Aalto University
School of Science
Master’s Programme in ICT Innovation – EIT Digital Master School

Rohit Saluja

Interpreting Multivariate Time Series for an Organization Health Platform

Master’s Thesis

Espoo, January 08, 2021

Supervisor:	Professor Quan Zhou, Aalto University Professor Anders Västberg, KTH Royal Institute of Technology
Thesis advisor(s):	Dr. Avleen Malhi, Aalto University Dr. Cicek Cavdar, KTH Royal Institute of Technology

AALTO UNIVERSITY
School of Science
Master’s Programme in ICT Innovation

ABSTRACT OF THE
MASTER´S THESIS

Author: Rohit Saluja

Title of the thesis: Interpreting Multivariate Time Series for an Organization Health Platform

Number of pages: xiii + 61

Date: 08/01/2021

Major or Minor: Autonomous Systems

Supervisor: Professor Quan Zhou, Aalto University
Professor Anders Västberg, KTH Royal Institute of Technology

Thesis advisor: Dr. Avleen Malhi, Aalto University
Professor Cicek Cavdar, KTH Royal Institute of Technology

Machine learning-based systems are rapidly becoming popular because it has been realized that machines are more efficient and effective than humans at performing certain tasks. Although machine learning algorithms are extremely popular, they are also very literal and undeviating. This has led to a huge research surge in the field of interpretability in machine learning to ensure that machine learning models are reliable, fair, and can be held liable for their decision-making process. Moreover, in most real-world problems just making predictions using machine learning algorithms only solves the problem partially. Time series is one of the most popular and important data types because of its dominant presence in the fields of business, economics, and engineering. Despite this, interpretability in time series is still relatively unexplored as compared to tabular, text, and image data. With the growing research in the field of interpretability in machine learning, there is also a pressing need to be able to quantify the quality of explanations produced after interpreting machine learning models. Due to this reason, evaluation of interpretability is extremely important. The evaluation of interpretability for models built on time series seems completely unexplored in research circles. This thesis work focused on achieving and evaluating model agnostic interpretability in a time series forecasting problem. The use case discussed in this thesis work focused on finding a solution to a problem faced by a digital consultancy company. The digital consultancy wants to take a data-driven approach to understand the effect of various sales related activities in the company on the sales deals closed by the company. The solution involved framing the problem as a time series forecasting problem to predict the sales deals and interpreting the underlying forecasting model. The interpretability was achieved using two novel model agnostic interpretability techniques, Local interpretable model- agnostic explanations (LIME) and Shapley additive explanations (SHAP). The explanations produced after achieving interpretability were evaluated using human evaluation of interpretability. The results of the human evaluation studies clearly indicate that the explanations produced by LIME and SHAP greatly helped lay humans in understanding the predictions made by the machine learning model. The human evaluation study results also indicated that LIME and SHAP explanations were almost equally understandable with LIME performing better but with a very small margin. The work done during this project can easily be extended to any time series forecasting or classification scenario for achieving and evaluating interpretability. Furthermore, this work can offer a very good framework for achieving and evaluating interpretability in any machine learning-based regression or classification problem.

Keywords: Interpretability, Forecasting, LIME, SHAP, Time series, Explainable artificial intelligence

Publishing language: English

Interpreting Multivariate Time Series for an Organization Health Platform

Rohit Saluja

12/18/2020

Master's Thesis

Examiner
Dr. Anders Västberg

Supervisor
Dr. Cicek Cavdar

Academic Advisor at Aalto University
Dr. Avleen Malhi

KTH Royal Institute of Technology
School of Electrical Engineering and Computer Science (EECS)
Department of Computer Science
SE-100 44 Stockholm, Sweden

