that there are more similar articles talking about related event/topic.
- *Stock price volatility* - we consider a company's stock price to be volatile if a standard deviation of the returns for 3 or 7-day periods is higher than a pre-defined threshold. These periods are considered as *stock price events*.
- *Stock price and sector index difference* - we identify the periods where a company's returns are significantly different (by a pre-defined threshold) from its sector index returns. This helps to find the periods where a company's stock price has different behaviour from the other companies within the sector.

After creating the above features for the news articles and financial data, two datasets are combined based on the date value. 5 different datasets are created using time shift where the financial data is shifted by the following windows: 0, -1, -2, -3 and -5 days. The intuition behind the time windows is as follows: if an article about *financially relevant* event is published today can we see an impact on a company's returns on the same day, after 1 day (-1), after two days (-2), after three days (-3) or after 5 days (-5).

Gates

We design the logic gates system that helps to filter the articles based on the features created above and created annotated training dataset. In total, we create 9 different types of logic gates configurations. The Figure 6.4 shows all 9 types of logic gates. For example, if we consider Type 2 Gate, when passing an article through this gate the article needs to meet the following conditions in order to be labelled as an *event* article: 1) the article has to be relevant - a company's name appears in the title and the article body AND the article has to contain an event synonym from the dictionary; 2) the article has to be published during the period when the company's returns were volatile OR there was a significant difference between the company's returns and the sector index returns. Both 1) AND 2) conditions has to be met for an article to be labelled as an *event* article (1), otherwise the article is labelled as *non-event* (0). The same logic is applied across all the gate types. The output is 9 types of different training datasets, as we want to analyse how different labelling conditions impact the final classification model accuracy.

The Gate Type 0 and the Gate Type 1 are the most commonly used methods to label the data in the current literature that are mostly reliant on stock price volatility and article relevance (i.e. if a company's name is mentioned in the title). The main difference of our data annotation approach is that we label the articles on more advanced features such as sector index difference, topics, related articles (clusters) and events synonyms.

To summarize, using ALGA's framework we generate a number of training datasets that we use for training classification models. The datasets contain news articles with binary labels where 1 is *financially relevant* event and 0 - *financially irrelevant* event. The logic gates filter the noisy data. It can be observed from the Figure 6.5 that the more restrictive gate is the fewer articles pass through it hence reducing the total number of positive samples. The Figures 6.6 and 6.7 represents the prevailing event topics in the datasets after passing the articles through the Gate Type 2 and Gate Type 8. As the Gate Type 8 is more restrictive there is more variation within the topics across different time periods - more noise is filtered.

6.5 Domain-Specific Event Classifier

Using the training datasets generated by ALGA's framework, the event classification models are built. We implement logistic regression algorithm to train the models. Considering the training set $\mathcal{D}$ with m training samples $\mathcal{D} = \{(x_i, y_i)|i = 1 : m\}$, where $x_i \in \mathbb{R}^{n+1}$ and $y \in \{0, 1\}$, the logistic regression is defined as per equation (6.8) where the cost function (6.9) is used to minimise the loss during the training to find the optimal model parameters (Murphy, 2012).

$$h_\theta(x_i) = \frac{1}{1 + \exp(-\theta x_i)} \tag{6.8}$$

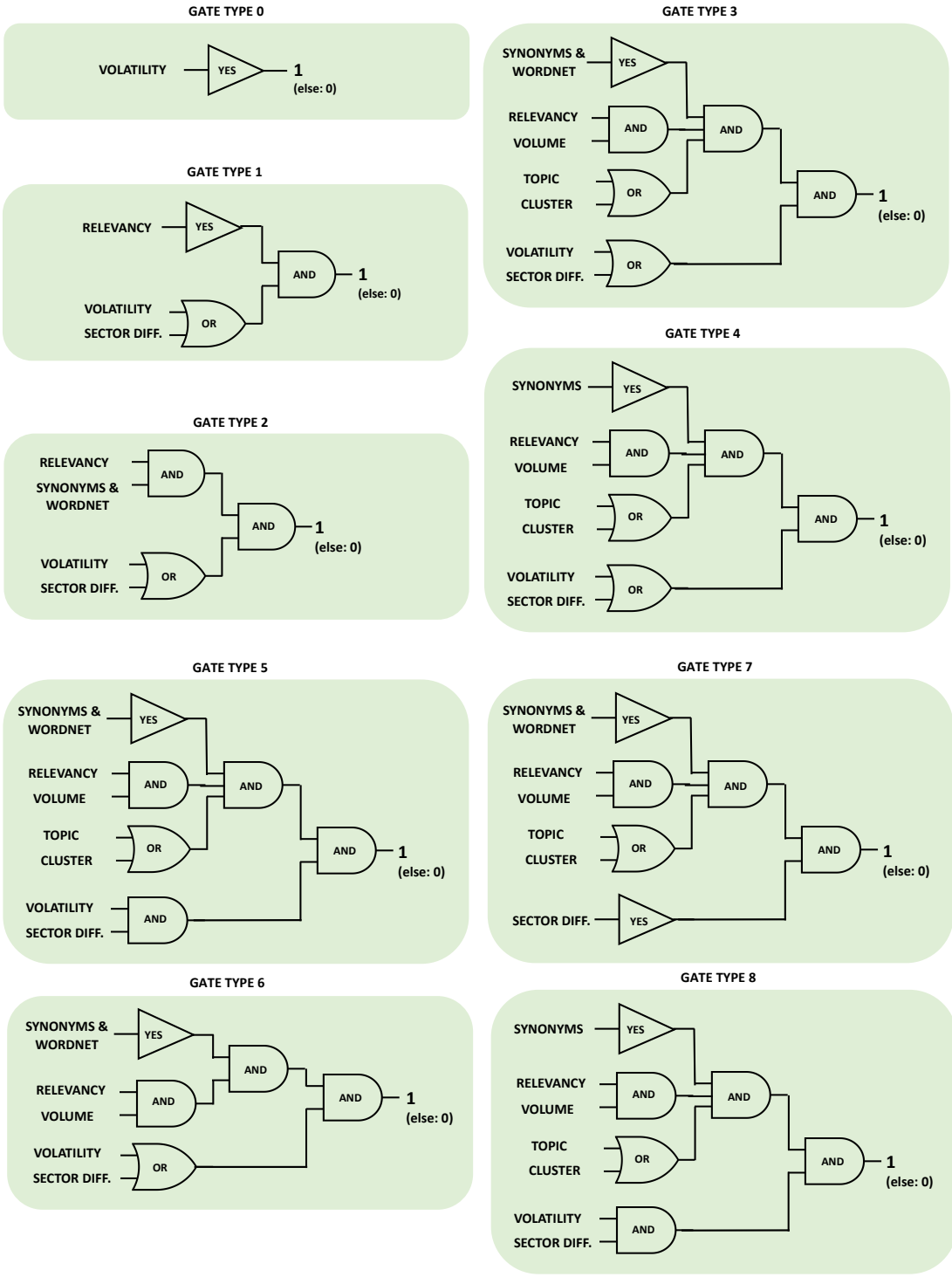


FIGURE 6.4: Logic gates systems - 9 types of different logic gates configurations that are used to labels the articles. The logic gates are part of ALGA's framework. They are used to impose a set of conditions on the articles for them to be labelled as event articles (1) or noise (0) during the annotation process.

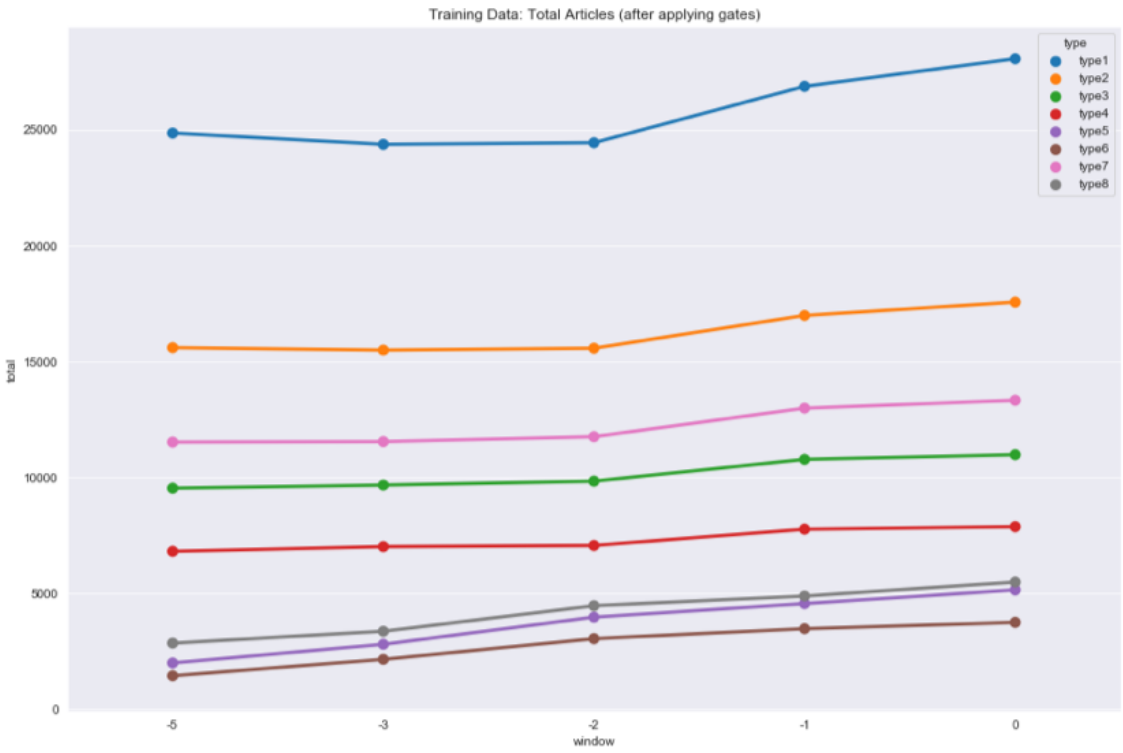


FIGURE 6.5: The size of generated training datasets populated by different types of gates.

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m} y_i log(h_\theta(x_i)) + (1 - y_i) log(1 - h_\theta(x_i)) \tag{6.9}$$

6.6 Results

To analyse the performance of the trained models using different training sets together with different embeddings - accuracy, F-1 score, ROC curves are used. In addition, as a benchmark, we also use random labelling approach. We randomly label the news articles and train classifiers on this dataset to compare it against other models.

Below we summarize the key results and insights from this study:

- The model trained on the dataset produced by the Gate Type 8, (the gate with the most conditions) has the highest accuracy. The Figures 6.8 and 6.9 show how the model accuracy varies across different gate types. The models in the figures are trained on the headlines where the companies names are masked. The Gate Type 0 and Gate Type 1 are the most common approaches in the literature to label the data - where the news articles about a company are labelled purely based on its stock price volatility (Gate Type 0) and in some cases considering company's name mention in a headline (Gate Type 1). It can be

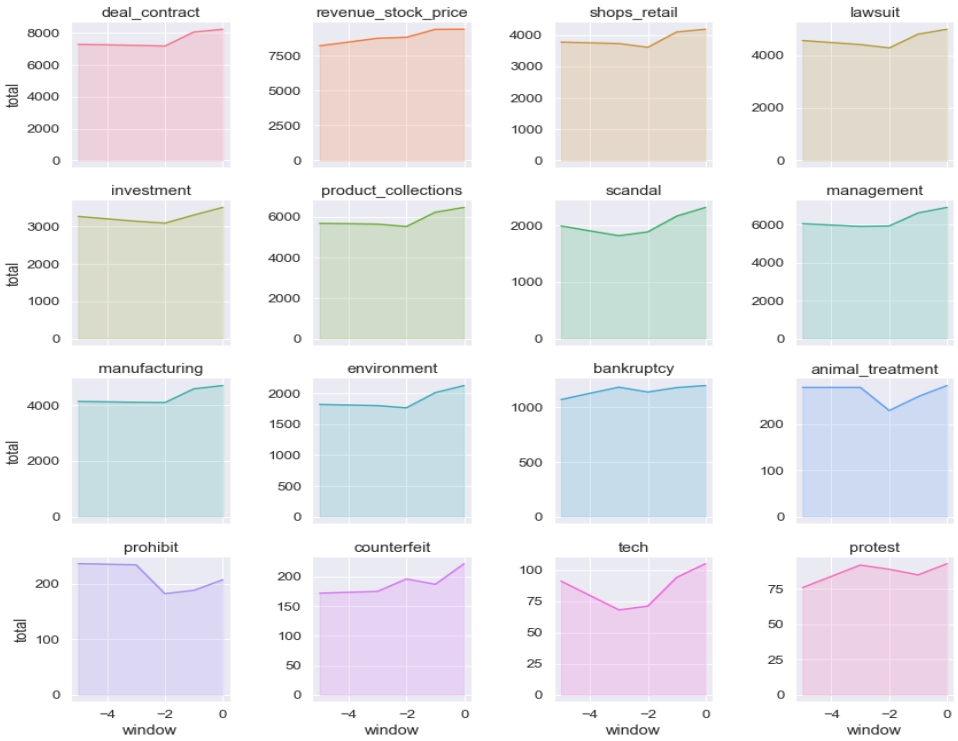


FIGURE 6.6: Topics generated by Gate Type 2 across windows [0,-5].

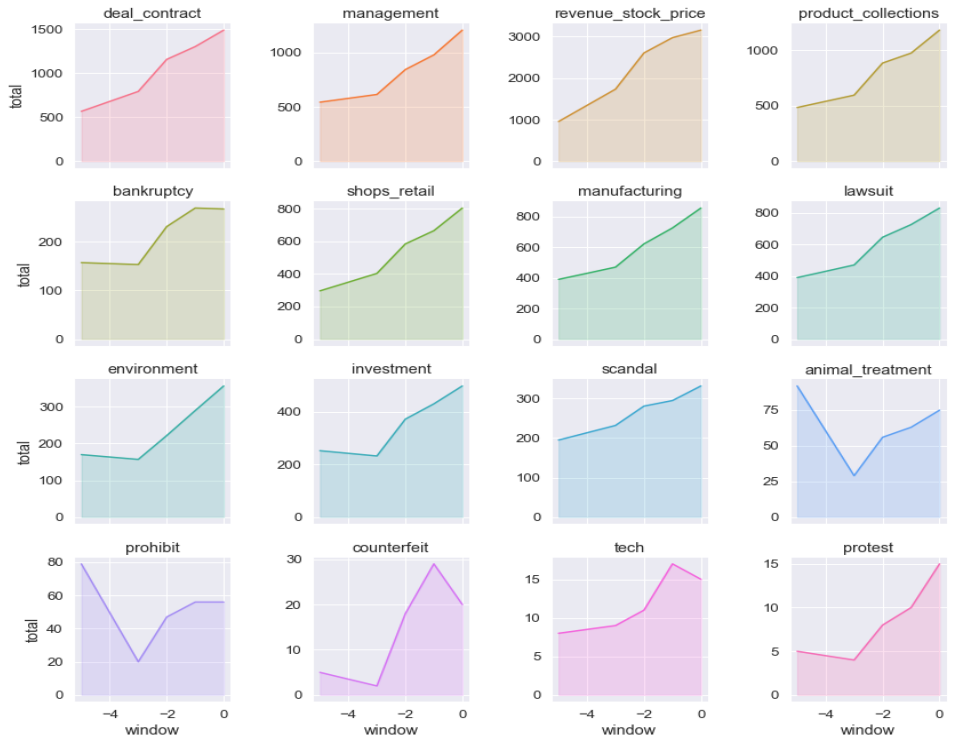


FIGURE 6.7: Topics generated by Gate Type 8 across windows [0,-5].

observed that the accuracy of both models that use the common labelling approach is between 0.5 and 0.6 which is closer to the random labelling method. Meanwhile, the Gate Type 4 and Gate Type 8 produce the highest accuracy around 0.85. Both of these gates have the most constraints imposed on the news articles during the labelling (see the Figure 6.4), this shows that by implementing stricter noise filtering techniques we can achieve better classification accuracy. The F-1 scores of the models show a similar trend (see the Figure 6.10).

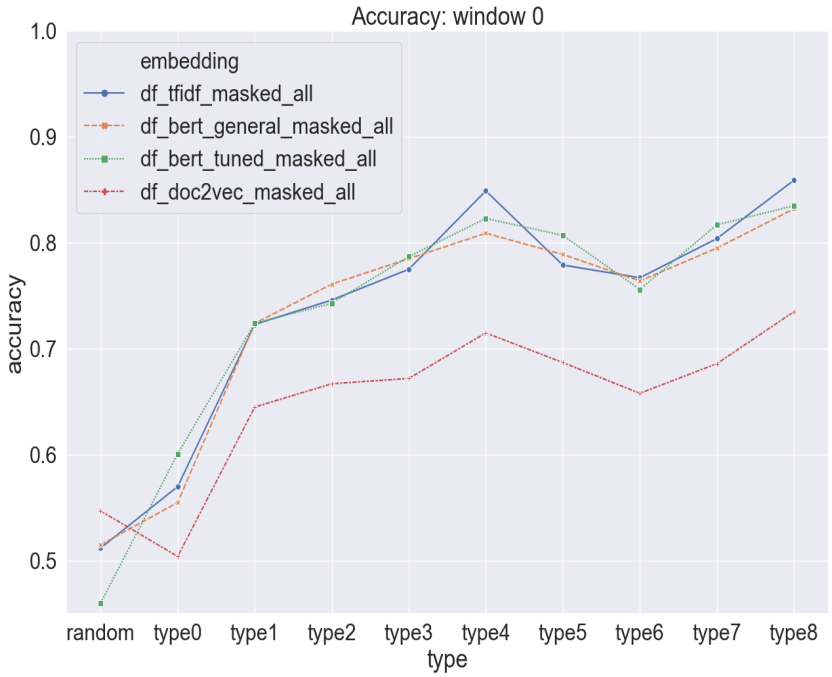


FIGURE 6.8: Accuracies of models trained on different types of datasets produced by ALGA's framework at time window 0. The accuracy of models varies across different gate types. The model trained on the dataset produced by the Gate Type 8, (the gate with the most conditions) has the highest accuracy.

- The models trained on the datasets produced by the Gate Type 4, Gate Type 5, Gate Type 8 are better at separating between two classes - achieving AUC above 0.88. The Figure 6.11 shows ROC curves for models trained on tuned BERT embeddings with masked company names across windows -1 and -5. Meanwhile, the model trained on the data generated by the Gate Type 0 achieves AUC value of approximately 0.55.

- The following three embeddings: tuned-BERT, general-BERT and TF-IDF performed similarly. The accuracy of the models trained using these embeddings are similar across all Gate Types, whilst the Doc2Vec - noticeably lower (see the Figures 6.8 and 6.9).

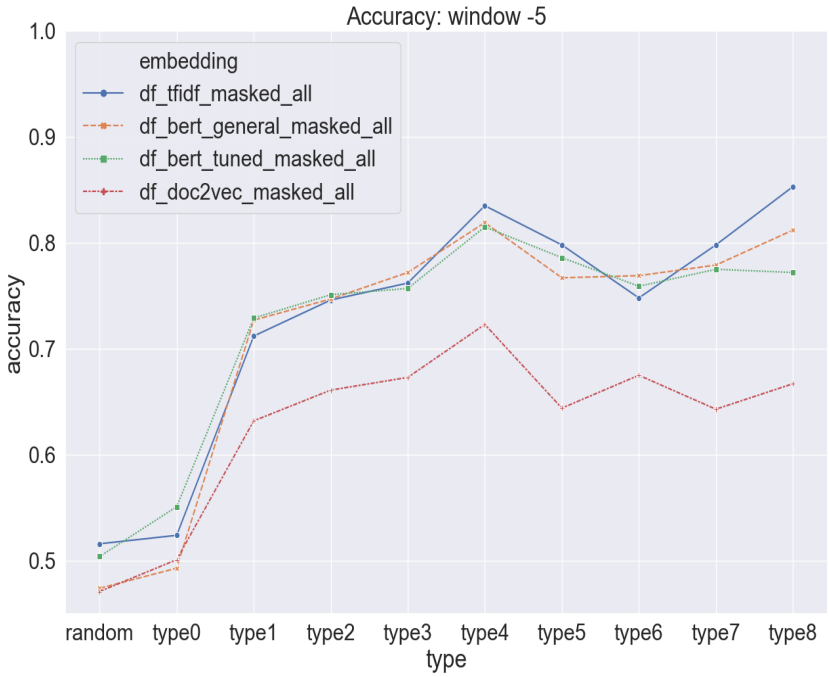


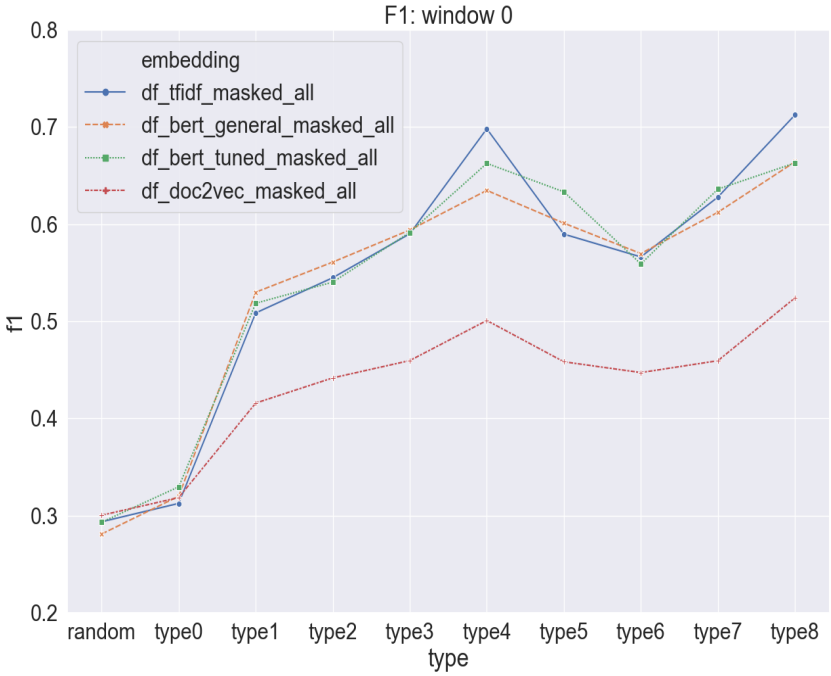
FIGURE 6.9: Accuracies of models trained on different types of datasets produced by ALGA's framework at time window -5. The accuracy of models varies across different gate types. The model trained on the dataset produced by the Gate Type 8, (the gate with the most conditions) has the highest accuracy.