Abstract

Machine learning-based systems are rapidly becoming popular because it has been realized that machines are more efficient and effective than humans at performing certain tasks. Although machine learning algorithms are extremely popular, they are also very literal and undeviating. This has led to a huge research surge in the field of interpretability in machine learning to ensure that machine learning models are reliable, fair, and can be held liable for their decision-making process. Moreover, in most real-world problems just making predictions using machine learning algorithms only solves the problem partially. Time series is one of the most popular and important data types because of its dominant presence in the fields of business, economics, and engineering. Despite this, interpretability in time series is still relatively unexplored as compared to tabular, text, and image data. With the growing research in the field of interpretability in machine learning, there is also a pressing need to be able to quantify the quality of explanations produced after interpreting machine learning models. Due to this reason, evaluation of interpretability is extremely important. The evaluation of interpretability for models built on time series seems completely unexplored in research circles. This thesis work focused on achieving and evaluating model agnostic interpretability in a time series forecasting problem.

The use case discussed in this thesis work focused on finding a solution to a problem faced by a digital consultancy company. The digital consultancy wants to take a data-driven approach to understand the effect of various sales related activities in the company on the sales deals closed by the company. The solution involved framing the problem as a time series forecasting problem to predict the sales deals and interpreting the underlying forecasting model. The interpretability was achieved using two novel model agnostic interpretability techniques, Local interpretable model- agnostic explanations (LIME) and Shapley additive explanations (SHAP). The explanations produced after achieving interpretability were evaluated using human evaluation of interpretability. The results of the human evaluation studies clearly indicate that the explanations produced by LIME and SHAP greatly helped lay humans in understanding the predictions made by the machine learning model. The human evaluation study results also indicated that LIME and SHAP explanations were almost equally understandable with LIME performing better but with a very small margin. The work done during this project can easily be extended to any time series forecasting or classification scenario for achieving and evaluating interpretability. Furthermore, this work can offer a very good framework for achieving and evaluating interpretability in any machine learning-based regression or classification problem.

Keywords

Interpretability, Forecasting, Local interpretable model-agnostic explanations, Shapley additive explanations, Time series, Explainable artificial intelligence

Sammanfattning

Maskininlärningsbaserade system blir snabbt populära eftersom man har insett att maskiner är effektivare än människor när det gäller att utföra vissa uppgifter. Även om maskininlärningsalgoritmer är extremt populära, är de också mycket bokstavliga. Detta har lett till en enorm forskningsökning inom området tolkbarhet i maskininlärning för att säkerställa att maskininlärningsmodeller är tillförlitliga, rättvisa och kan hållas ansvariga för deras beslutsprocess. Dessutom löser problemet i de flesta verkliga problem bara att göra förutsägelser med maskininlärningsalgoritmer bara delvis. Tidsserier är en av de mest populära och viktiga datatyperna på grund av dess dominerande närvaro inom affärsverksamhet, ekonomi och teknik. Trots detta är tolkningsförmågan i tidsserier fortfarande relativt outforskad jämfört med tabell-, text- och bilddata. Med den växande forskningen inom området tolkbarhet inom maskininlärning finns det också ett stort behov av att kunna kvantifiera kvaliteten på förklaringar som produceras efter tolkning av maskininlärningsmodeller. Av denna anledning är utvärdering av tolkbarhet extremt viktig. Utvärderingen av tolkbarhet för modeller som bygger på tidsserier verkar helt outforskad i forskarkretsar. Detta uppsatsarbete fokuserar på att uppnå och utvärdera agnostisk modelltolkbarhet i ett tidsserieprognosproblem.