- The models trained on the (masked company's name) TF-IDF embeddings show a better performance in differentiating between the classes achieving the AUC value of above 0.91 (see the Figure 6.12). In comparison, both BERT embeddings (tuned and general) are fluctuating around a similar AUC value of 0.9.

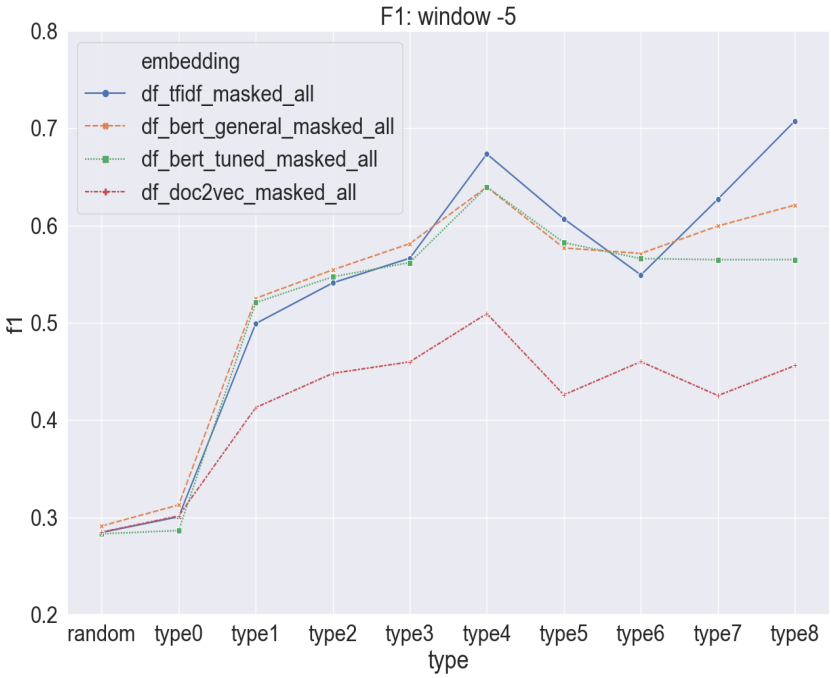
6.7 Conclusion

In this study, we present the original work in building the domain-specific financial news events detection model. We introduce a new scalable and transferable automatic data labelling framework - ALGA (**A**utomatic **L**ogic **G**ate **A**nnotator). The framework is able to systematically label the training data for the event detection classification task. For this study, the ALGA's framework is designed to annotate the news from a single Apparel/Footwear industry although the method can be easily transferable to other domains/industries by using different input data and domain-specific event dictionary. We show the state-of-the-art performance of the event detection model trained on the data produced by ALGA. As expected, by implementing more article filtering conditions to eliminate noise, better event classification models can be build as we find that the event classifier trained on the data

produced by the most restrictive gate (Gate Type 8) achieves the highest accuracy of 0.85. In future work, we will apply the framework to other industries and train different types of classification models such as Deep Neural Networks with attention layers. In addition, as future work, the configuration of the gates could be optimally selected (e.g. using evolutionary dynamics) by learning the gate structure where the objective is to maximize the performance of the final classification model.

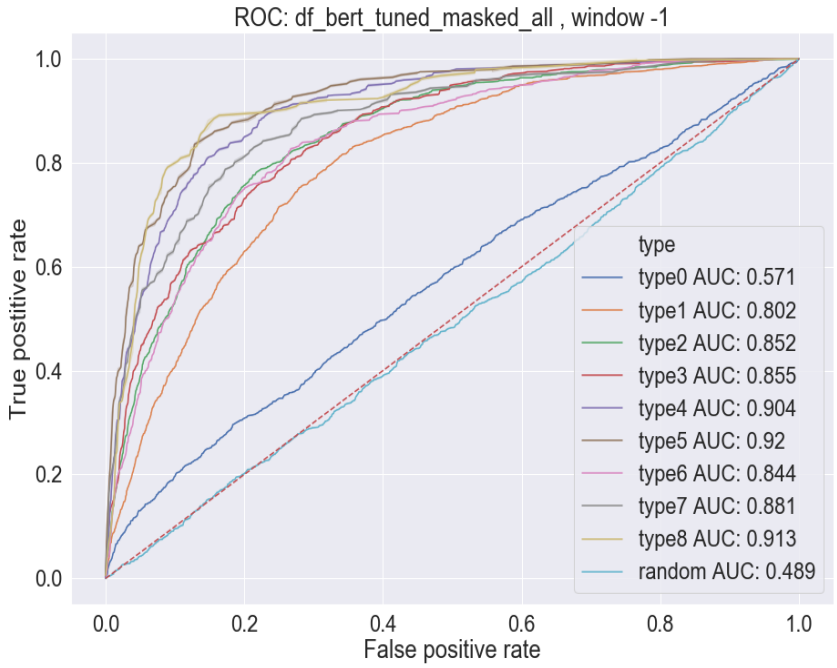


(A)

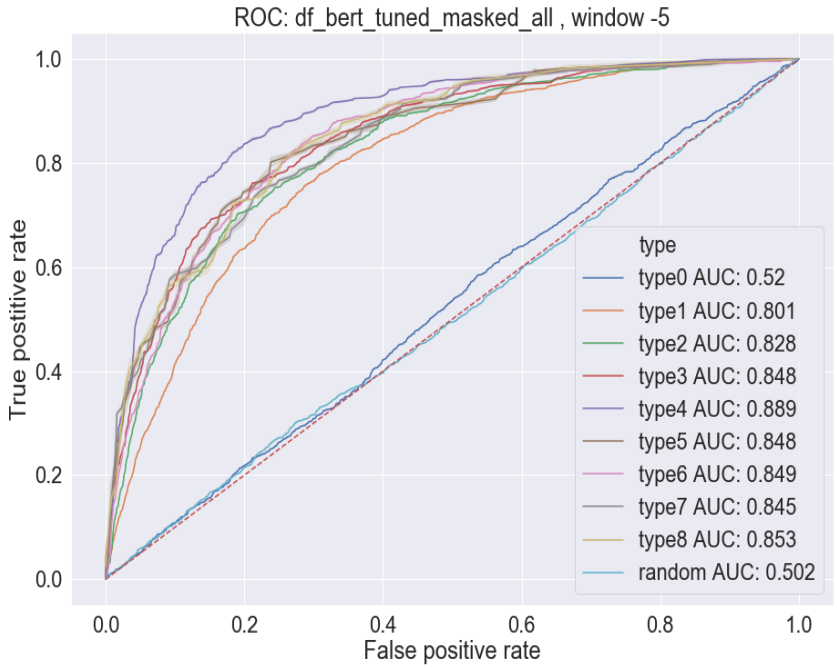


(B)

FIGURE 6.10: F1-scores of model trained on different types of datasets produced by ALGA's framework at time windows 0 and -5. The F1-scores of models varies across different gate types. Similarly as in the Figures 6.8 and 6.9, the model trained on the dataset produced by the Gate Type 8, (the gate with the most conditions) has the highest F1-score.

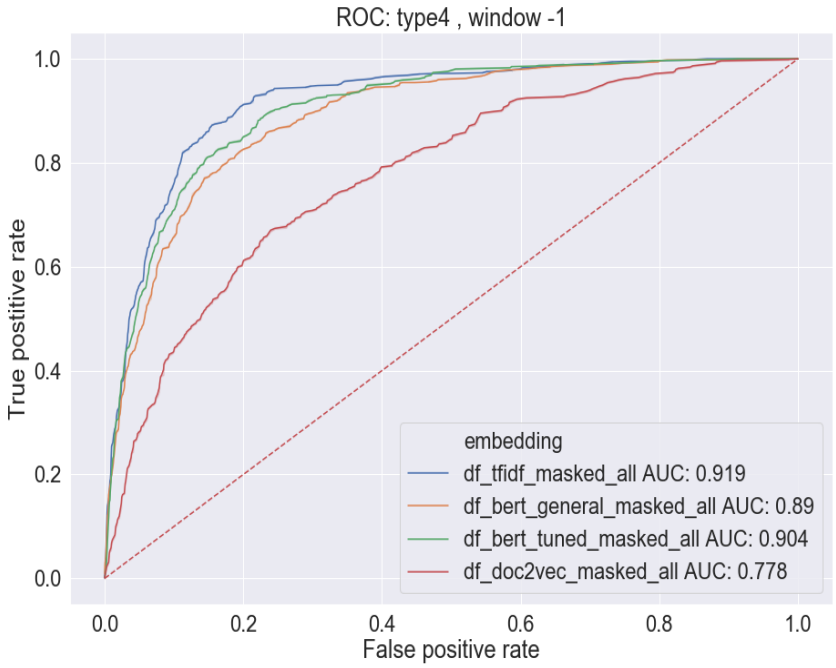


(A)

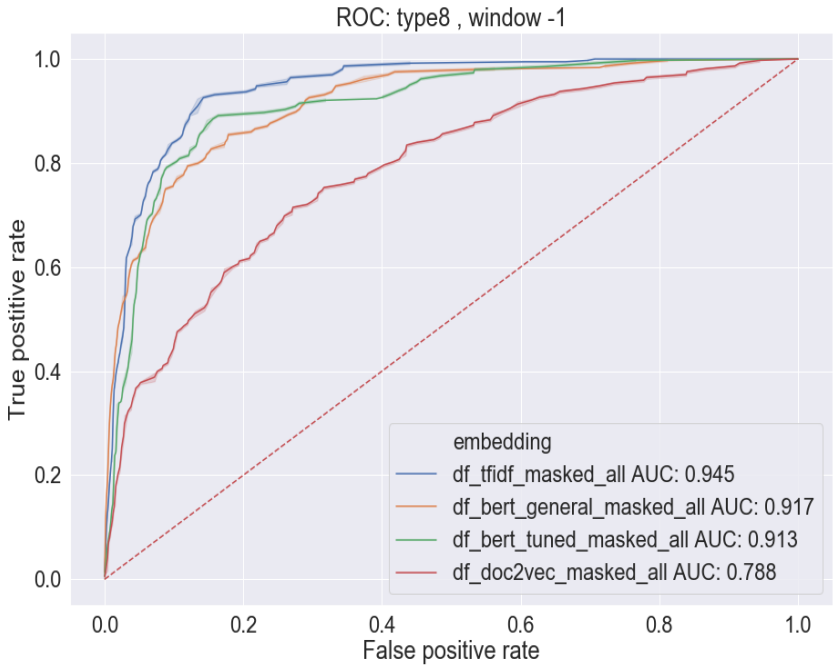


(B)

FIGURE 6.11: ROC curves for models trained on tuned BERT embeddings with masked company names across windows -1 and -5. The models trained on the datasets produced by the Gate Type 4, Gate Type 5, Gate Type 8 are better at separating between two binary classes - achieving AUC above 0.88.



(A)



(B)

FIGURE 6.12: ROC curves for models trained on different embeddings and data (with masked company's name) generated by the Gate Type 4. The models trained on the TF-IDF embeddings show a better performance in differentiating between the classes achieving the AUC value of above 0.91. In comparison, both BERT embeddings (tuned and general) are fluctuating around a similar AUC value of 0.9.

Chapter 7

Conclusion and Future Work

This chapter provides a summary of the key research work outcomes, insights and contribution to science. We also identify the opportunities for further research work.

7.1 Conclusion

With the growth of the social media, online blogs, news and other information content generated on the Internet, more and more investors are trying to utilize the signals generated from these alternative data sources for the portfolio construction. Although the key problem with the online content is that it is extremely noisy containing a lot of irrelevant information. It becomes difficult to select the data sources and build models that can identify financially useful information from the noise. In addition, in order to build better models that extract signals from the online content, they have to be domain-specific i.e. designed for a specific industry. For example, a popularity and consumer engagement on the Instagram or Twitter could be important metrics for a consumer fashion company *Adidas*, meanwhile, the same metrics would not be so relevant for a business-to-business telecommunication equipment company *Cisco*. Hence, the alternative data sources and models built to extract the trading signals have to be carefully selected and industry-specific.

Unlike the majority of previous studies that use alternative data for financial decision making, in this research work, we focus on companies that operate in a single Apparel and Footwear industry in order to design domain-specific models to analyse the equities. The main objective of this work is to design new frameworks for extracting financially relevant signals about fashion companies from alternative data sources. During this study, the following types of alternative data are used: a fashion company's popularity on social media platform Instagram and company-related news articles. The research work consists of three independent experiments. The outcomes from the experiments and the key insights are summarized below.

1. *Relationship between Instagram Popularity and Stock Prices*: the experiment aims to explore whether the changes in a fashion company's popularity on Instagram have a relationship with its financial performance.

- We find that there is a statistically significant cross-correlation between the changes in followers and returns for certain assets (*Michael Kors* and *Ralph Lauren*) where returns lag between 16 to 21 days.
- The yearly mean changes in followers and the change in the stock price over the same period are correlated. This means that if a company is accumulating followers at a high rate throughout a year, its share price is likely to increase as well.
- Our proposed Instagram trading strategy (*Instagram Strategy 2*) generated positive mean returns (higher returns than a random strategy), showing that the relationship between a company's popularity and its share price could be related only at some points in time and not continuously, i.e. during the *event* periods.
- In general, we find that even though the signals derived from the changes in Instagram followers are weak, the data source can still contribute to the overall performance of a strategy if used in conjunction with other information.

2. *Dynamic Density-based News Clustering*: the goal is to design an unsupervised framework for extracting events from the noisy news articles without any prior knowledge about the events.

   - We present the Dynamic Density-based News Clustering Framework to cluster a company's articles and extract events. The framework uses the DBSCAN clustering algorithm where its cluster density and size parameters are selected dynamically using a grid search where the objective is to find a set of parameters that maximises the inter-cluster Information Entropy.
   - The proposed framework is able to extract more events from the data in comparison to a clustering algorithm that uses static cluster density and size parameters.

3. *ALGA: Automatic Logic Gate Annotator for Event Detection*: the objective is to build a domain-specific (with a focus on Apparel and Footwear industry) financially relevant news events classifier that is able to identify news stories that are likely to impact a company's stock price.

   - First, we build an original news labelling framework ALGA (Automatic Logic Gate Annotator) that automatically labels large amounts of news articles using a rules-based approach and separate news stories into financially relevant and noise.
   - Second, using the training data generated by the framework, the financial news event detection classifier is trained that demonstrates the state-of-the-art results in domain-specific event detection problem.

7.2 Contribution to Science

The contributions of this research work to science are summarized as per following.

- To our knowledge, we present the original work in applying Machine Learning and Natural Language Processing techniques for analysing alternative data on trend-driven fashion equities for investment decision making.
- We explore Instagram followers as an alternative data source and show that it could be successfully used to generate trading signals and help to infer a company's future financial performance.
- Introduce the new metrics to quantify and track the popularity of a fashion brand on the social media channels (e.g. Instagram) for investing and trading.
- Unique design of the Dynamic Density-based News Clustering framework that is able to find events clusters of various sizes in the news articles without the expert knowledge, i.e. predefined parameters. The framework dynamically adjusts the cluster parameters based on the inter-cluster Information Entropy.
- We present the original Automatic Logic Gate Annotator framework (ALGA) that is able to automatically label news articles for the event detection classification problem at scale. The framework demonstrates transferability to other domains and the classification models trained on such data can be easily interpretable, explainable.
- We train the Apparel and Footwear financially-relevant news events classifier using the datasets generated by ALGA's framework and show the state-of-the-art performance in a domain-specific financial event detection task.
- Creation of the *Fashion Finance Dictionary* that contains 320 phrases related to various financially-relevant events in the Apparel and Footwear industry.

7.3 Further Work

The work presented in this thesis is an original study of an application of Machine Learning and Natural Language Processing (NLP) for the alternative data analysis of fashion assets. With this study, we fill in a research gap in the literature and start a new domain-specific research area in fashion finance. This study opens new opportunities for further academic and industry research. We identify further research opportunities below.

Consumer Sentiment on Instagram

Instagram is driven by visual content and therefore it became an essential social media platform for fashion brands to showcase and sell. Although, unlike Twitter, there

is still a lack of research done utilizing data from Instagram for investment decision making. In this research, we present new metrics to quantify a brand's popularity on Instagram. In addition, other data points such as comments on brand profiles, brand mentions could be investigated further to analyse consumer sentiment. This would give additional insight into how the brand is perceived by the public and how it is changing over time.

Automatic News Labelling for Financial Analysis across Different Domains

In this research work, we introduce an original work of automatic news articles labelling framework ALGA. The framework is designed to label news articles about fashion equities, although it can be easily adjusted to another domain by changing the *Fashion Finance Dictionary* that contains fashion events synonyms to a dictionary from another domain e.g. Technology or Real Estate. The transferability of the ALGA's framework could be investigated further to explore the properties of the framework and how it would work in other industry domains.

Deep Learning for Financial Event Detection

During this study, we train a Logistic Regression classifier for the fashion news events detection task using the data generated by the ALGA's framework. As a further research work, other classification models, for example, Deep Neural Networks with Attention layers, Transformer Neural Networks could be trained to see if they would improve the classification results.

E-commerce Data for Investment Decision Making

There is a wide range of different alternative data sources (in addition to social media or news) that could be used to analyse fashion equities. These sources could help to infer the company's revenues before they are publicly announced. For example, in order to gain better insight into a brand's sales, the data from its online shop could be analysed. The brand's or retailer's e-commerce sites can be scraped daily to track product availability, new products in stock, changes in prices, discounting, etc. These data points can provide with valuable insights into the financial health of a company. For example, if a company is heavily discounting most of its stock can signal a reduction in its future profit margins.

Alternative Data Sources

In addition to the data analysed in this research or mentioned above, other data sources could be explored to analyse Apparel and Footwear companies. As further research work, the data sources such as the following could be explored: a company's job listings, online store traffic data, product reviews, etc.

Appendix A

Appendixes

A.1 Relationship between Instagram Popularity and Stock Prices

A.1.1 Datasets

The dataset used for this experiment includes 11 publicly traded apparel/footwear equities and their associated daily Instagram follower counts, daily stock prices and reported revenues for a period between 2014-2018.

In order to better visualise the trends in the datasets we plot normalized time series plots of all three data types (followers, stock prices and revenues) for every company. Below is a list of all the plots:

- The Figure A.1 shows normalized time series plot of all three data types (followers, stock prices and revenues) for *Michael Kors*.

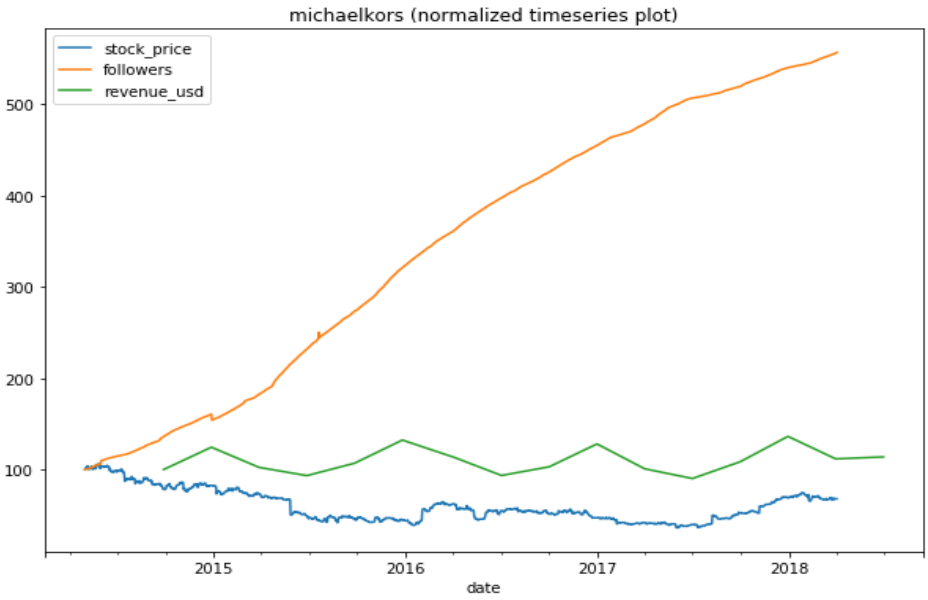


FIGURE A.1: Michael Kors: normalized Instagram followers, stock prices and revenues during 2014-2018.

- The Figure A.3 shows normalized time series plot of all three data types (followers, stock prices and revenues) for *Hugo Boss*.

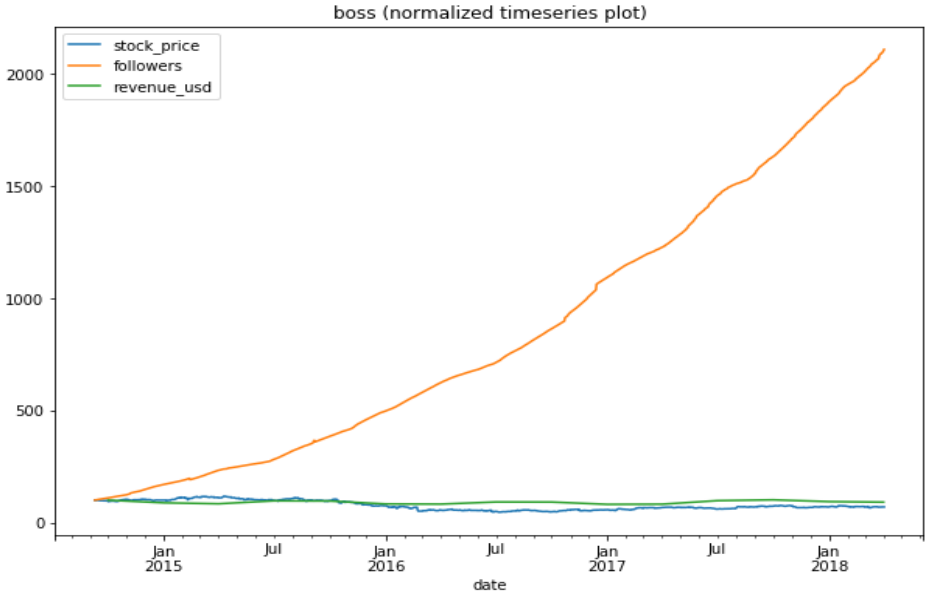


FIGURE A.2: Hugo Boss: normalized Instagram followers, stock prices and revenues during 2014-2018.

- The Figure A.3 shows normalized time series plot of all three data types (followers, stock prices and revenues) for *Hugo Boss*.

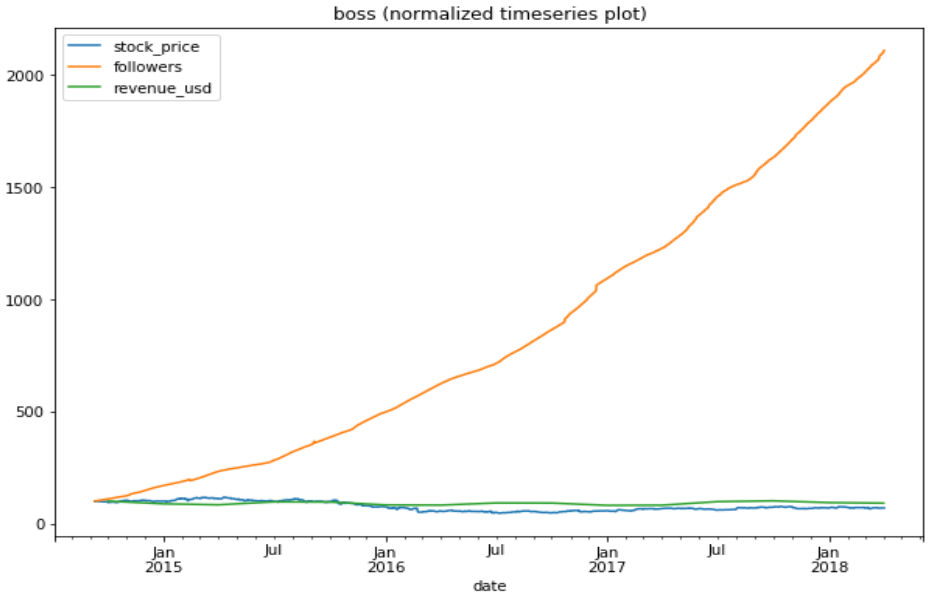


FIGURE A.3: Hugo Boss: normalized Instagram followers, stock prices and revenues during 2014-2018.

- The Figure A.4 shows normalized time series plot of all three data types (followers, stock prices and revenues) for *Brunello Cucinelli*.
- The Figure A.5 shows normalized time series plot of all three data types (followers, stock prices and revenues) for *Salvatore Ferragamo*.

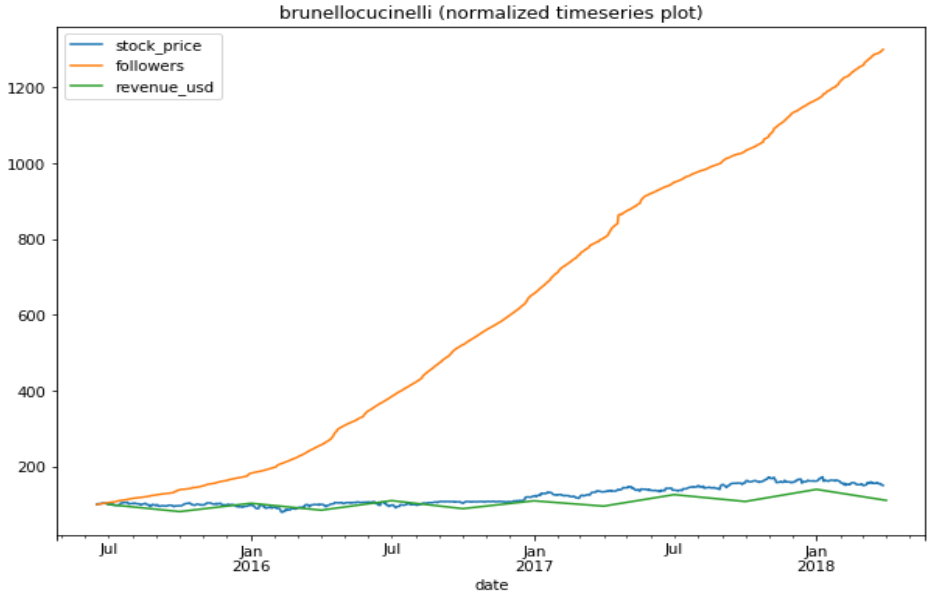


FIGURE A.4: Brunello Cucinelli: normalized Instagram followers, stock prices and revenues during 2015-2018.

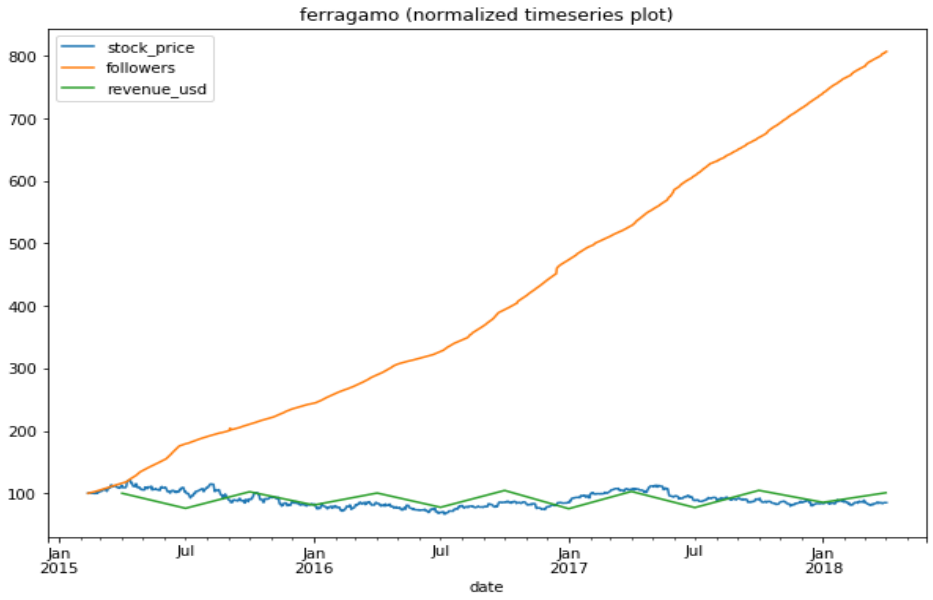


FIGURE A.5: Salvatore Ferragamo: normalized Instagram followers, stock prices and revenues during 2015-2018.

- The Figure A.6 shows normalized time series plot of all three data types (followers, stock prices and revenues) for *Hermes*.
- The Figure A.7 shows normalized time series plot of all three data types (followers, stock prices and revenues) for *Moncler*.
- The Figure A.8 shows normalized time series plot of all three data types (followers, stock prices and revenues) for *Mulberry*.

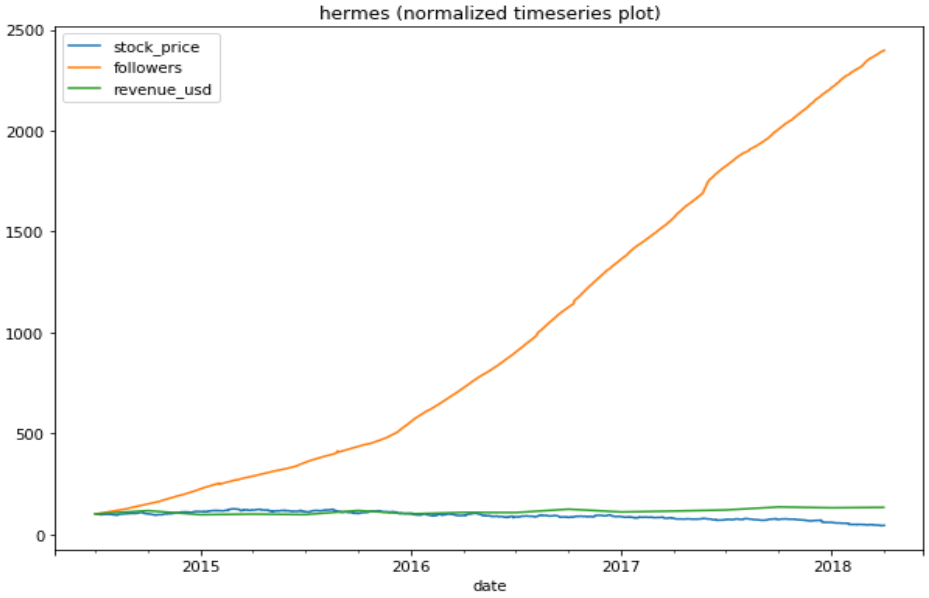


FIGURE A.6: Hermes: normalized Instagram followers, stock prices and revenues during 2014-2018.

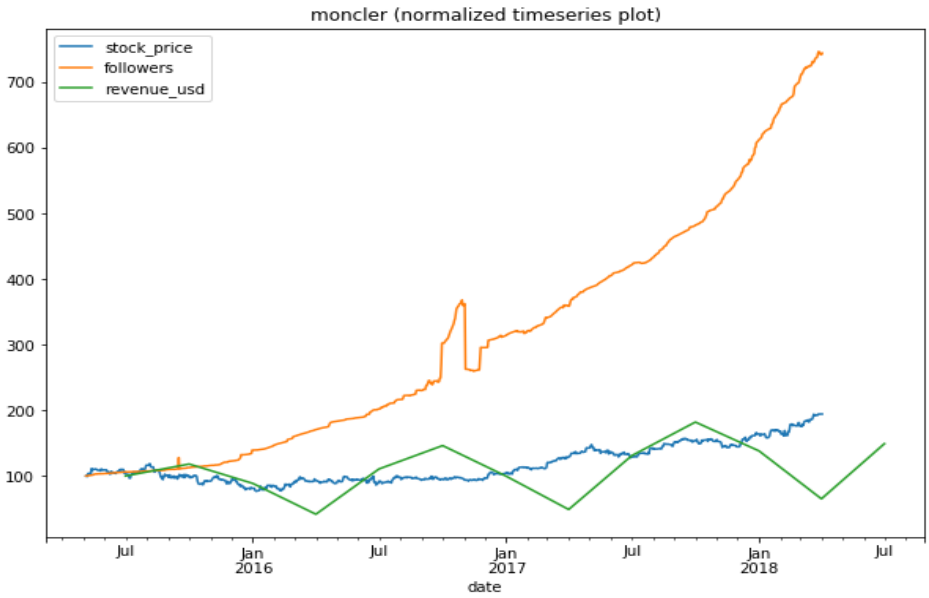


FIGURE A.7: Moncler: normalized Instagram followers, stock prices and revenues during 2015-2018.

- The Figure A.9 shows normalized time series plot of all three data types (followers, stock prices and revenues) for *Prada*.
- The Figure A.10 shows normalized time series plot of all three data types (followers, stock prices and revenues) for *Ralph Lauren*.

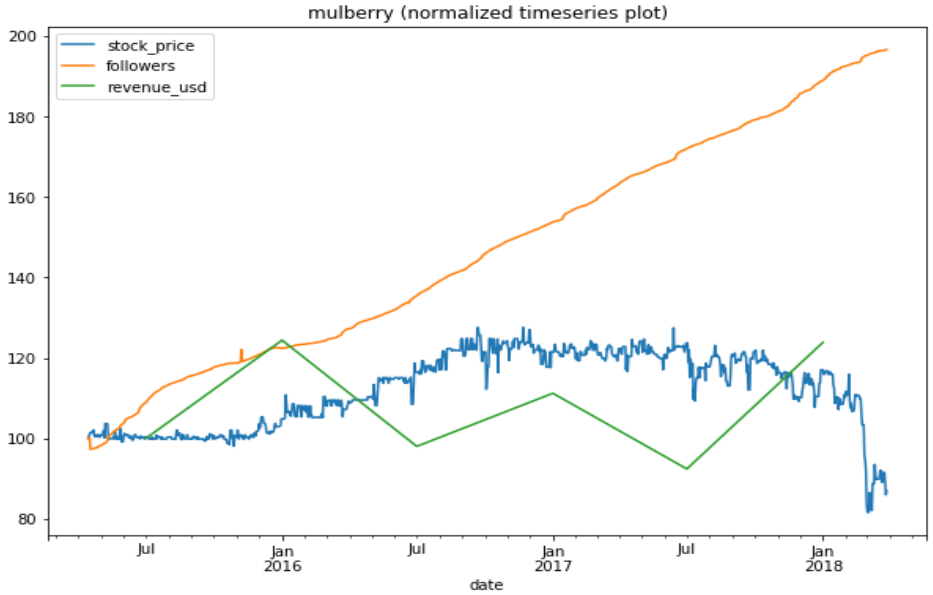


FIGURE A.8: Mulberry: normalized Instagram followers, stock prices and revenues during 2015-2018.

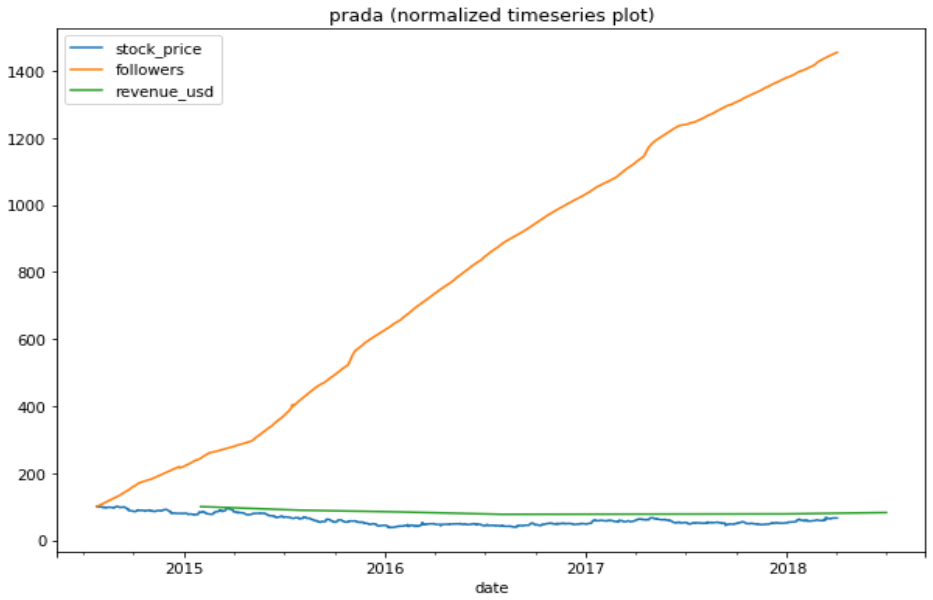


FIGURE A.9: Prada: normalized Instagram followers, stock prices and revenues during 2014-2018.

A.1.2 Feature Engineering

After creating the features, they are tested for stationary and normality using graphical methods. Bellow are the frequency distributions of a selection of features for all the companies used for the analysis.

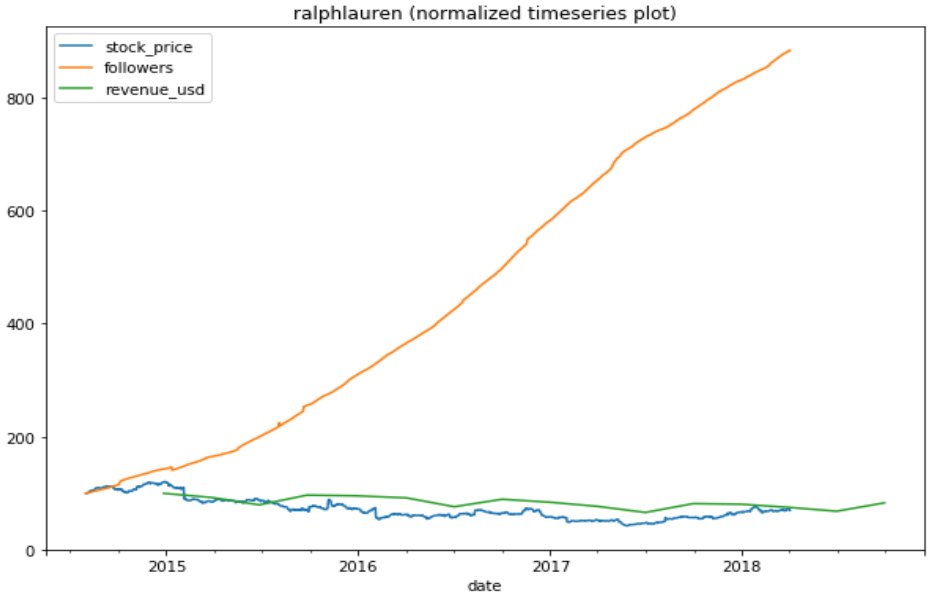
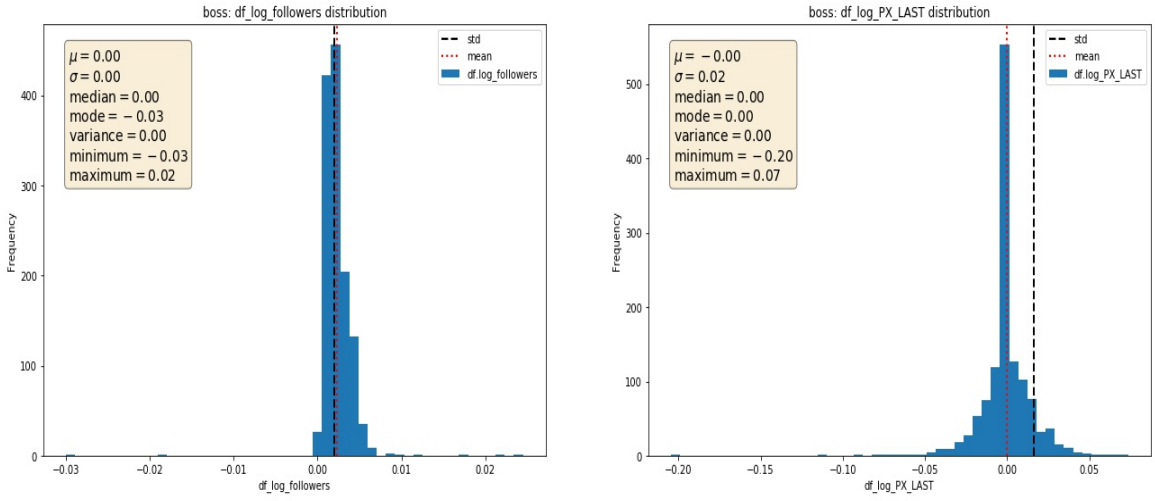


FIGURE A.10: Ralph Lauren: normalized Instagram followers, stock prices and revenues during 2014-2018.