Fokus ligger i att hitta lösningen på ett problem som ett digitalt konsultföretag står inför som användningsfall. Det digitala konsultföretaget vill använda en datadriven metod för att förstå effekten av olika försäljningsrelaterade aktiviteter i företaget på de försäljningsavtal som företaget stänger. Lösningen innebar att inrama problemet som ett tidsserieprognosproblem för att förutsäga försäljningsavtalen och tolka den underliggande prognosmodellen. Tolkningsförmågan uppnåddes med hjälp av två nya tekniker för agnostisk tolkbarhet, lokala tolkbara modellagnostiska förklaringar (LIME) och Shapley additiva förklaringar (SHAP). Förklaringarna som producerats efter att ha uppnått tolkbarhet utvärderades med hjälp av mänsklig utvärdering av tolkbarhet. Resultaten av de mänskliga utvärderingsstudierna visar tydligt att de förklaringar som produceras av LIME och SHAP starkt hjälpte människor att förstå förutsägelserna från maskininlärningsmodellen. De mänskliga utvärderingsstudieresultaten visade också att LIME- och SHAP-förklaringar var nästan lika förståeliga med LIME som presterade bättre men med en mycket liten marginal. Arbetet som utförts under detta projekt kan enkelt utvidgas till alla tidsserieprognoser eller klassificeringsscenarier för att uppnå och utvärdera tolkbarhet. Dessutom kan detta arbete erbjuda en mycket bra ram för att uppnå och utvärdera tolkbarhet i alla maskininlärningsbaserade regressions- eller klassificeringsproblem.

Nyckelord

Tolkbarhet, Prognoser, Lokala tolkningsbara modell-agnostiska förklaringar, Shapley additiva förklaringar, Tidsserier, Förklarbar artificiell intelligens

Acknowledgments

I am very thankful to my examiner Anders Västberg for all his guidance and support throughout the thesis work. His guidance was pivotal in overcoming the major challenges I faced during my thesis. I would also like to sincerely thank my supervisor Cicek Cavdar for all her valuable inputs. Her suggestions were crucial in shaping my thesis work. I would like to thank Avleen Malhi, my academic advisor at Aalto University for all the encouragement and guidance. She was always kind enough to take time out of her busy schedule to address all my questions and provide valuable feedback which drastically improved the quality of my work. I have learned a lot from her.

I am grateful to Futurice for letting me conduct my thesis work at their company. They always provided me with a friendly and supportive environment.

Special thanks to Samanta Knapič for sharing her knowledge about the human evaluation of interpretability with me.

I would also like to thank my friends: Abhishek Mahajan, Prabhat Sharma, Mahima Bhutani, and Siddhant Gupta for all their help and support.

Most importantly, I am indebted to my parents for their constant support through the thick and thin of life. They have always stood behind me resolutely and encouraged me to keep aiming higher. Whatever little I have managed to achieve would not have been possible without their unconditional care and support.

Stockholm, October 2020
Rohit Saluja

Table of contents

List of Figures

List of Tables

List of acronyms and abbreviations

AI	Artificial Intelligence
ECG	Electrocardiogram
GDPR	General Data Protection Regulation
KNN	K Nearest Neighbors
LIME	Local Interpretable Model-Agnostic Explanations
ML	Machine Learning
LSTM	Long Short-Term Memory
RBF	Radial Basis Function
SHAP	SHapley Additive exPlanations
SVM	Support Vector Machine
SVR	Support Vector Regression
XAI	Explainable Artificial Intelligence

1 Introduction

This chapter describes the specific problem that this thesis addresses, the context of the problem, the goals of this thesis project, research methodology and outlines the structure of the thesis. The chapter also sheds some light on the delimitations and the ethical, sustainability, and societal issues related to this work.

1.1 Background

Machine learning is rapidly seeping into almost every industry as it has been realized that machines are faster, more efficient, and effective than humans at performing specific tasks [1, p. 1]. Industries, where machine learning has gained a lot of popularity, include the healthcare, financial and retail industries. However, building predictive machine learning models that capture complex relationships of the data, using them to make predictions, and evaluating the models using an accuracy metric only solves a part of the problem. Humankind as a curiosity-driven species has an innate desire to understand the reasoning behind the decisions made by a machine learning model. Doshi-Velez and Kim [1, pp. 3-4] in their research work state that building a machine learning model that performs well at making decisions only offers an incomplete description of most real-world tasks. It is also important to understand the *why* behind the decision made by a machine learning model as opposed to just finding out the decision made because it only partially solves the real-world problem. Another problem highlighted by Ribeiro et al. [2] is that humans will not use a machine learning model that they don't understand because of trust issues.

Due to these reasons, recent years have witnessed a lot of criticism towards machine learning. This has pushed researchers to work on understanding the decisions made by machine learning models. One way that can help humans understand the decisions made by machine learning models is by using interpretability techniques. According to Miller [3], interpretability is defined as "the degree to which a human can understand the cause of a decision".

Interpretability in machine learning can be achieved in the following ways. It can either be achieved by building machine learning models that are inherently interpretable due to their simple structure. For example, models built with linear regression, logistic regression, or short decision trees are inherently interpretable. Another way is to interpret a model after it has been trained. This is called post hoc or model agnostic interpretability [4, p. 25]. The different kinds of tools/techniques that are used to interpret a model can be broadly divided into two categories. The first category includes model-specific interpretability tools that only work with specific models. Since the methods used to interpret intrinsically interpretable models are limited to these models, they by default fall under the category of model-specific interpretability tools. The second category includes model agnostic tools that are used to interpret models post training (post hoc) and can be used to interpret any machine learning model. Model agnostic tools are generally used to interpret machine learning models that are complex in nature. These complex models are also referred to as black-box models [4, p. 26]. Interpretability tools can produce two kinds of interpretations, either local or global. Global interpretability helps in understanding the overall effect of input values on the output of the machine learning model. Local interpretability helps in understanding the effect of input values on the output produced by the machine learning model for a particular instance in the dataset. A good human-understandable output produced after interpreting the model is called an explanation [4, pp. 26-29].

The different types of data that are fed into machine learning algorithms can be categorized into five broad categories - numerical data, categorical data, text data, image data, and time series data. Time series contains data points measured over a period of time and can be differentiated from the other data types based on the time dimension attached to it [5]. Time series holds a lot of importance because it is the most dominant type of data being generated in the fields of business, economics, science, and engineering as most of the measurements are performed over time. Common examples of time series data include the stock data generated in the financial markets, the data generated by various sensors in mobile phones, data generated by various smart home devices, data generated by autonomous vehicles about its environment, data collected by most companies about its sales and electrocardiogram (ECG) data generated in the field of medicine.

A time series that consists of a single value sequentially measured over a period of time is called a univariate time series whereas a time series that consists of two of more values sequentially measured over a period of time is called a multivariate time series [6, p. 16].

Time series forecasting is the process of understanding the past and using it to predict the future. During the process of time series forecasting, the historical observations of the time series are used to build models capable of predicting the future values of the series.

This project was conducted in a company called Futurice at their headquarters in Helsinki, Finland. Futurice is a multinational digital innovation and engineering company with 8 offices spread across five countries. Futurice was started in the year 2000, has experienced 3000+ projects and is a work home to 600+ employees. The project was conducted under the Exponential AI team that is driving the objective of making Futurice a data-driven organization.

1.2 Problem

Due to the presence of time series data in every major industry, there is also an increasing need for understanding the decisions made by models trained on time series data.

However, interpretability with time series data can be challenging due to the complex non-linear temporal dependencies in the data. The elements of time series data are connected to each other with the time dimension and the sequence of the data in a time series data becomes extremely important. While working with time series data, just identifying the features leading to the decision of a machine learning model is not enough. It is also extremely important to identify how the features from different points in time affect the decision of the machine learning model [7].

Simple models like linear regression and short decision trees can be used to achieve interpretability but they might not be sufficient to capture complex relationships of the underlying data. Complex black-box machine learning models can be used to achieve good accuracy on the data but might not be interpretable. This leads to a tradeoff between accuracy and interpretability [4, p. 140]. Moreover, during the experimentation process of any machine learning project, various machine learning models are built, and their performance is evaluated. Using tools that are specifically built for achieving interpretability of certain models can hinder the flexibility of trying out different models for a good performance. To address the aforementioned concerns, it is important to think in terms of model agnostic interpretability for time series.