Hugo Boss

The Figures A.11-A.15 show the frequency distributions of a selection of features for *Hugo Boss* that are assumed to be normally distributed and used for the further analysis.

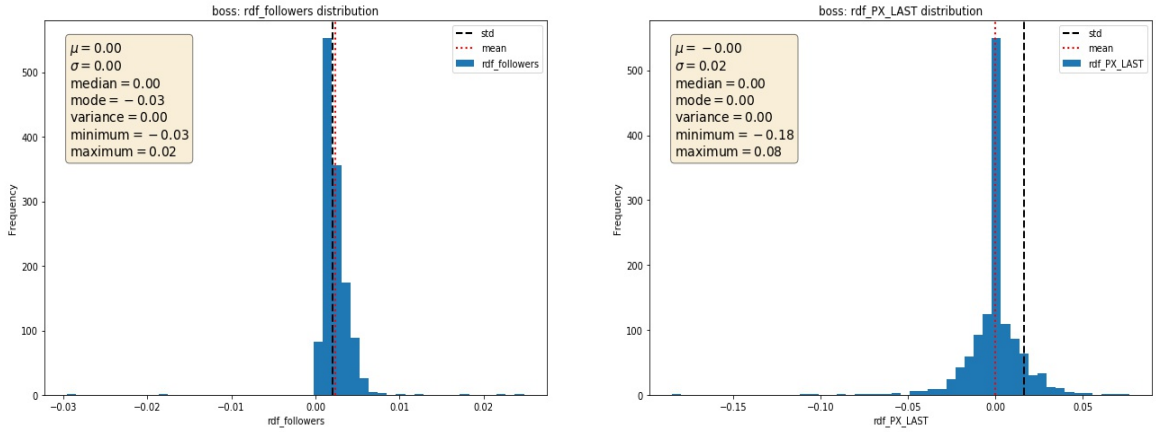
The Figure A.11 shows the frequency distributions of a change in logarithmic transforms of followers (A) and stock prices (B) for *Hugo Boss*.



(A) (B)

FIGURE A.11: Hugo Boss: shows the frequency distributions of a change in logarithmic transforms of followers (A) and stock prices (B) (2014-2018).

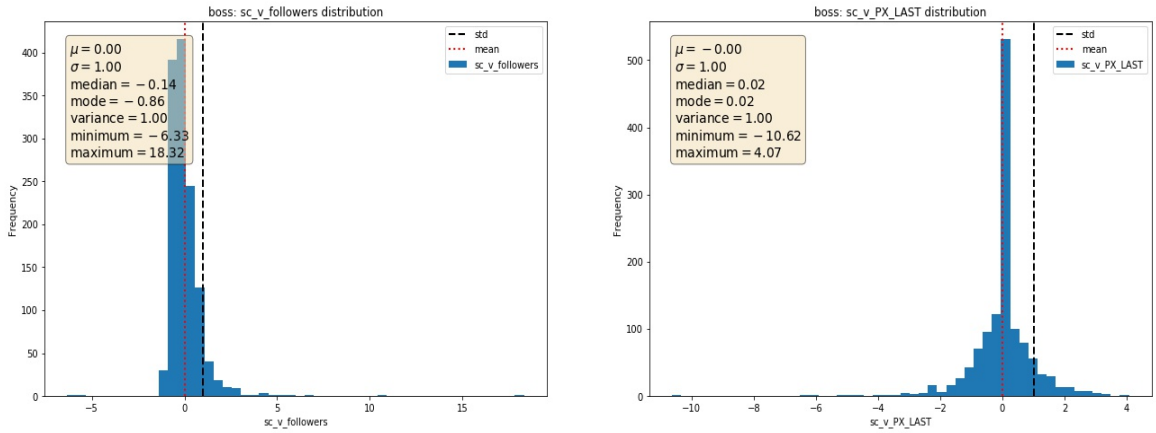
The Figure A.12 shows the frequency distributions of a relative change in followers (A) and stock prices (B) for *Hugo Boss*.



(A) (B)

FIGURE A.12: Hugo Boss: shows the frequency distributions of a relative change in followers (A) and stock prices (B) (2014-2018).

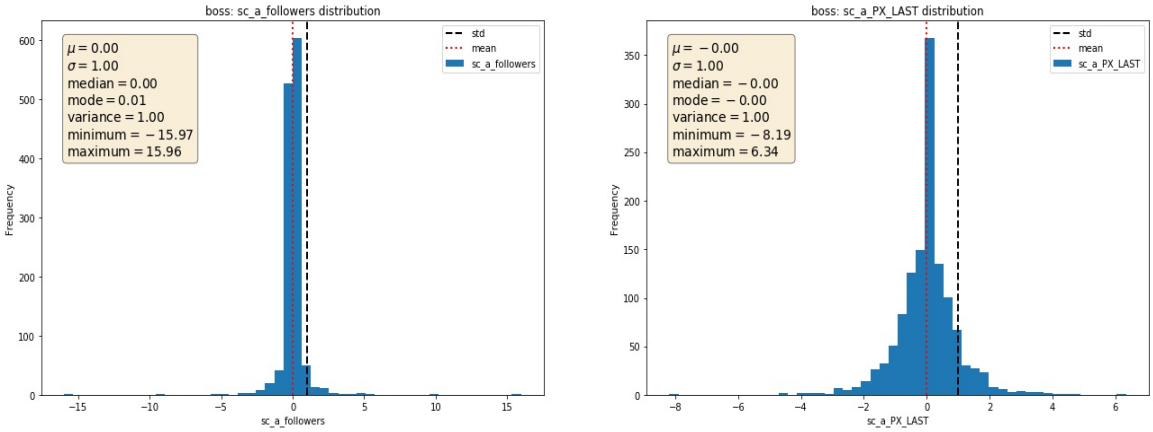
The Figure A.13 shows the frequency distributions of followers velocity (A) and stock prices velocity (B) for *Hugo Boss*.



(A) (B)

FIGURE A.13: Hugo Boss: shows the frequency distributions of followers velocity (A) and stock prices velocity (B) (2014-2018).

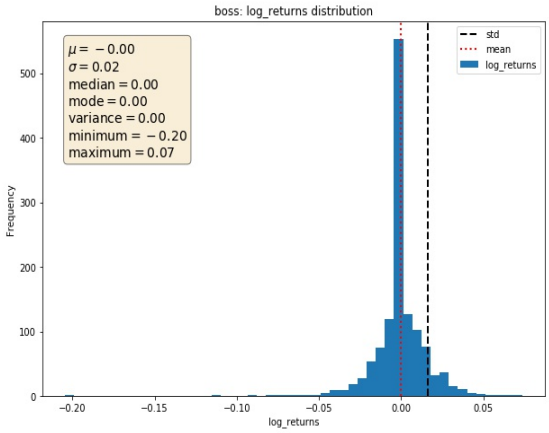
The Figure A.14 shows the frequency distributions of followers acceleration (A) and stock prices acceleration (B) for *Hugo Boss*.



(A) (B)

FIGURE A.14: Hugo Boss: shows the frequency distributions of followers acceleration (A) and stock prices acceleration (B) (2014-2018).

The Figure A.15 shows the frequency distributions of the logarithmic transform of returns for *Hugo Boss*.



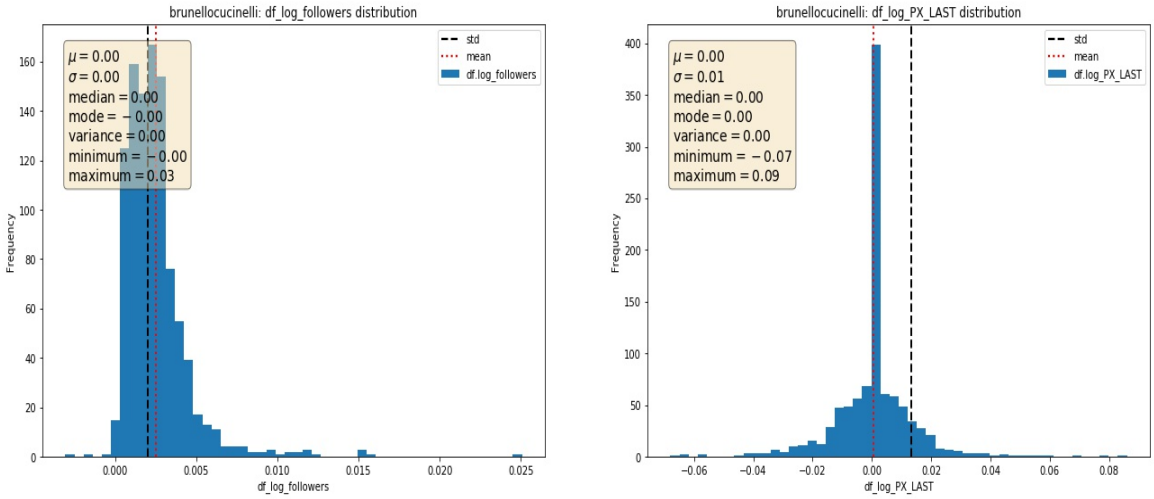
(A)

FIGURE A.15: Hugo Boss: shows the frequency distributions of the logarithmic transform of returns (2014-2018).

Brunello Cucinelli

The Figures A.16-A.20 show the frequency distributions of a selection of features for *Brunello Cucinelli* that are assumed to be normally distributed and used for the further analysis.

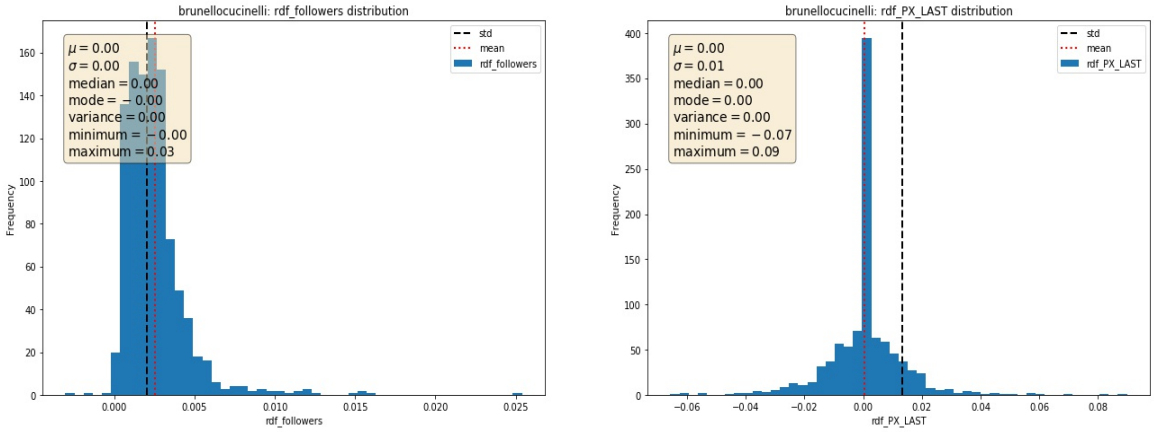
The Figure A.16 shows the frequency distributions of a change in logarithmic transforms of followers (A) and stock prices (B) for *Brunello Cucinelli*.



(A) (B)

FIGURE A.16: Brunello Cucinelli: shows the frequency distributions of a change in logarithmic transforms of followers (A) and stock prices (B) (2014-2018).

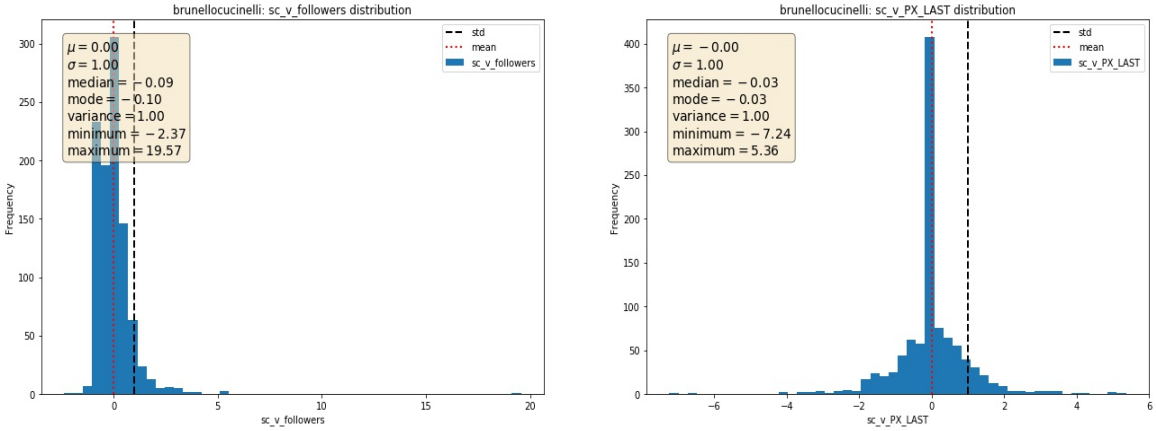
The Figure A.17 shows the frequency distributions of a relative change in followers (A) and stock prices (B) for *Brunello Cucinelli*.



(A) (B)

FIGURE A.17: Brunello Cucinelli: shows the frequency distributions of a relative change in followers (A) and stock prices (B) (2014-2018).

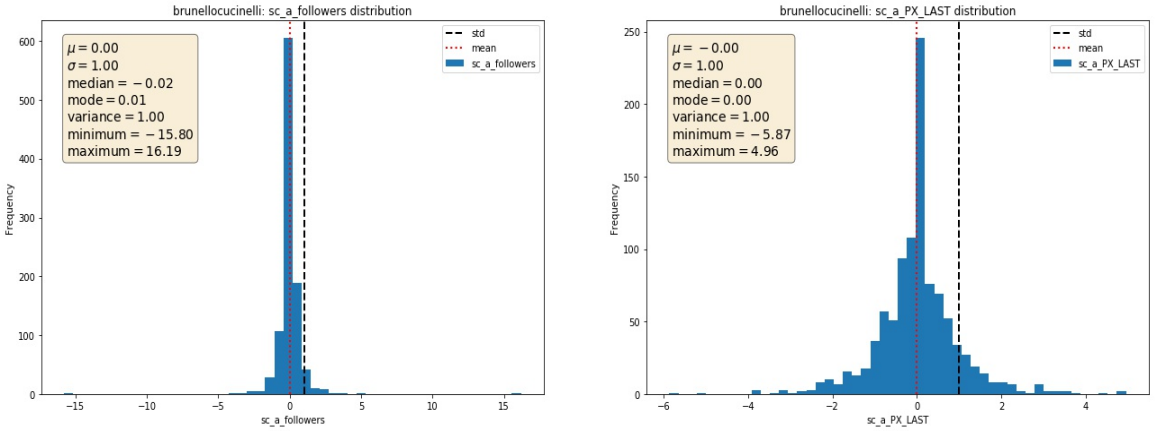
The Figure A.18 shows the frequency distributions of followers velocity (A) and stock prices velocity (B) for *Brunello Cucinelli*.



(A) (B)

FIGURE A.18: Brunello Cucinelli: shows the frequency distributions of followers velocity (A) and stock prices velocity (B) (2014-2018).

The Figure A.19 shows the frequency distributions of followers acceleration (A) and stock prices acceleration (B) for *Brunello Cucinelli*.



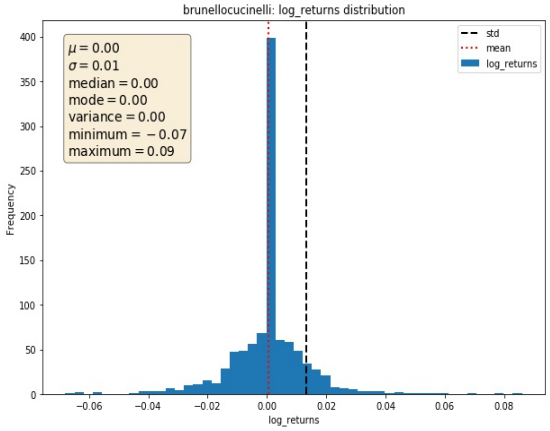
(A) (B)

FIGURE A.19: Brunello Cucinelli: shows the frequency distributions of followers acceleration (A) and stock prices acceleration (B) (2014-2018).

The Figure A.20 shows the frequency distributions of the logarithmic transform of returns for *Brunello Cucinelli*.

Salvatore Ferragamo

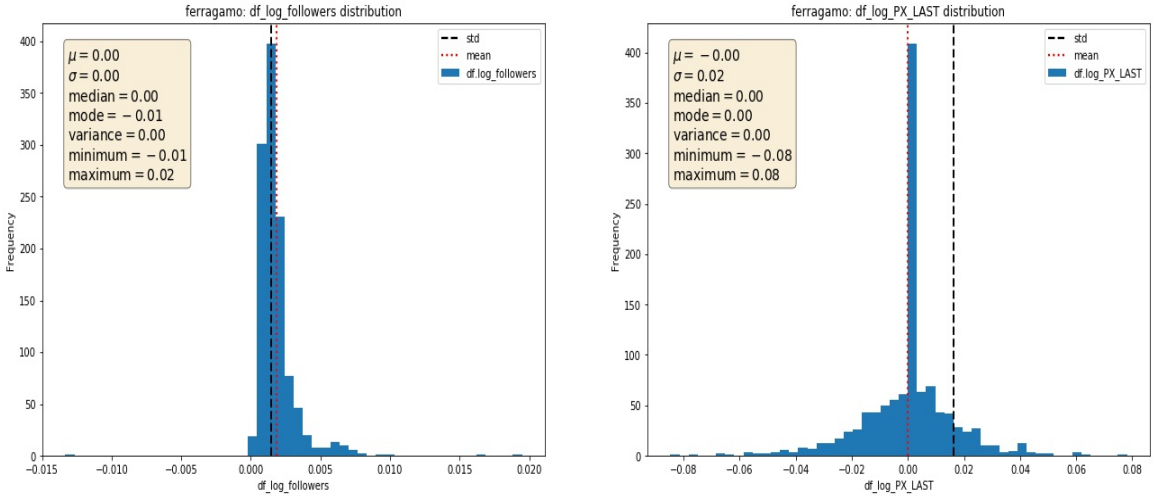
The Figures A.21-A.25 show the frequency distributions of a selection of features for *Salvatore Ferragamo* that are assumed to be normally distributed and used for the further analysis.



(A)

FIGURE A.20: Brunello Cucinelli: shows the frequency distributions of the logarithmic transform of returns (2014-2018).

The Figure A.21 shows the frequency distributions of a change in logarithmic transforms of followers (A) and stock prices (B) for *Salvatore Ferragamo*.

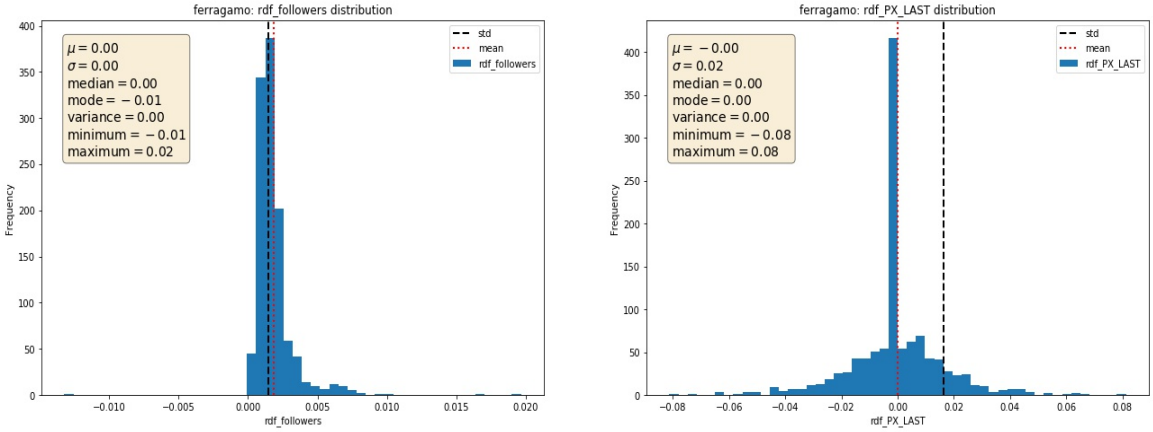


(A) (B)

FIGURE A.21: Salvatore Ferragamo: shows the frequency distributions of a change in logarithmic transforms of followers (A) and stock prices (B) (2014-2018).