The first research question being addressed in this thesis: **How to achieve model agnostic interpretability in a time series forecasting problem?**

The ultimate goal of interpretability is to aid humans in understanding the decision-making process of machine learning models. As highlighted by Doshi-Velez and Kim [1, p. 1], different kinds of explanations produced after interpreting machine learning models may not be equally interpretable. Moreover, their research work claims that, with the growing research in the field of interpretability in machine learning, there is also a pressing need to be able to quantify the quality of explanations produced after interpreting the machine learning models. Due to these reasons, the evaluation of interpretability is extremely important and the research work in this field is gaining a lot of momentum. For example, Nguyen [8] evaluates different explanations produced by various interpretability techniques in a text classification problem and compares their results. Hase and Bansal [9] in their research work focus on the evaluation of explanations produced using various interpretability techniques on a sentiment classification problem. Yu et al. [10] focus on the evaluation of the explanations by interpreting the music generated by an artificial intelligence (AI) system. However, the evaluation of interpretability for machine learning models built with time series data is still unexplored, especially in the context of a time series forecasting problem.

The second research question being addressed in this thesis: **How to evaluate the interpretability of a time series forecasting model?**

1.3 Purpose

The purpose of this thesis is to build a framework/approach for achieving model agnostic interpretability in a multivariate time series forecasting problem. Another big purpose of this thesis is to evaluate the explanations produced by interpreting the machine learning model for time series forecasting.

The degree project was conducted at a company called Futurice in Helsinki, Finland. Futurice being a consultancy company is highly dependent on the sales of its consultancy services for running the business. However, the current system of understanding, if they are having enough sales activity in the company, is completely hunch based. Futurice has zero visibility into the aspects that drive its sales. In an attempt to become a more data-driven organization, Futurice wants to understand the effect of various sales-related activities conducted in the company on the sales deals closed by the company. Naturally, these various sales activities have sequentially occurred across time and have a time dimension attached to them. This makes the type of the data time series.

Using a data-driven strategy to understand the effect of the sales activities on the sales of the company can be used to inform the strategy of the sales teams at the company. A data-driven sales strategy can also help weed out human bias from the decision-making process.

1.4 Goals

The goal of this project is to achieve interpretability on a predictive model with multivariate time series data in order to understand the decision-making process of the underlying model and draw useful actionable insights. The goal of the degree project can be divided into four sub-goals.

1. Preparing the relevant time series data and formulating the forecasting problem.
2. Building time series forecasting models with different lag variables and selecting the best model based on an evaluation metric.
3. Interpreting the time series forecasting model to produce human understandable explanations.
4. Evaluating the explanations produced in the previous step (step 3).

1.5 Ethics, Sustainability and Societal Issues

Any work done in the field of interpretability has a huge ethical and societal impact attached to it. In fact, one of the main reasons why research in the field of interpretability is growing at a rapid pace is because of the realization that just building highly accurate machine learning models is not good enough. It is equally important to ensure that these models are reliable and fair. Machine learning algorithms are extremely powerful, but they are also very literal. For example, a machine learning model tasked to find the cheapest possible cure for cancer would probably find ways to end lives rather than saving them. Another example with respect to ethics as mentioned in [1, p. 3], talks about the need for an unbiased loan approval classifier. A loan approval classifier should not discriminate against people on the basis of race, ethnicity, or gender. Due to ethical and societal concerns, the

European Union passed the right to explanation act [11] which states that all algorithmic systems should be able to provide explanations.

Time series data is extremely common and is largely present in the field of economics, business, science, and technology. The work done during this thesis is shedding light on interpretability in time series and has the potential to have a huge impact in terms of addressing ethical and societal issues.

1.6 Research Methodology

The research methodology can be summarized into the following steps:

1. An empirical research will be conducted to address the research questions as it is hard to address the research questions from a theoretical point of view due to the high complexity of the machine learning and interpretability methods involved. Semi structured interviews will be conducted primarily with the sales employees to understand the sales process of the company and gather the required domain knowledge to build the relevant sales activity features. Semi structured interviews were chosen because they enable the interviewees to answer a particular set of questions but also allows them to share their opinions, thoughts, and experiences freely [12]. A raw dataset will be built on the basis of the interview results.
2. Post that, the time series forecasting problem will be formulated as a supervised learning problem which involves creating lag variables using the sliding window method. Approaching a time series forecasting problem as a supervised learning problem permits the use of various machine learning algorithms. Machine learning algorithms possess the ability to capture nonlinear relationships between the features and the target variable whereas the traditional statistical models for time series forecasting can only capture linear relationships [6, p. 14]. Moreover, the major focus of this project work lies in interpretability. Model agnostic interpretability techniques are also incapable of capturing the temporal dependencies of time series data while producing explanations [4, p. 191].
3. A classical machine learning algorithm, support vector regression (SVR) will be used for building the forecasting model. A classical machine learning model algorithm was preferred over fairly advanced deep learning algorithms due to the paucity of data. Deep learning algorithms are extremely powerful but are also extremely data intensive. It was seen from [13] that the performance of support vector regression was really good for time series forecasting.
4. Model agnostic interpretability techniques Shapley additive explanations (SHAP) [14] and Local interpretable model agnostic explanations (LIME) [2] will be used for interpretability because they are feature attribution methods. A feature attribution method presents the explanations of a complex machine learning model in terms of the contribution of features leading to the prediction. This makes the explanations intuitive for humans [4, p. 35].
5. The last task will involve the evaluation of the explanations produced after interpreting the SVR model using LIME and SHAP. The initial research by Doshi-Velez and Kim [1, p. 4] suggests three ways of evaluating interpretability. These are application grounded evaluation, human grounded evaluation, and functionally grounded evaluation. Furthermore, the research discusses that application grounded evaluation is expensive in terms of cost and time because it needs to be conducted with domain experts as compared to human grounded evaluation which can be conducted with lay humans. Designing an experiment for application grounded evaluation will be infeasible during this degree project due to constraints of time and resources and therefore, human grounded evaluation will be used.

1.7 Delimitations

The scope of interpretability is limited to local interpretability. Only two model agnostic interpretability techniques, SHAP and, LIME will be used during this project. The scope can be extended to other sophisticated model agnostic interpretability techniques.

Due to constraints of time and resources, the scope of evaluation will be limited to human evaluation of interpretability which involves conducting simple experiments with lay humans. The work can be extended to conduct application-grounded evaluation which involves domain experts performing reals tasks.

1.8 Structure of the thesis

This thesis is divided into five chapters. Chapter 2 presents relevant background information. It discusses concepts of time series, time series forecasting, and approaching time series as a supervised learning problem. Chapter 2 also discusses various concepts of interpretability including its definition, importance, types, and scope. Moreover, the background section extensively discusses model agnostic interpretability and two novel techniques of achieving model agnostic interpretability, LIME and SHAP. Chapter 2 concluded by shedding light on different approaches for the evaluation of interpretability.

Chapter 3 presents the methodology and methods used to solve the problems introduced in chapter 1. This chapter is divided into two main sections. The first section discusses the reasoning behind choosing the chosen methods and the second section focuses on application of the chosen method.

Chapter 4 discusses the results and analysis. Chapter 5 contains information about the conclusion and future work.

2 Background

The background section starts with an introduction to time series data and the concept of approaching time series forecasting as a supervised learning problem. Post that, the background section discusses Support vector regression (SVR) and various concepts of interpretability ranging from its importance to different types of interpretability and methods available for interpretability. The last section of the background section discusses the similar work that has focused on model agnostic interpretability of machine learning prediction models for time series data.