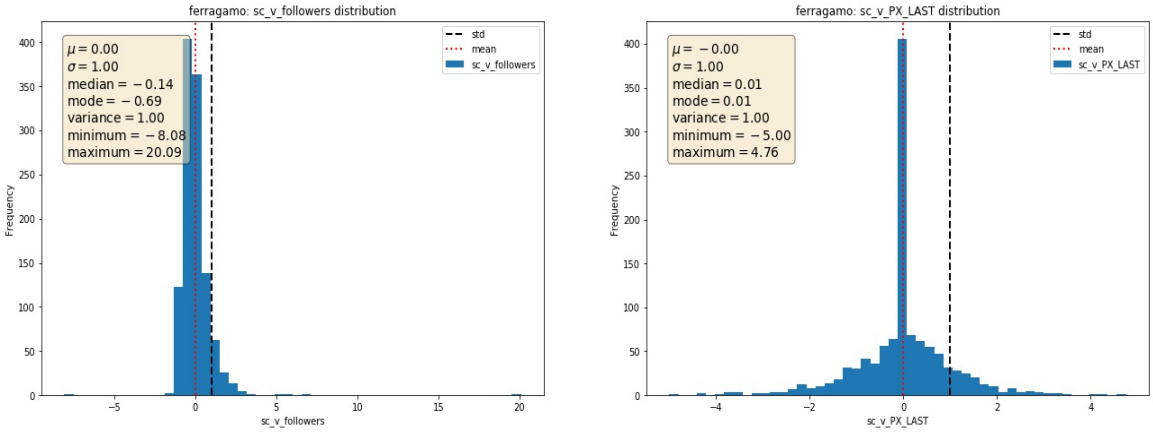
The Figure A.22 shows the frequency distributions of a relative change in followers (A) and stock prices (B) for *Salvatore Ferragamo*.

The Figure A.23 shows the frequency distributions of followers velocity (A) and stock prices velocity (B) for *Salvatore Ferragamo*.



(A) (B)

FIGURE A.22: Salvatore Ferragamo: shows the frequency distributions of a relative change in followers (A) and stock prices (B) (2014-2018).

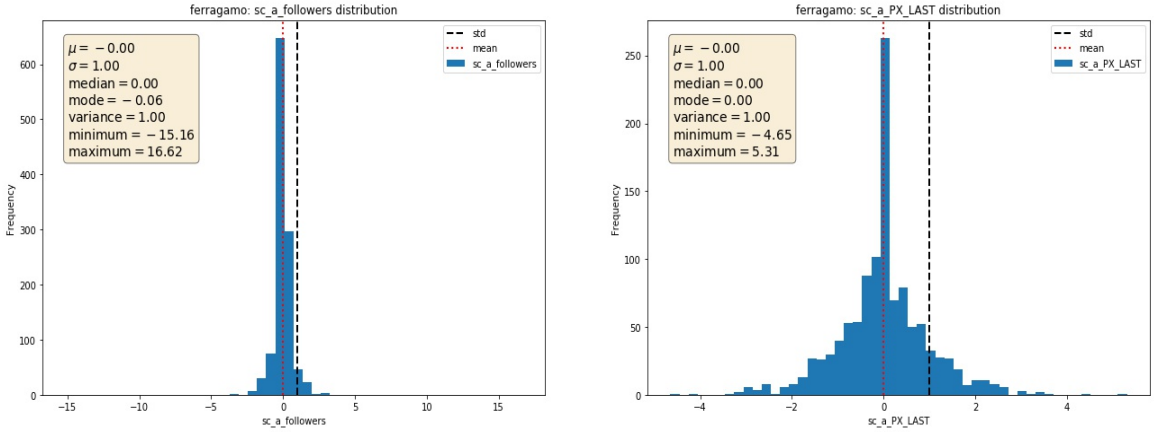


(A) (B)

FIGURE A.23: Salvatore Ferragamo: shows the frequency distributions of followers velocity (A) and stock prices velocity (B) (2014-2018).

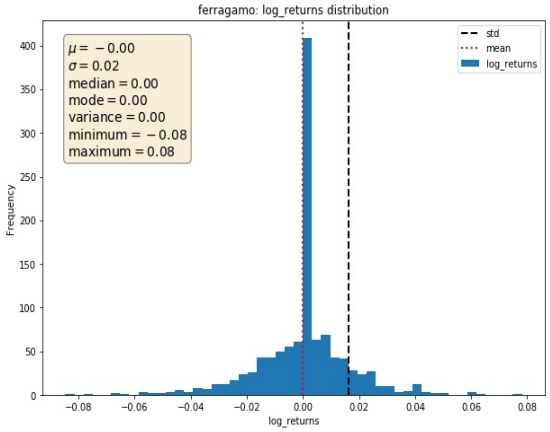
The Figure A.24 shows the frequency distributions of followers acceleration (A) and stock prices acceleration (B) for *Salvatore Ferragamo*.

The Figure A.25 shows the frequency distributions of the logarithmic transform of returns for *Salvatore Ferragamo*.



(A) (B)

FIGURE A.24: Salvatore Ferragamo: shows the frequency distributions of followers acceleration (A) and stock prices acceleration (B) (2014-2018).



(A)

FIGURE A.25: Salvatore Ferragamo: shows the frequency distributions of the logarithmic transform of returns (2014-2018).

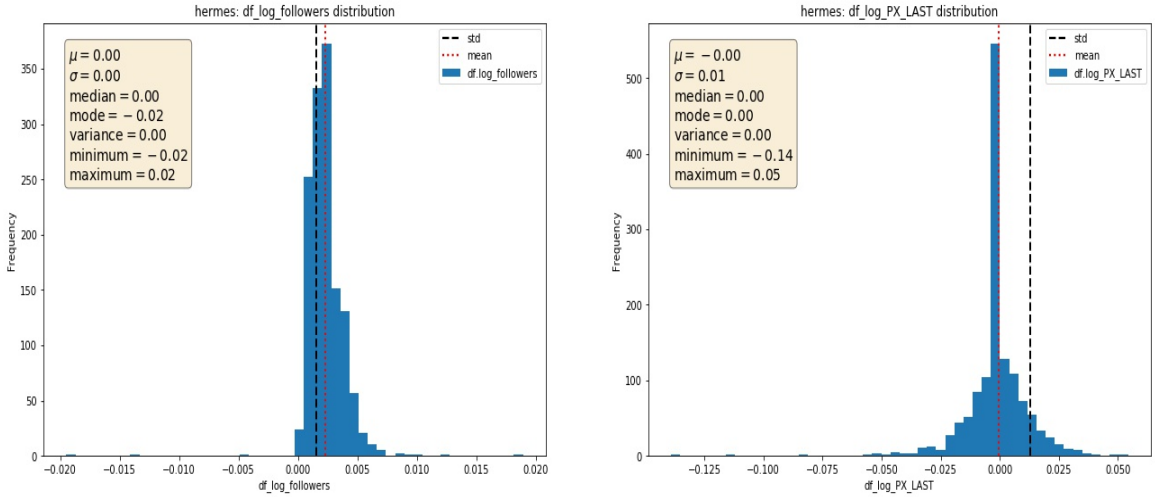
Hermes

The Figures A.26-A.30 show the frequency distributions of a selection of features for *Hermes* that are assumed to be normally distributed and used for the further analysis.

The Figure A.26 shows the frequency distributions of a change in logarithmic transforms of followers (A) and stock prices (B) for *Hermes*.

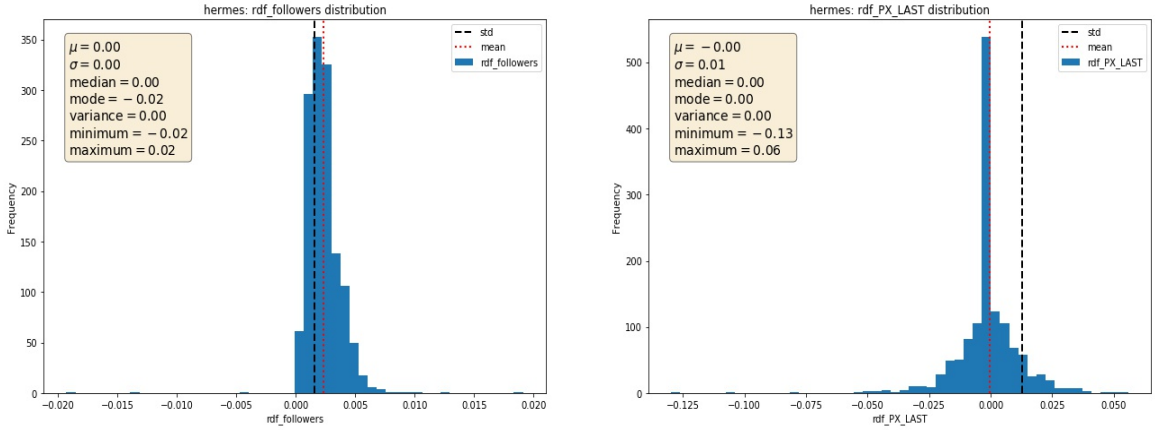
The Figure A.27 shows the frequency distributions of a relative change in followers (A) and stock prices (B) for *Hermes*.

The Figure A.28 shows the frequency distributions of followers velocity (A) and stock prices velocity (B) for *Hermes*.



(A) (B)

FIGURE A.26: Hermes: shows the frequency distributions of a change in logarithmic transforms of followers (A) and stock prices (B) (2014-2018).



(A) (B)

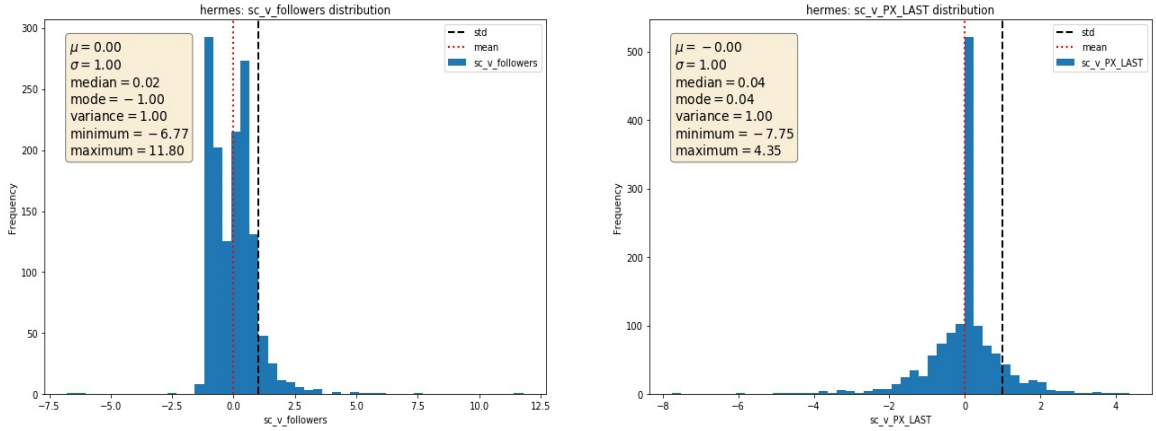
FIGURE A.27: Hermes: shows the frequency distributions of a relative change in followers (A) and stock prices (B) (2014-2018).

The Figure A.29 shows the frequency distributions of followers acceleration (A) and stock prices acceleration (B) for *Hermes*.

The Figure A.30 shows the frequency distributions of the logarithmic transform of returns for *Hermes*.

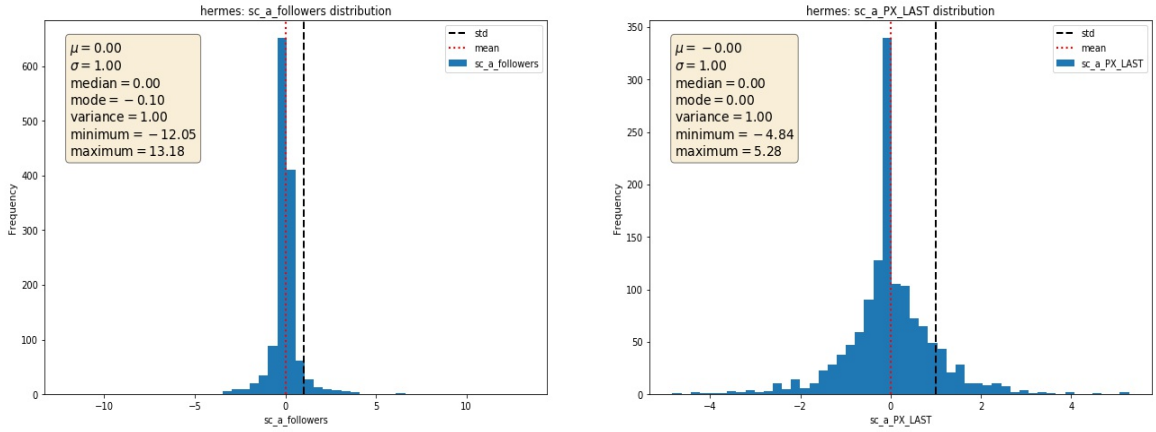
Michael Kors

The Figures A.31-A.35 show the frequency distributions of a selection of features for *Michael Kors* that are assumed to be normally distributed and used for the further



(A) (B)

FIGURE A.28: Hermes: shows the frequency distributions of followers velocity (A) and stock prices velocity (B) (2014-2018).



(A) (B)

FIGURE A.29: Hermes: shows the frequency distributions of followers acceleration (A) and stock prices acceleration (B) (2014-2018).

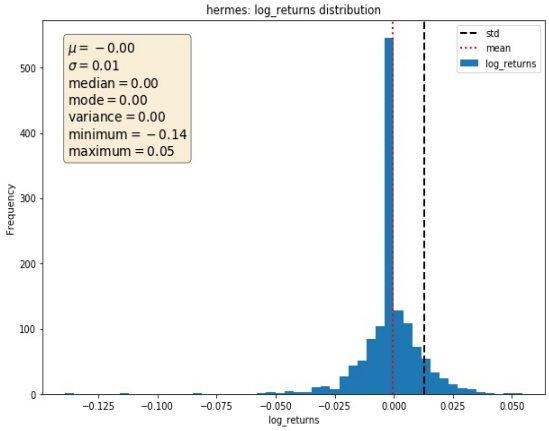
analysis.

The Figure A.31 shows the frequency distributions of a change in logarithmic transforms of followers (A) and stock prices (B) for *Michael Kors*.

The Figure A.32 shows the frequency distributions of a relative change in followers (A) and stock prices (B) for *Michael Kors*.

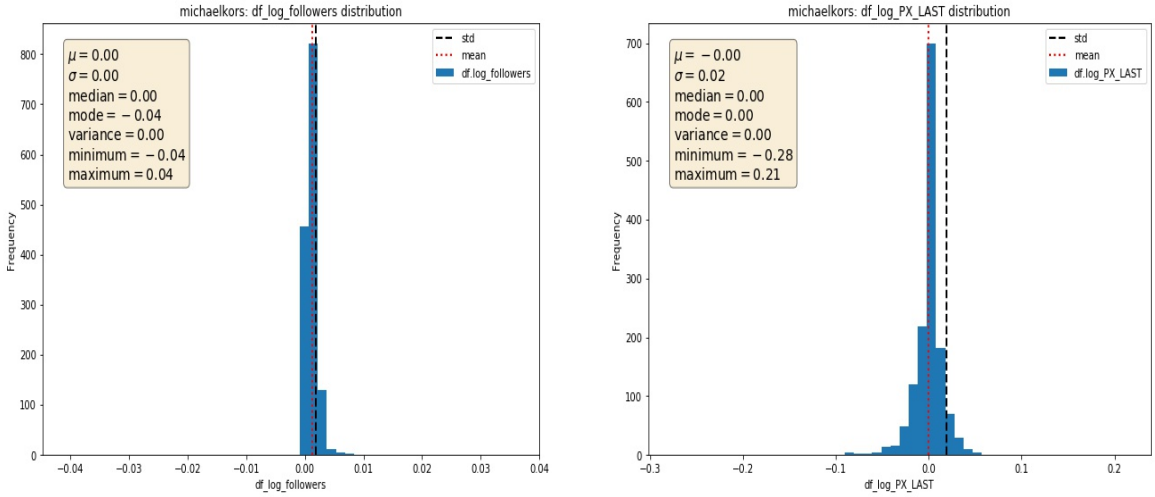
The Figure A.33 shows the frequency distributions of followers velocity (A) and stock prices velocity (B) for *Michael Kors*.

The Figure A.34 shows the frequency distributions of followers acceleration (A) and stock prices acceleration (B) for *Michael Kors*.



(A)

FIGURE A.30: Hermes: shows the frequency distributions of the logarithmic transform of returns (2014-2018).



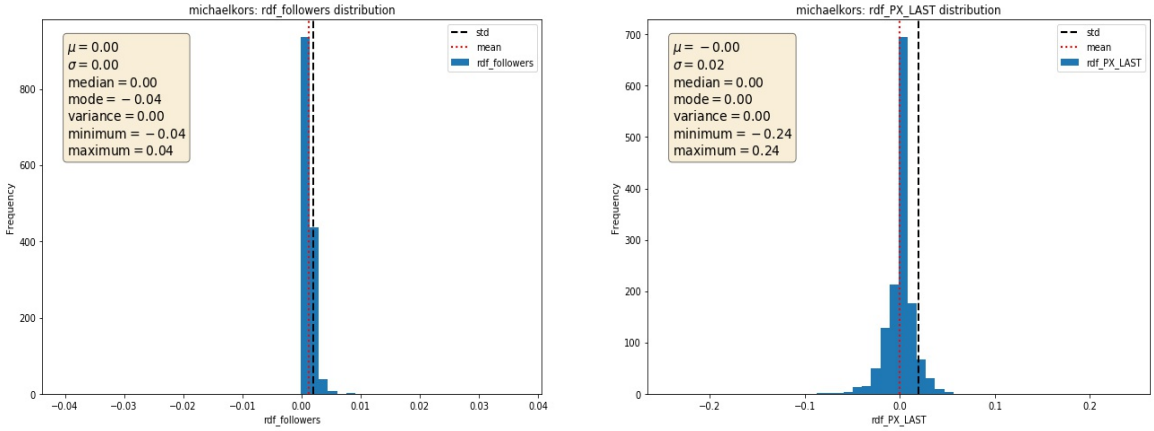
(A) (B)

FIGURE A.31: Michael Kors: shows the frequency distributions of a change in logarithmic transforms of followers (A) and stock prices (B) (2014-2018).

The Figure A.35 shows the frequency distributions of the logarithmic transform of returns for *Michael Kors*.

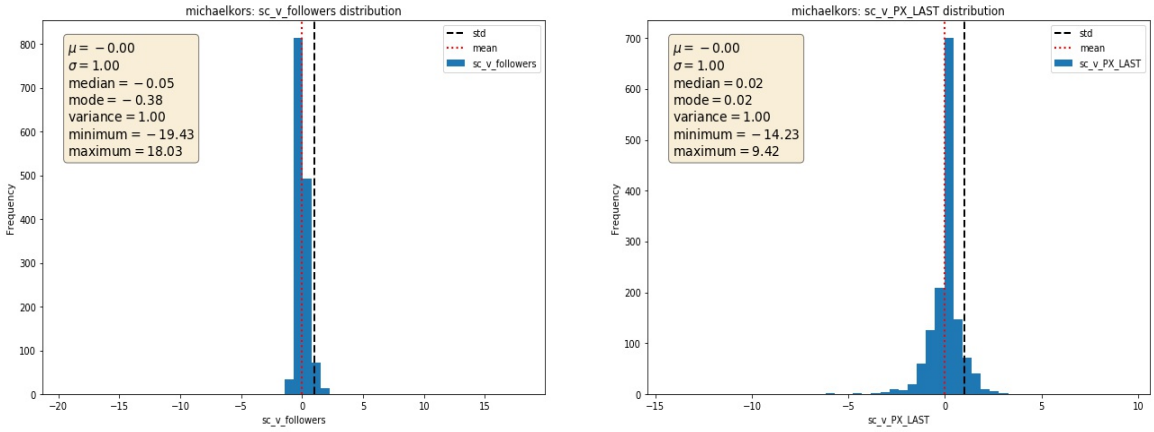
Moncler

The Figures A.36-A.40 show the frequency distributions of a selection of features for *Moncler* that are assumed to be normally distributed and used for the further analysis.



(A) (B)

FIGURE A.32: Michael Kors: shows the frequency distributions of a relative change in followers (A) and stock prices (B) (2014-2018).



(A) (B)

FIGURE A.33: Michael Kors: shows the frequency distributions of followers velocity (A) and stock prices velocity (B) (2014-2018).

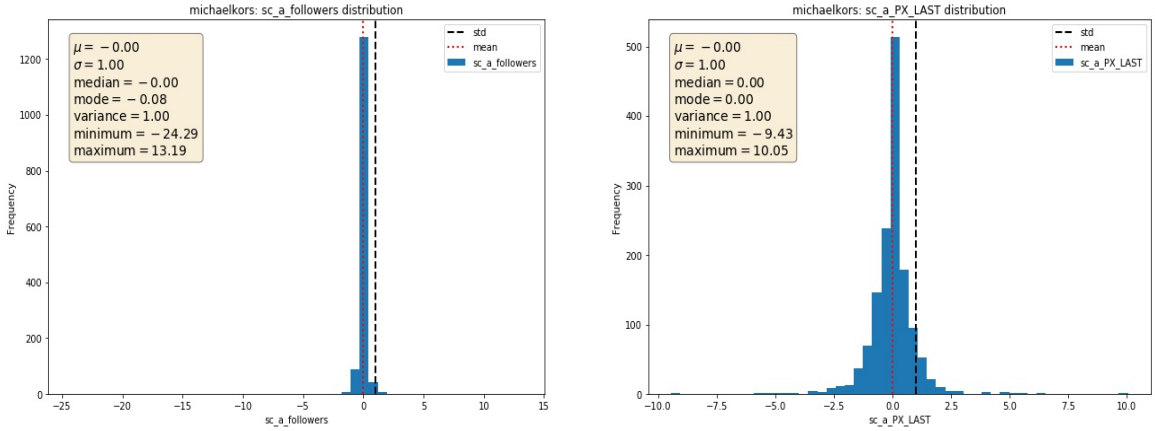
The Figure A.36 shows the frequency distributions of a change in logarithmic transforms of followers (A) and stock prices (B) for *Moncler*.