2.1 Time Series Data

Data points that are sequentially measured over a period of time form a time series data. The most important parameter that defines a time series data and separates it from other forms of data is the dependence on the dimension of time [5]. Time series that has data generated at regular intervals of time is called a regular time series and the time series with data generated at an irregular interval of time is called an irregular time series. The most common examples of time series data are stock prices recorded at the end of every day, sales of a company recorded at the end of every month, revenue generated by a company every month, hourly average temperature readings of a particular location, reading generated by a heart rate monitor every second, reading recorded by an IoT sensor during an experiment, etc. In fact, time series forms one of the most important data types in the fields of business, economics, science, and engineering as most of the measurements are performed over time.

A time series that consists of a single value sequentially measured over a period of time is called a univariate time series whereas a time series that consists of two of more values sequentially measured over a period of time is called a multivariate time series. For example, the average perspiration of a country recoded every year forms a univariate time series. Average perspiration recorded along with average pollution, average wind speed, and average temperature on a yearly basis (same time interval) forms a multivariate time series. Figure 2-1 shows an example of a univariate time series adapted from [6, p. 64]. Figure 2-2 shows an example of a multivariate time series adapted from [15].

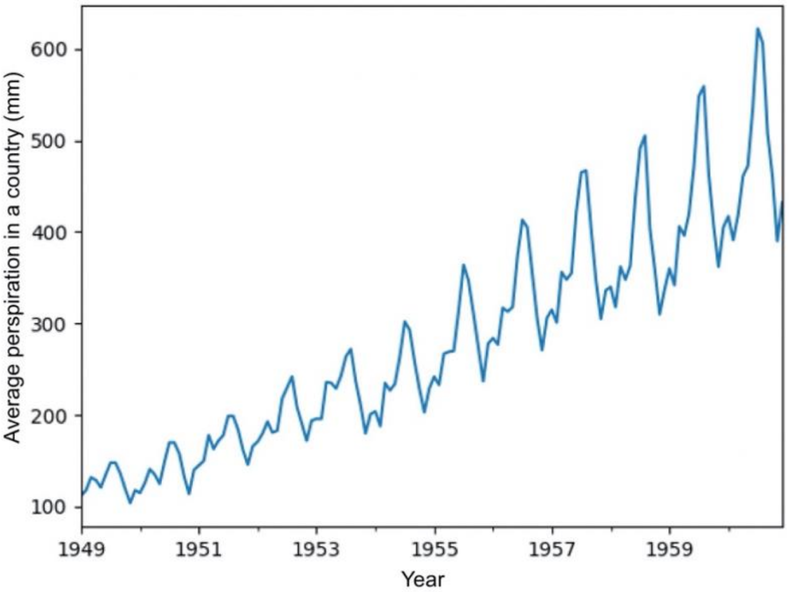


Figure 2-1: Average perspiration in a country recorded against year is an example of a univariate time series [6, p. 64]

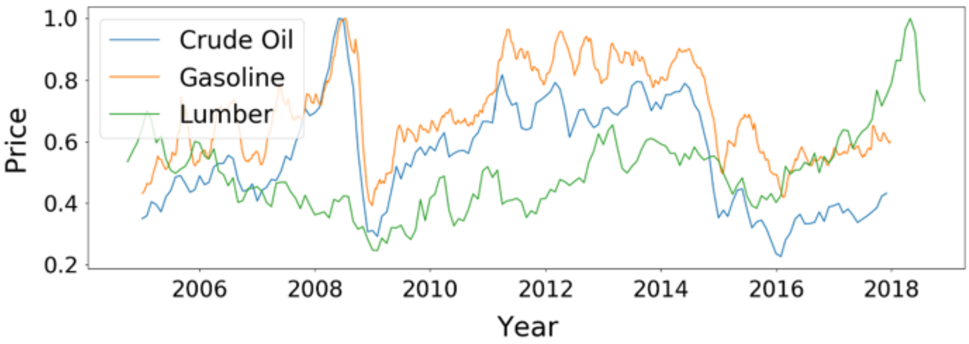


Figure 2-2: Price of Crude Oil, Gasoline and Lumber simultaneously recorded against a year forms an example of multivariate time series [15]