The Figure A.37 shows the frequency distributions of a relative change in followers (A) and stock prices (B) for *Moncler*.

The Figure A.38 shows the frequency distributions of followers velocity (A) and stock prices velocity (B) for *Moncler*.

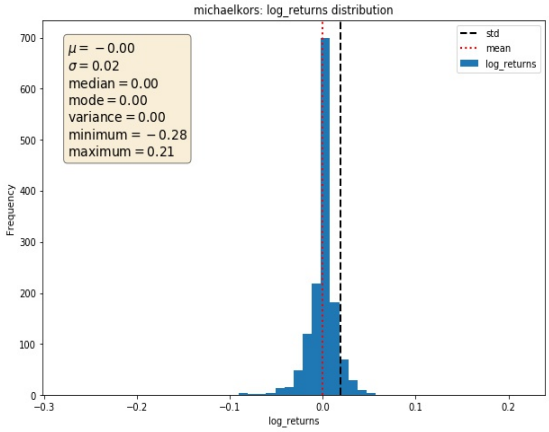
The Figure A.39 shows the frequency distributions of followers acceleration (A) and stock prices acceleration (B) for *Moncler*.

The Figure A.40 shows the frequency distributions of the logarithmic transform of returns for *Moncler*.



(A) (B)

FIGURE A.34: Michael Kors: shows the frequency distributions of followers acceleration (A) and stock prices acceleration (B) (2014-2018).



(A)

FIGURE A.35: Michael Kors: shows the frequency distributions of the logarithmic transform of returns (2014-2018).

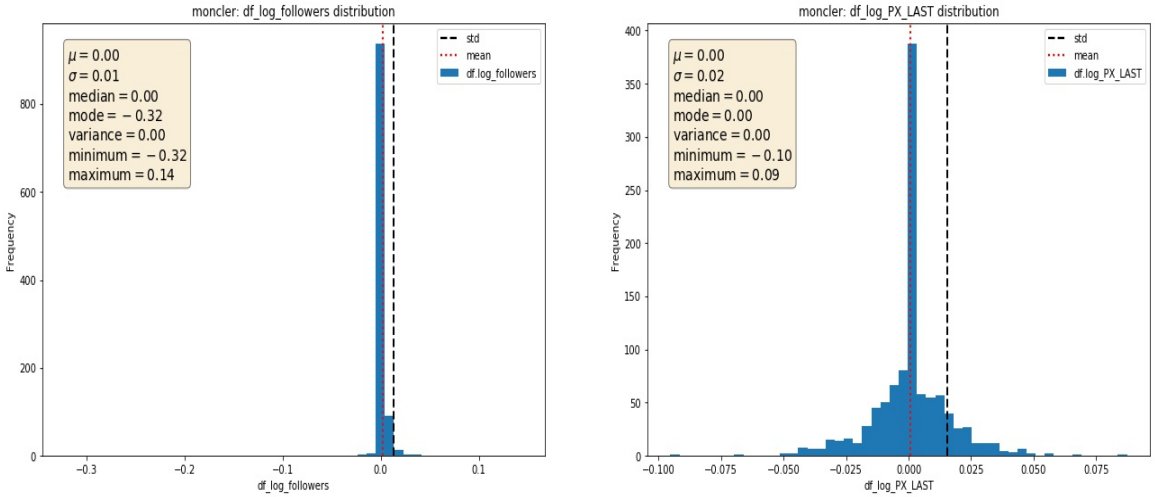
Mulberry

The Figures A.41-A.45 show the frequency distributions of a selection of features for *Mulberry* that are assumed to be normally distributed and used for the further analysis.

The Figure A.41 shows the frequency distributions of a change in logarithmic transforms of followers (A) and stock prices (B) for *Mulberry*.

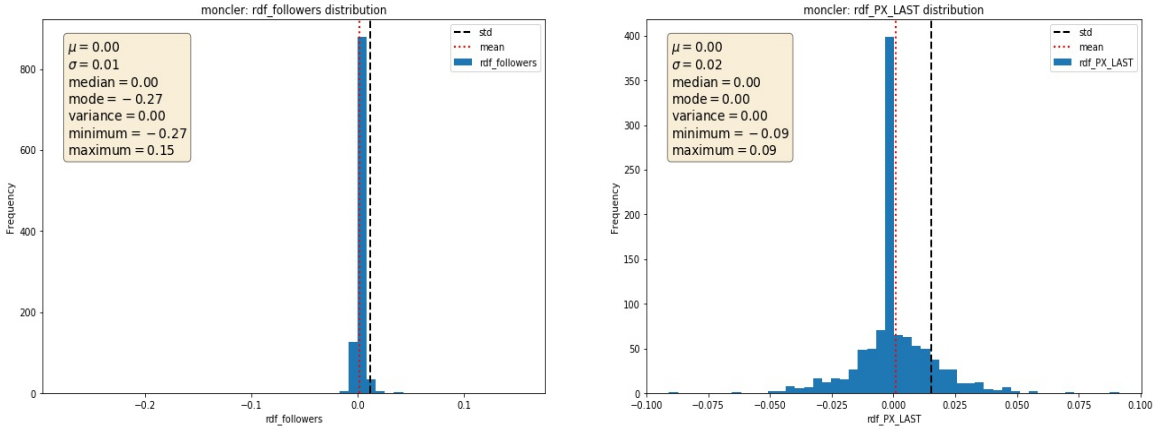
The Figure A.42 shows the frequency distributions of a relative change in followers (A) and stock prices (B) for *Mulberry*.

The Figure A.43 shows the frequency distributions of followers velocity (A) and stock prices velocity (B) for *Mulberry*.



(A) (B)

FIGURE A.36: Moncler: shows the frequency distributions of a change in logarithmic transforms of followers (A) and stock prices (B) (2014-2018).



(A) (B)

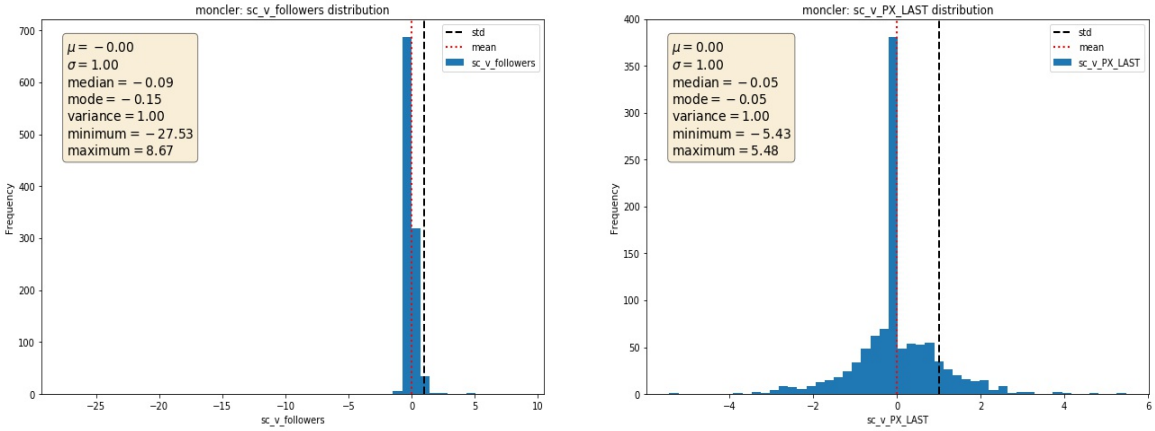
FIGURE A.37: Moncler: shows the frequency distributions of a relative change in followers (A) and stock prices (B) (2014-2018).

The Figure A.44 shows the frequency distributions of followers acceleration (A) and stock prices acceleration (B) for *Mulberry*.

The Figure A.45 shows the frequency distributions of the logarithmic transform of returns for *Mulberry*.

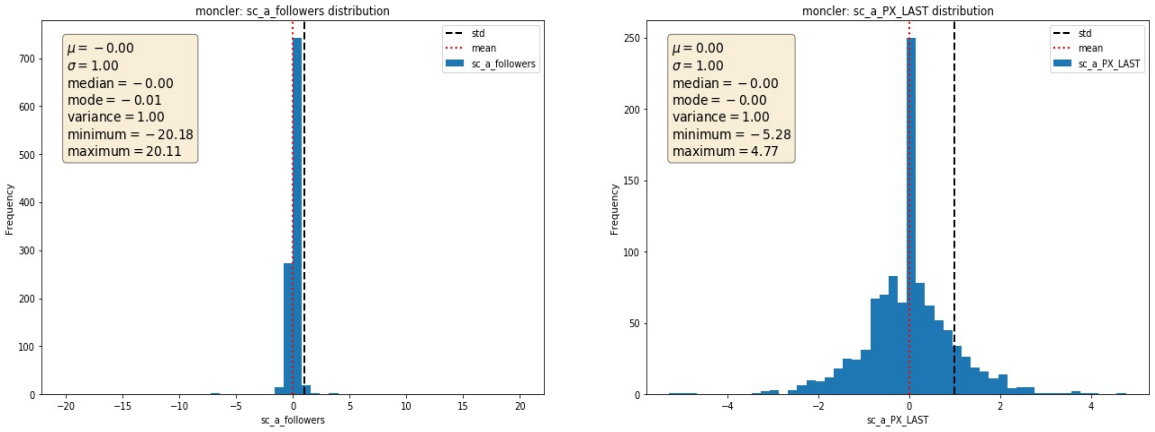
Prada

The Figures A.46-A.50 show the frequency distributions of a selection of features for *Prada* that are assumed to be normally distributed and used for the further analysis.



(A) (B)

FIGURE A.38: Moncler: shows the frequency distributions of followers velocity (A) and stock prices velocity (B) (2014-2018).



(A) (B)

FIGURE A.39: Moncler: shows the frequency distributions of followers acceleration (A) and stock prices acceleration (B) (2014-2018).

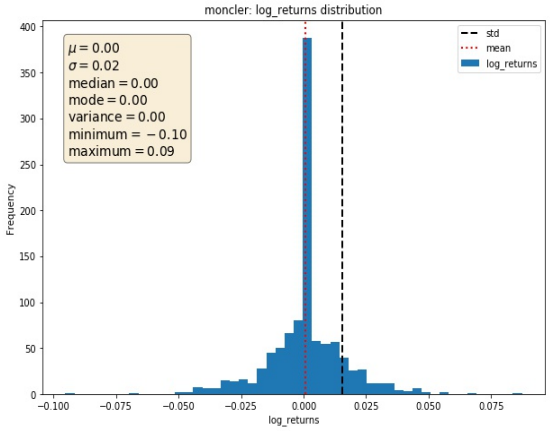
The Figure A.46 shows the frequency distributions of a change in logarithmic transforms of followers (A) and stock prices (B) for *Prada*.

The Figure A.47 shows the frequency distributions of a relative change in followers (A) and stock prices (B) for *Prada*.

The Figure A.48 shows the frequency distributions of followers velocity (A) and stock prices velocity (B) for *Prada*.

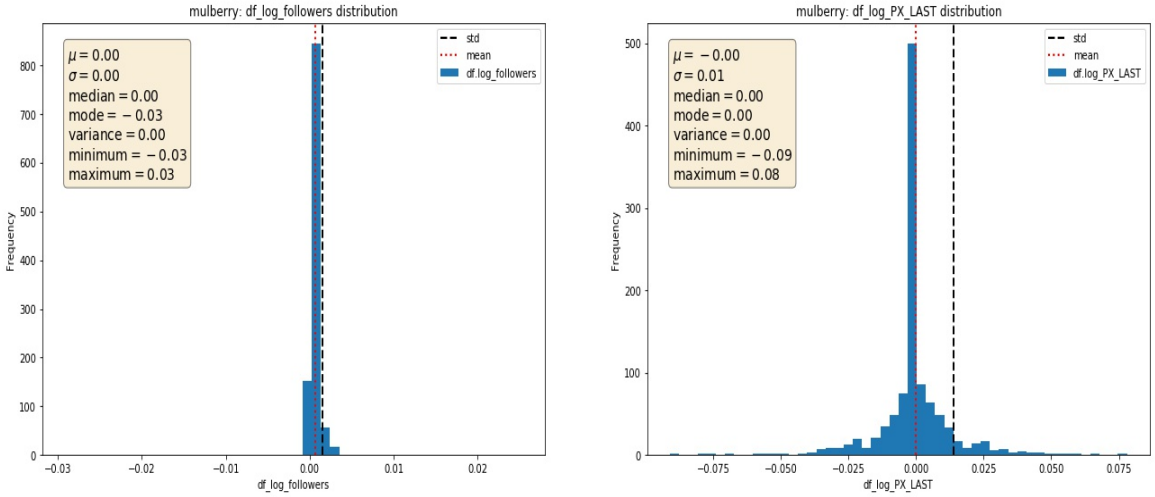
The Figure A.49 shows the frequency distributions of followers acceleration (A) and stock prices acceleration (B) for *Prada*.

The Figure A.50 shows the frequency distributions of the logarithmic transform of returns for *Prada*.



(A)

FIGURE A.40: Moncler: shows the frequency distributions of the logarithmic transform of returns (2014-2018).



(A) (B)

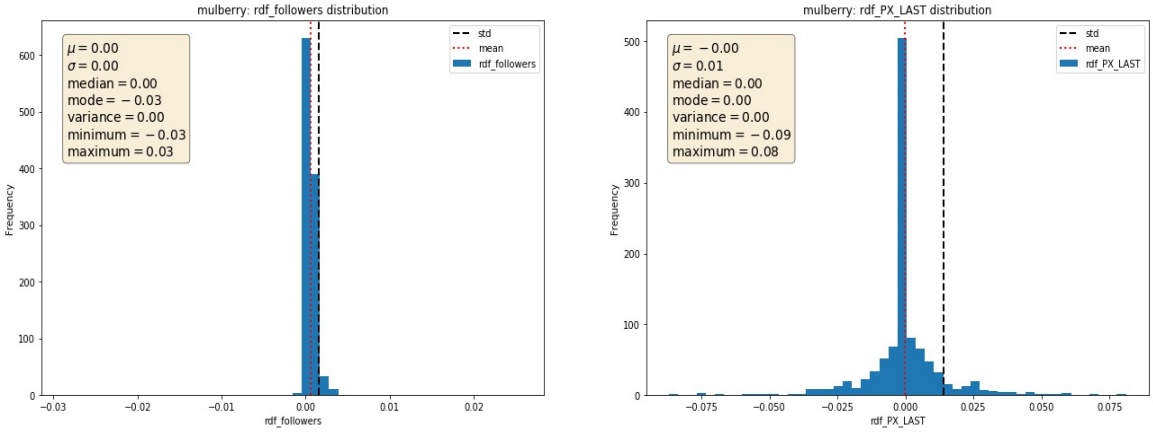
FIGURE A.41: Mulberry: shows the frequency distributions of a change in logarithmic transforms of followers (A) and stock prices (B) (2014-2018).

Ralph Lauren

The Figures A.51-A.55 show the frequency distributions of a selection of features for *Ralph Lauren* that are assumed to be normally distributed and used for the further analysis.

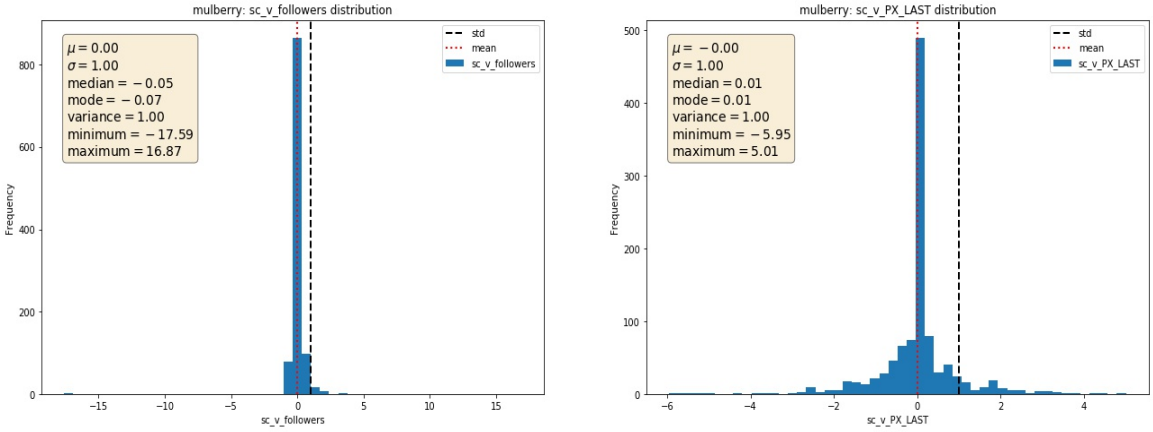
The Figure A.51 shows the frequency distributions of a change in logarithmic transforms of followers (A) and stock prices (B) for *Ralph Lauren*.

The Figure A.52 shows the frequency distributions of a relative change in followers (A) and stock prices (B) for *Ralph Lauren*.



(A) (B)

FIGURE A.42: Mulberry: shows the frequency distributions of a relative change in followers (A) and stock prices (B) (2014-2018).



(A) (B)

FIGURE A.43: Mulberry: shows the frequency distributions of followers velocity (A) and stock prices velocity (B) (2014-2018).

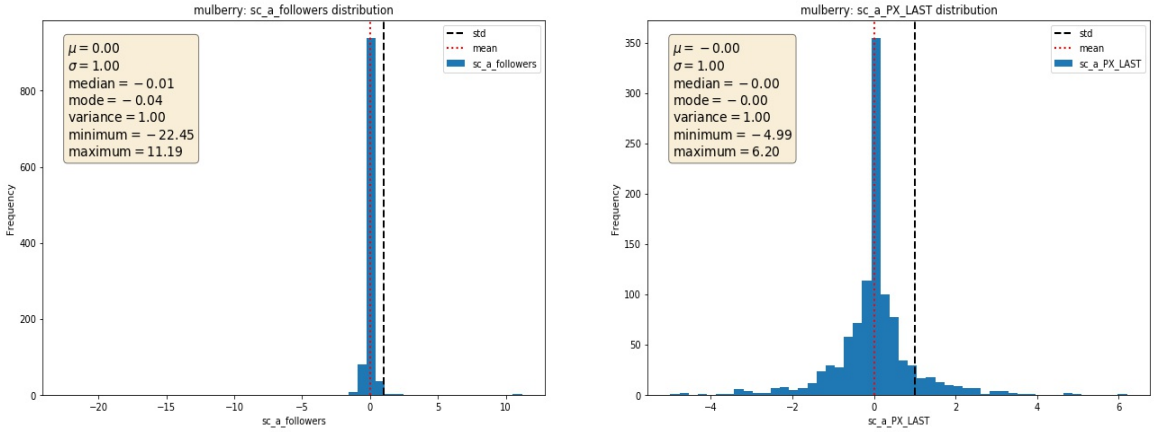
The Figure A.53 shows the frequency distributions of followers velocity (A) and stock prices velocity (B) for *Ralph Lauren*.

The Figure A.54 shows the frequency distributions of followers acceleration (A) and stock prices acceleration (B) for *Ralph Lauren*.

The Figure A.55 shows the frequency distributions of the logarithmic transform of returns for *Ralph Lauren*.

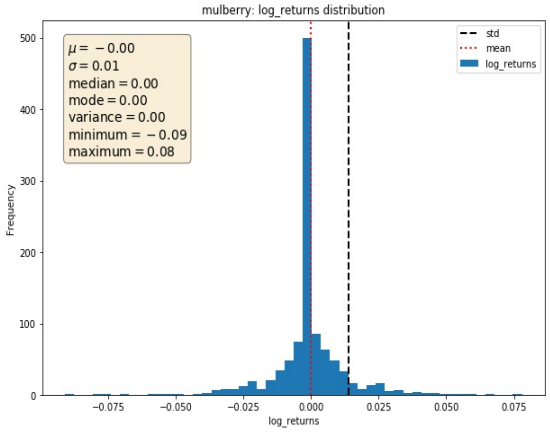
Under Armour

The Figures A.56-A.60 show the frequency distributions of a selection of features for *Under Armour* that are assumed to be normally distributed and used for the further



(A) (B)

FIGURE A.44: Mulberry: shows the frequency distributions of followers acceleration (A) and stock prices acceleration (B) (2014-2018).