2.1.1 Time series forecasting

Time series forecasting is the process of understanding the past and using it to predict the future. During the process of time series forecasting, the historical observations of the time series are used to build models capable of predicting the future values of the series [6, p. 9]. To further deepen the knowledge of time series forecasting, it is important to get familiar with the common time series nomenclature. The current state of time is defined with subscript t and the observation at the current state of time is described at obs (t). Times before the current time are considered negative relative to the current time are called lag times and the observations at those times are called lagged observations. For example, the time prior to current time ***t*** is referred to as ***t-1,*** the time prior to that is ***t-2*** and the of time references in the same direction follow the order. A time instance with a lag of 20 would be referred to as ***t-20***. The observations at ***t-1*** and ***t-2*** would be ***obs (t-1)*** and ***obs (t-2)*** respectively. Similarly, the time in the future is considered positive with respect to the current time ***t***, called lead time, and is represented by ***t+1***, ***t+2,*** and so on. Following the same chronology, the observations at time instance ***t+1*** and ***t+2*** are represented by ***obs(t+1)*** and ***obs(t+2)*** [6, p. 10].

To summarize:

- The current time which is used as the time of reference is represented by ***t***.
- The previous or lagging time steps are represented by ***t-n***, where n is a particular time instance (e.g. t-1, t-2, t-3, and henceforth)
- The future or leading time steps are represented by ***t+n,*** where n is a particular time step (e.g. t+1, t+2, t+3, and henceforth)

2.1.2 Time series forecasting as a supervised learning problem

Supervised learning is the type of machine learning problem where an algorithm tries to learn a mathematical function that maps the input variable X to an output variable Y as accurately as possible. The input variable can also be called the input feature and the output variable can also be called the target variable. This learned mapping function is then used to predict the output for new input variables.

Time series forecasting can be framed as a supervised learning problem and this enables the use of various linear and nonlinear machine learning algorithms to predict values at future time steps. This is achieved using the values from the previous time steps in a time series to predict the values in the future time steps [6, p. 14].

In one step forecasting, the values from the previous time steps are used to predict the value at the next time step. In multi-step forecasting the values from the previous time steps are used to predict values at two or more future time steps.

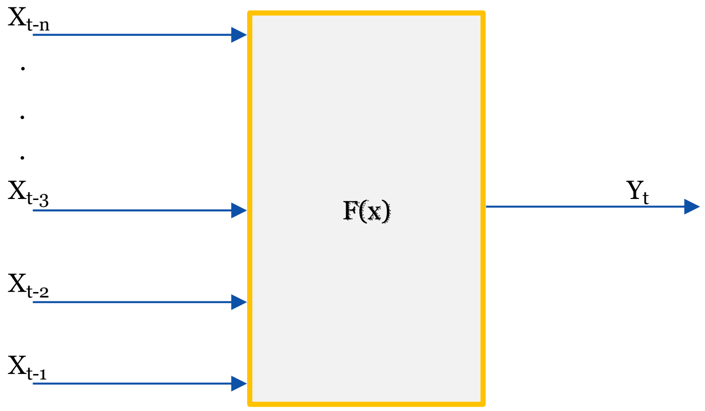


Figure 2-3: Approaching time series forecasting as a supervised learning problem

Figure 2-3 depicts the procedure of approaching a unit step time series forecasting as a supervised machine learning problem. The set of input features formed using the lag values of a univariate time series are denoted by X_{t-1}, X_{t-2}, X_{t-3}...... X_{t-n} where n is denoted by the final lag value or the size of the lag and is less than the total of data points (rows) in the dataset. The machine learning model is denoted by F(x) and the output, Y_t is the predicted value at a future time step. Similarly, the concept can be extended to multivariate time series and lag values of different variables can be used to predict the values at future time steps.

2.2 Support Vector Regression

Support Vector Machine (SVM) is a supervised learning method that is commonly used for machine learning problems of classification type. SVM projects the training data into a higher feature dimensional space using