(A)

FIGURE A.45: Mulberry: shows the frequency distributions of the logarithmic transform of returns (2014-2018).

analysis.

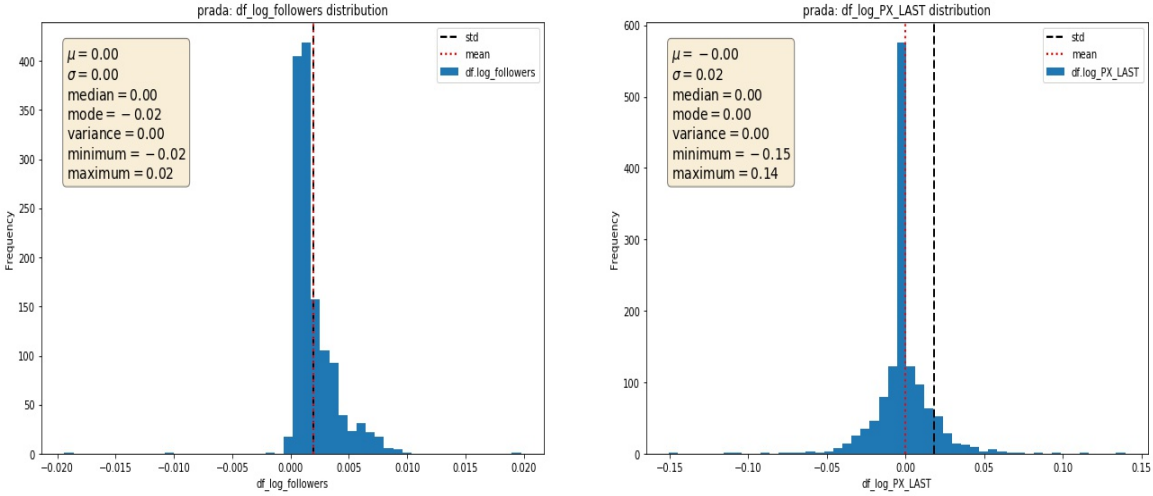
The Figure A.56 shows the frequency distributions of a change in logarithmic transforms of followers (A) and stock prices (B) for *Under Armour*.

The Figure A.57 shows the frequency distributions of a relative change in followers (A) and stock prices (B) for *Under Armour*.

The Figure A.58 shows the frequency distributions of followers velocity (A) and stock prices velocity (B) for *Under Armour*.

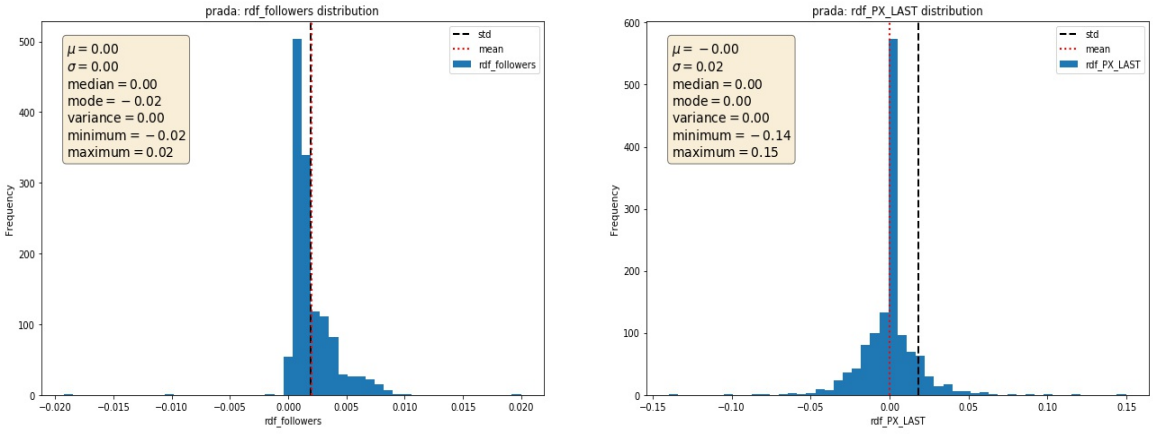
The Figure A.59 shows the frequency distributions of followers acceleration (A) and stock prices acceleration (B) for *Under Armour*.

The Figure A.60 shows the frequency distributions of the logarithmic transform of returns for *Under Armour*.



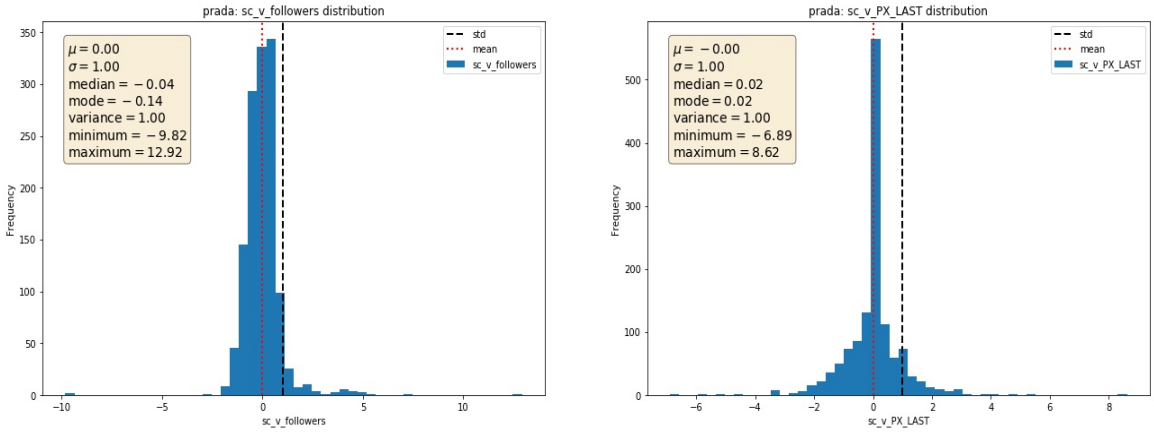
(A) (B)

FIGURE A.46: Prada: shows the frequency distributions of a change in logarithmic transforms of followers (A) and stock prices (B) (2014-2018).



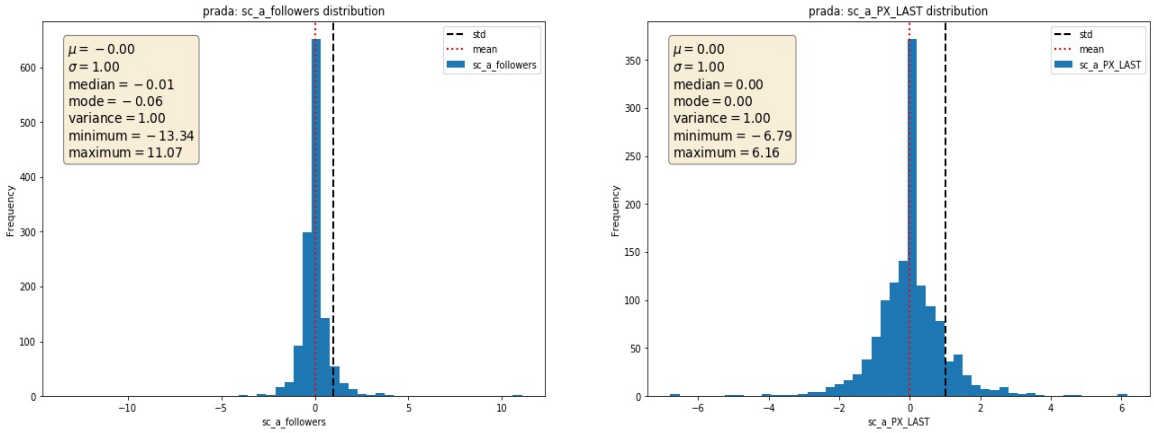
(A) (B)

FIGURE A.47: Prada: shows the frequency distributions of a relative change in followers (A) and stock prices (B) (2014-2018).



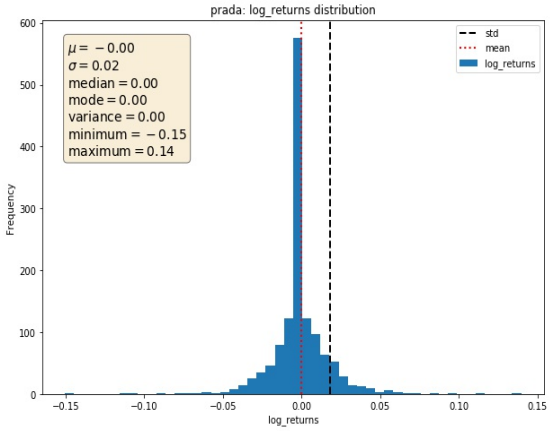
(A) (B)

FIGURE A.48: Prada: shows the frequency distributions of followers velocity (A) and stock prices velocity (B) (2014-2018).



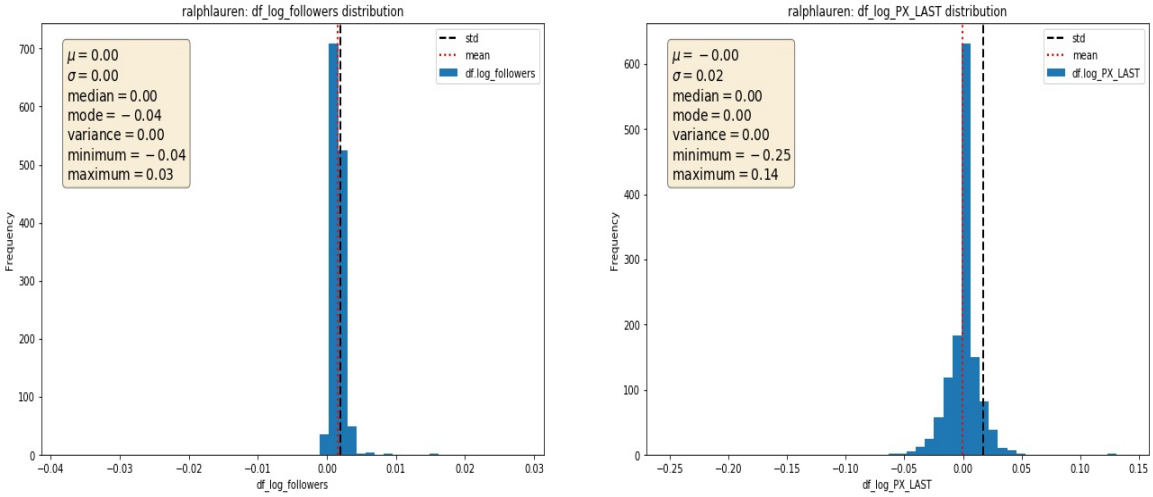
(A) (B)

FIGURE A.49: Prada: shows the frequency distributions of followers acceleration (A) and stock prices acceleration (B) (2014-2018).



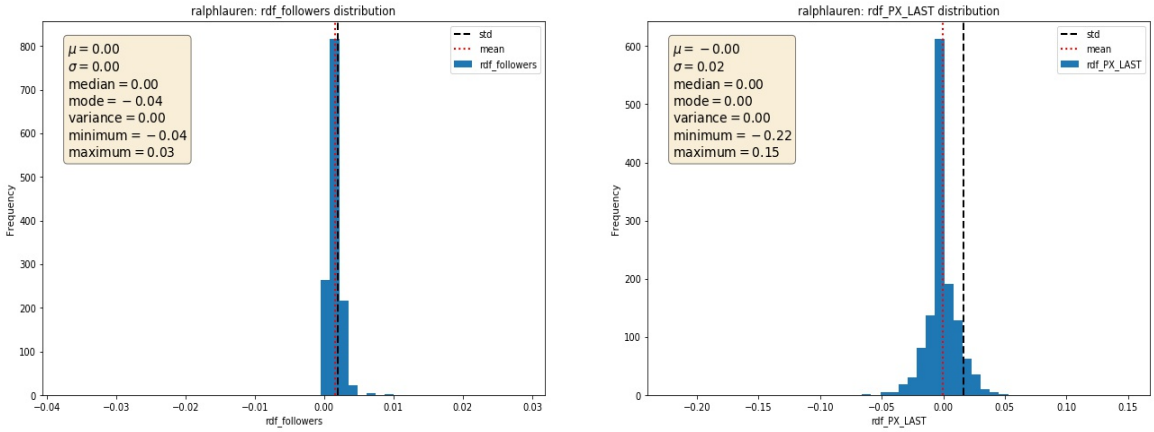
(A)

FIGURE A.50: Prada: shows the frequency distributions of the logarithmic transform of returns (2014-2018).



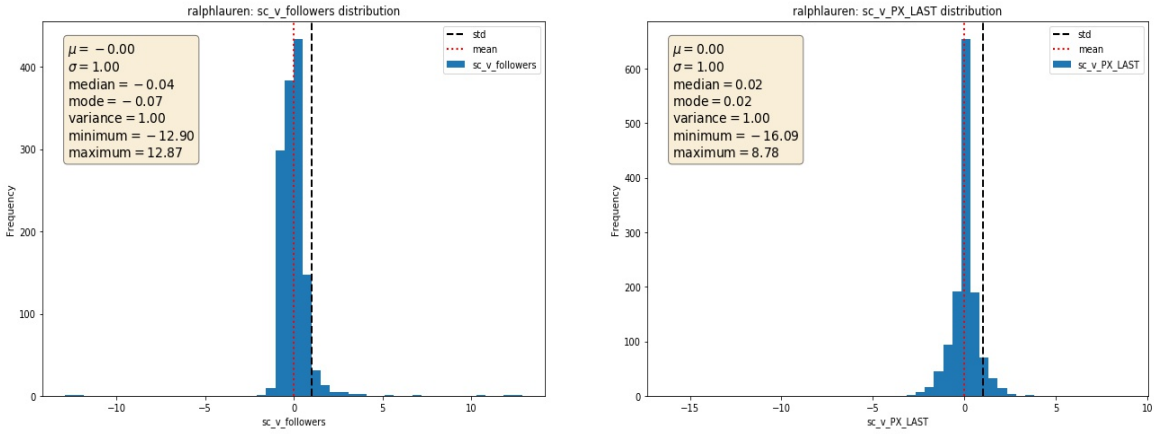
(A) (B)

FIGURE A.51: Ralph Lauren: shows the frequency distributions of a change in logarithmic transforms of followers (A) and stock prices (B) (2014-2018).



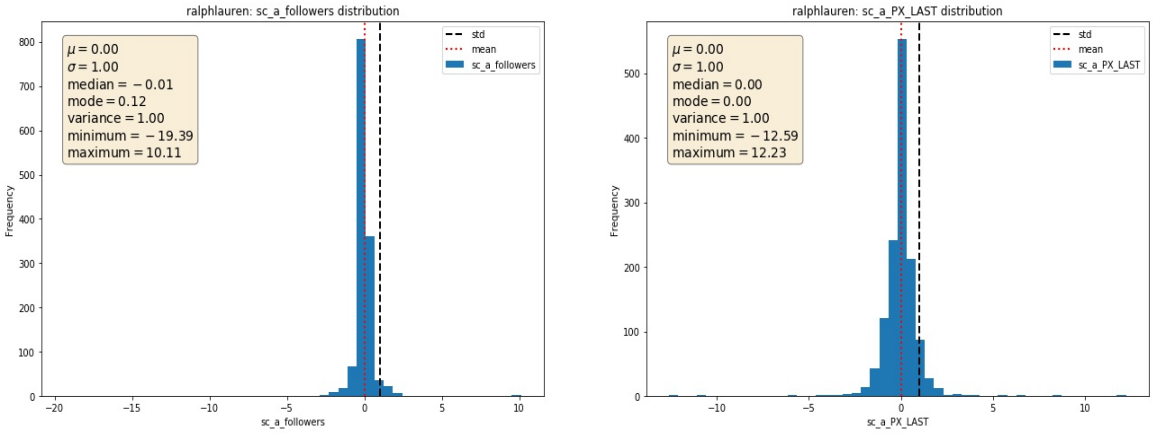
(A) (B)

FIGURE A.52: Ralph Lauren: shows the frequency distributions of a relative change in followers (A) and stock prices (B) (2014-2018).



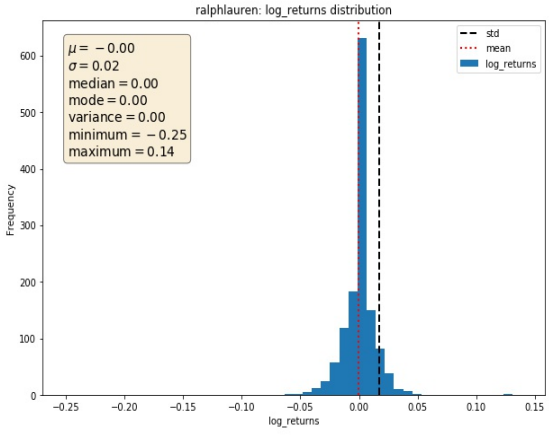
(A) (B)

FIGURE A.53: Ralph Lauren: shows the frequency distributions of followers velocity (A) and stock prices velocity (B) (2014-2018).



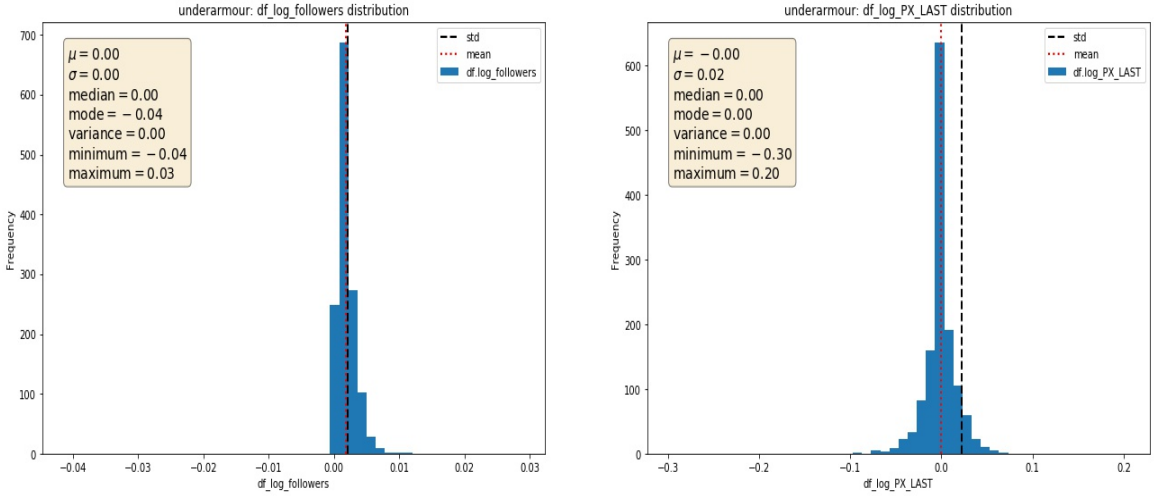
(A) (B)

FIGURE A.54: Ralph Lauren: shows the frequency distributions of followers acceleration (A) and stock prices acceleration (B) (2014-2018).



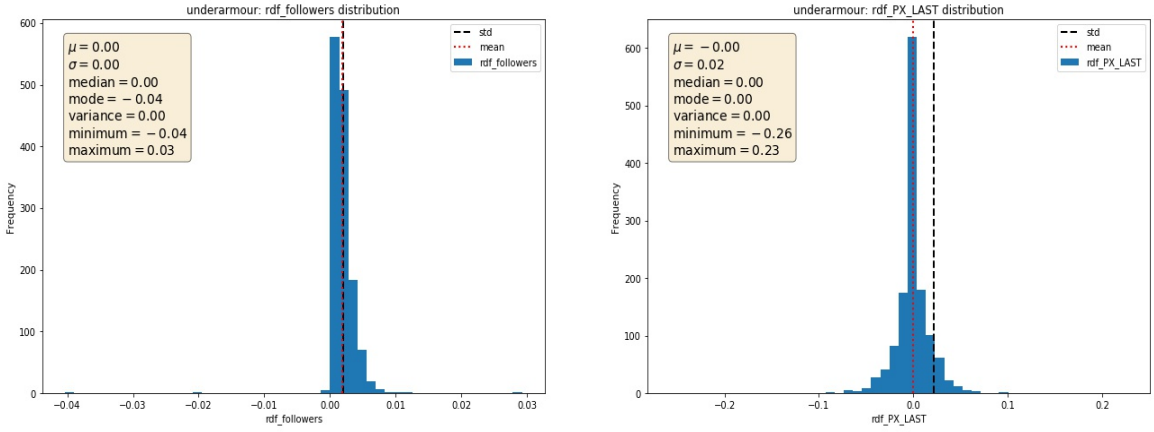
(A)

FIGURE A.55: Ralph Lauren: shows the frequency distributions of the logarithmic transform of returns (2014-2018).



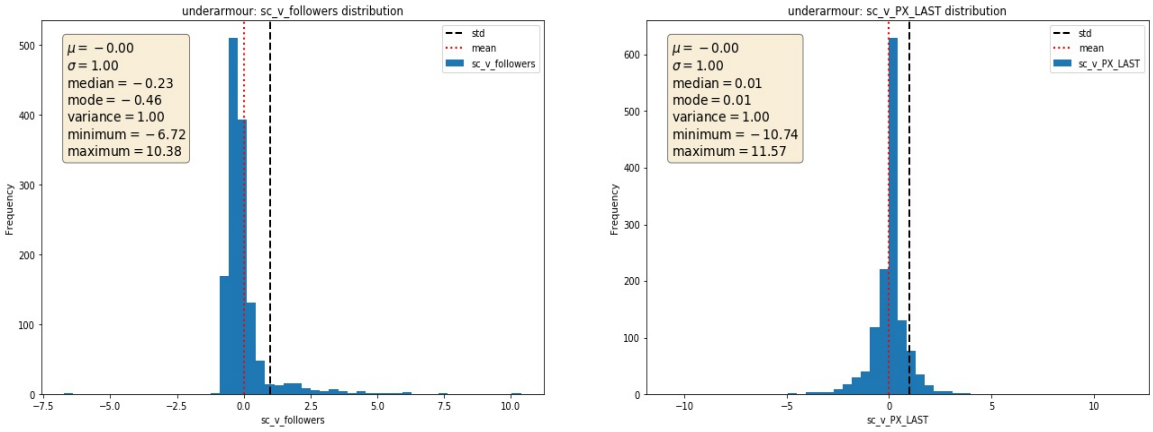
(A) (B)

FIGURE A.56: Under Armour: shows the frequency distributions of a change in logarithmic transforms of followers (A) and stock prices (B) (2014-2018).



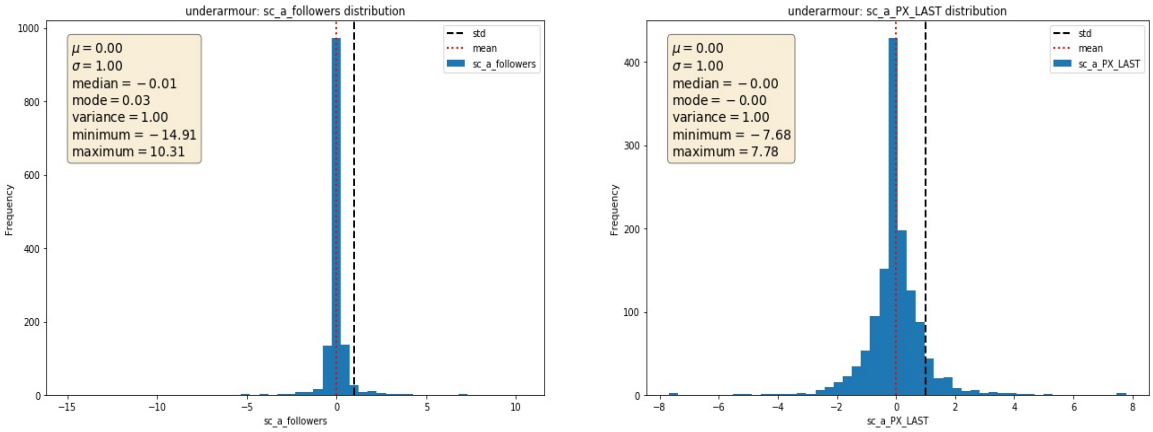
(A) (B)

FIGURE A.57: Under Armour: shows the frequency distributions of a relative change in followers (A) and stock prices (B) (2014-2018).



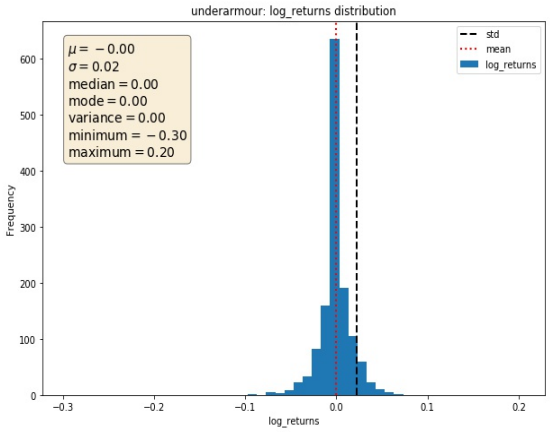
(A) (B)

FIGURE A.58: Under Armour: shows the frequency distributions of followers velocity (A) and stock prices velocity (B) (2014-2018).



(A) (B)

FIGURE A.59: Under Armour: shows the frequency distributions of followers acceleration (A) and stock prices acceleration (B) (2014-2018).



(A)

FIGURE A.60: Under Armour: shows the frequency distributions of the logarithmic transform of returns (2014-2018).

Bibliography

Aaker, D.A. (1991). *Managing Brand Equity*. Free Press, NY.

Akita R., Yoshihara A. Matsubara T. Uehara K. (2016). "Deep learning for stock prediction using numerical and textual information". In: *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, pp. 1–6. DOI: 10.1109/ICIS.2016.7550882.

Atkins A., Niranjan M. Gerding E. (2018). "Financial news predicts stock market volatility better than close price". In: *The Journal of Finance and Data Science* 4.2, pp. 120 –137. ISSN: 2405-9188. DOI: https://doi.org/10.1016/j.jfds.2018.02.002. URL: http://www.sciencedirect.com/science/article/pii/S240591881730048X.

Baker C., Nancarrow C. Tinson J. (2005). "The Mind versus Market Share Guide". In: pp. 523–540.

Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Springer.

Blei, D. M. (Apr. 2012). "Probabilistic Topic Models". In: *Commun. ACM* 55.4, pp. 77–84. ISSN: 0001-0782. DOI: 10.1145/2133806.2133826. URL: http://doi.acm.org/10.1145/2133806.2133826.

Blei D. M., Ng Andrew Y. Jordan M. I. (Mar. 2003). "Latent Dirichlet Allocation". In: *J. Mach. Learn. Res.* 3, pp. 993–1022. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=944919.944937.

Chang C.Y., Zhang Y. Teng Z. Bozanic Z. Ke B. (2016). "Measuring the Information Content of Financial News". In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. Osaka, Japan: The COLING 2016 Organizing Committee, pp. 3216–3225. URL: https://www.aclweb.org/anthology/C16-1303.

Chen H., De P. Hu Y. Hwang B. (2014). "Wisdom of Crowds: The Value of stock opinions transmitted through social media". In:

Christodoulides G., De Chernatony L. Furrer O. Shiu E. Abimbola T. (2006). "Conceptualising and Measuring the Equity of Online Brands". In: pp. 799–825.

Christodoulides, G. (2009). "Consumer Based Brand Equity Conceptualization and Measurements: Literature Review". In:

Craig MacKinlay, A (Feb. 1997). "Event Studies in Economics and Finance". In: *Journal of Economic Literature* 35, pp. 13–39.

Deerwester S. C., Dumais S. T. Furnas G. Harshman R. Landauer T. K. Lochbaum K. E. Streeter L. (1988). "Computer information retrieval using latent semantic structure: Us patent". In:

Deng S., Zhang N. Zhang W. Chen J. Z Pan J. Chen H. (Mar. 2019). "Knowledge-Driven Stock Trend Prediction and Explanation via Temporal Convolutional Network". In: DOI: 10.1145/3308560.3317701.

Devlin J., Chang M. Lee K. Toutanova K. (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *CoRR* abs/1810.04805. arXiv: 1810.04805. URL: http://arxiv.org/abs/1810.04805.

Dimpfl T., Jank S. (2016). "Can Internet Search Queries Help to Predict Stock Market Volatility?" In: pp. 171–192.

Ding X., Zhang Y. Liu T. Duan J. (Oct. 2014). "Using Structured Events to Predict Stock Price Movement: An Empirical Investigation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1415–1425. DOI: 10.3115/v1/D14-1148. URL: https://www.aclweb.org/anthology/D14-1148.

— (2015). "Deep Learning for Event-driven Stock Prediction". In: *Proceedings of the 24th International Conference on Artificial Intelligence*. IJCAI'15. Buenos Aires, Argentina: AAAI Press, pp. 2327–2333. ISBN: 978-1-57735-738-4. URL: http://dl.acm.org/citation.cfm?id=2832415.2832572.

— (2016). "Knowledge-Driven Event Embedding for Stock Prediction". In: *COLING*.

Eisner B., Rocktaschel T. Augenstein I. Bosnjak M. Riedel S. (2016). "emoji2vec: Learning Emoji Representations from their Description". In:

Elman, J. (1990). "Finding structure in time". In: pp. 179–211.

Ester M., Kriegel H. P. Sander J. Xu X. (1996). "A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD'96. Portland, Oregon: AAAI Press, pp. 226–231. URL: http://dl.acm.org/citation.cfm?id=3001460.3001507.

Felbo B., Mislove A. Sogaard A. Rahwan I. Lehmann S. (2017). "Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm". In: pp. –.

Feldwick, P. (1996). "What is brand equity anyway, and how do you measure it?" In: pp. 85–104.

Feuerriegel S., Pröllochs N. (2018). "Investor Reaction to Financial Disclosures Across Topics: An Application of Latent Dirichlet Allocation". In: *CoRR* abs/1805.03308. arXiv: 1805.03308. URL: http://arxiv.org/abs/1805.03308.

Feuerriegel S., Ratku A. Neumann D. (2016). "Analysis of How Underlying Topics in Financial News Affect Stock Prices Using Latent Dirichlet Allocation". In: *2016 49th Hawaii International Conference on System Sciences (HICSS)*, pp. 1072–1081. DOI: 10.1109/HICSS.2016.137.

Graham, B. (1949). *The intelligent investor : a book of practical counsel*. MIT Press.

Han S., Hao X. Huang H. (2018). "An event-extraction approach for business analysis from online Chinese news". In: *Electronic Commerce Research and Applications* 28, pp. 244 –260. ISSN: 1567-4223. DOI: https://doi.org/10.1016/j.elerap.2018.02.006. URL: http://www.sciencedirect.com/science/article/pii/S1567422318300243.

Harris, Zellig S. (1981). "Distributional Structure". In: *Papers on Syntax*. Dordrecht: Springer Netherlands, pp. 3–22. ISBN: 978-94-009-8467-7. DOI: 10.1007/978-94-009-8467-7_1. URL: https://doi.org/10.1007/978-94-009-8467-7_1.

Heidari S., Alborzi M. Radfar R. Afsharkazemi M. A. Rajabzadeh Ghatari A. (2019). "Big data clustering with varied density based on MapReduce". In: *Journal of Big Data* 6.1, p. 77. ISSN: 2196-1115. DOI: 10.1186/s40537-019-0236-x. URL: https://doi.org/10.1186/s40537-019-0236-x.

Hochreiter S., Schmidhuber J. (1997). "Long short-term memory". In: pp. 1735–1780.

Hu Z., Liu W. Bian J. Liu X. Liu T. (Dec. 2017). "Listening to Chaotic Whispers: A Deep Learning Framework for News-oriented Stock Trend Prediction". In: DOI: 10.1145/3159652.3159690.

Huynh H., Dang L. M. Duong D. (Dec. 2017). "A New Model for Stock Price Movements Prediction Using Deep Neural Network". In: pp. 57–62. DOI: 10.1145/3155133.3155202.

Jiamiao Wang Xindong Wu, Lei Li (2017). *Semantic Connection Based Topic Evolution*. URL: https://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14399/14242.

Kim, Y. (2014). "Convolution Neural Networks for Sentence Classification". In:

Konchitchki Y., O'Leary D. E. (2011). "Event study methodologies in information systems research". In: *Int. J. Accounting Inf. Systems* 12, pp. 99–115.

Lavrenko V., Schmill M. Lawrie D. Ogilvie P. Jensen D. Allan J. (2000). "Mining of Concurrent Text and Time Series". In: *In Proceedings of the 6th ACM SIGKDD International Conference of Knowledge Discovery and Data Mining Workshop on Text Mining*, pp. 37–44.

Le Q. V., Mikolov T. (2014). "Distributed Representations of Sentences and Documents". In: *CoRR* abs/1405.4053. arXiv: 1405.4053. URL: http://arxiv.org/abs/1405.4053.

Li Y., Jin T. Xi M. Liu S. Luo Z. (2018). "Massive Text Mining for Abnormal Market Trend Detection". In: *2018 IEEE International Conference on Big Data (Big Data)*, pp. 4135–4141.

Li X., Xie H. Song Y. Zhu S. Li Q. Wang F. L. (2015). "Does Summarization Help Stock Prediction? A News Impact Analysis". In: *IEEE Intelligent Systems* 30.3, pp. 26–34. ISSN: 1541-1672. DOI: 10.1109/MIS.2015.1.

Liu Q., Cheng X. Su S. Zhu S. (2018a). "Hierarchical Complementary Attention Network for Predicting Stock Price Movements with News". In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. CIKM

'18. Torino, Italy: ACM, pp. 1603–1606. ISBN: 978-1-4503-6014-2. DOI: 10.1145/3269206.3269286. URL: http://doi.acm.org/10.1145/3269206.3269286.

Liu D., Li Y. Thomas M. (2017). "A road map for Natural Language Processing research in Information Systems". In:

Liu G., Wang X. (2019). "A Numerical-Based Attention Method for Stock Market Prediction With Dual Information". In: *IEEE Access* 7, pp. 7357–7367. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2886367.

Liu Y., Zeng Q. Yang H. Carrio A. (2018b). "Stock Price Movement Prediction from Financial News with Deep Learning and Knowledge Graph Embedding". In: *Knowledge Management and Acquisition for Intelligent Systems*. Cham: Springer International Publishing, pp. 102–113. ISBN: 978-3-319-97289-3.

Loughran T., McDonald B. (2011). "When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks". In: *The Journal of Finance* 66.1, pp. 35–65. DOI: 10.1111/j.1540-6261.2010.01625.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.2010.01625.x.

Luhn, H. P. (1957). "A statistical approach to the mechanized encoding and searching of literary information". In: pp. 309–317.

Luo X., Zhang J. Duan W. (2013). "Social Media and Firm Equity Value". In: pp. 146–163.

Luss R., D'Aspremont A. (2009). "Predicting abnormal returns from news using text classification". In: *Quantitative Finance* 15.6, pp. 999–1012. DOI: 10.1080/14697688.2012.672762. URL: https://doi.org/10.1080/14697688.2012.672762.

MediaKix (2019). "Influencer Marketing 2019: Key Statistics from Our Influencer Marketing Survey". In: URL: https://mediakix.com/influencer-marketing-resources/influencer-marketing-industry-statistics-survey-benchmarks/.

Merello S., Picasso A. Ma Y. Oneto L. Cambria E. (2018). "Investigating Timing and Impact of News on the Stock Market". In: *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 1348–1354. DOI: 10.1109/ICDMW.2018.00191.

Mikolov T., Chen K. Corrado G. Dean J. (2013). "Efficient estimation of word representations in vector space". In:

Miller, G. A. (Nov. 1995). "WordNet: A Lexical Database for English". In: *Commun. ACM* 38.11, pp. 39–41. ISSN: 0001-0782. DOI: 10.1145/219717.219748. URL: http://doi.acm.org/10.1145/219717.219748.

Miller H., Thebault-Spieker J. Chang S. Johnson I. (2016). "'Blissfully happy' or 'ready to fight': Varying Interpretations of Emoji". In:

Murphy, K. (2012). *Machine Learning. A Probabilistic Perspective*. MIT Press.

Nofer M., Hinz O. (2014). "Are crowds on the internet wiser than experts? The case of a stock prediction comunity". In:

O'Connor A., J. (2012). "The Power of Popularity: An Empirical Study of the Relationship Between Social Media Fan Counts and Brand Company Stock Prices". In: pp. 229–235.

Oncharoen P., Vateekul P. (2018). "Deep Learning for Stock Market Prediction Using Event Embedding and Technical Indicators". In: *2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA)*, pp. 19–24.

Park C.S., Srinivasan V. (1994). "A Survey-Based Method for Measuring and Understanding Brand Equity and its Extendibility". In: pp. 271–288.

Porter, M. F. (1980). "An algorithm for suffix stripping". In: pp. 130–137.

Preis T., Moat H. S. Curme C. Avakian A. Kenett D. Stanley H. E. (2013). "Quantifying Wikipedia Usage Patterns Before Stock Market Moves". In:

Radford, A. (2018). "Improving Language Understanding by Generative Pre-Training". In:

Raggio D., R. Leone P. R. (2007). "The theoretical separation of brand equity and brand value: Managerial implications for strategic planning". In:

Ranco G., Bordino I. Bormetti G. Caldarelli G. Lilo F. Treccani M. (2016). "Coupling News Sentiment with Web Browsing Data Improves Prediction of Intra-Day Proce Dynamics". In:

Research, Cogent (2008). "Social media's impact on personal finance and investing". In: URL: http://www.cogentresearch.com.

Research, Quintly (2017). *Use of Emojis Can Lead to 47.7Instagram*. Tech. rep. Accessed May 2018. URL: https://www.quintly.com/blog/2017/01/instagram-emoji-study-higher-interactions.

Romero P.J., Balch T. (2014). *What Hedge Funds Really Do. An Introduction to Portfolio Management*. Business Expert Press.

Schumaker R. P., Chen H. (2009). "Textual Analysis of Stock Market Prediction Using Breaking Financial News: The AZFin Text System". In: *ACM Trans. Inf. Syst.* 27.2, 12:1–12:19. ISSN: 1046-8188. DOI: 10.1145/1462198.1462204. URL: http://doi.acm.org/10.1145/1462198.1462204.

Shankar V., Azar P. Fuller M. (2008). "BRAN*EQT: Multi-category Brand Equity Model and its Application at Allstate". In: pp. 567–584.

Silverman, Bernard W (2018). *Density estimation for statistics and data analysis*. Routledge.

Sohangir S., Wang D. Pomeranets A. Khoshgoftaar T.M. (2018). "Big Data: Deep Learning for financial sentiment analysis". In:

Sparck Jones, K. (1964). "Synonymy and Semantic Classification". In:

— (1972). "A statistical interpretation of term specificity and its application in retrieval". In: *Journal of Documentation* 28, pp. 11–21.

Srivastava R.K., Shervani T.A. Fahey L. (1998). "Market-Based Assets and Shareholder Value: A Framework for Analysis". In: pp. 2–18.

Surowiecki, J. (2004). *The Wisdom of Crowds. Why the Many Are Smarter Than the Few*. Anchor Doubleday, USA.

Swait J. Erdem, T. Louviere J. Dubelaar C. (1993). "The Equalization Price: A Measure of Consumer - Perceived Brand Equity". In: pp. 23–45.

Tan J., Wang J. Rinprasertmeechai D. Xing R. Li Q. (2019). "A Tensor-based eLSTM Model to Predict Stock Price using Financial News". In: *Proceedings of the 52nd Hawaii International Conference on System Sciences*.

Tetlock, P. C. (2007). "Giving Content to Investor Sentiment: The Role of Media in the Stock Market". In: *The Journal of Finance* 62.3, pp. 1139–1168. DOI: 10.1111/j.1540-6261.2007.01232.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1540-6261.2007.01232.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.2007.01232.x.

Umezu, Y. (2017). "Can AI Improve Portfolio Managers' Investment Decision Making?" In:

Vargas M. R., de Lima B. S. L. P. Evsukoff A. G. (2017). "Deep learning for stock market prediction from financial news articles". In: *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, pp. 60–65. DOI: 10.1109/CIVEMSA.2017.7995302.

Vaswani A., Shazeer N. Parmar N. Uszkoreit J. Jones L. Gomez A. N. Kaiser L. Polosukhin I. (2017). "Attention Is All You Need". In: *CoRR* abs/1706.03762. arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762.

Wijeratne S., Balasuriya L. Sheth A. Doran D. (2017). "A semantics-based measure of emoji similarity". In:

Yang H., Chen Y. Liu K. Xiao Y. Zhao J. (2018). "DCFEE: A Document-level Chinese Financial Event Extraction System based on Automatically Labeled Training Data". In: *Proceedings of ACL 2018, System Demonstrations*. Melbourne, Australia: Association for Computational Linguistics, pp. 50–55. DOI: 10.18653/v1/P18-4009. URL: https://www.aclweb.org/anthology/P18-4009.

Yang L., Zhang Z. Xiong S. Wei L. Ng J. Xu L. Dong R. (2019). "Explainable Text-Driven Neural Network for Stock Prediction". In: *CoRR* abs/1902.04994. arXiv: 1902.04994. URL: http://arxiv.org/abs/1902.04994.

Yoo B., Donthu N. (2001). "Developing and Validating a Multidimensional Consumer-Based Brand Equity Scale". In: pp. 1–14.

Yu Y., Duan W. Cao Q. (2012). "The impact of social media and conventional media on firm equity value: A sentiment analysis approach". In:

Zhang X., Zhang Y. Wang S. Yao Y. Fang B. Yu P. (2018). "Improving Stock Market Prediction via Heterogeneous Information Fusion". In: *CoRR* abs/1801.00588. arXiv: 1801.00588. URL: http://arxiv.org/abs/1801.00588.

Zhou X., Yang Wang W. (2018). "MojiTalk: Generating emotional responses at scale". In: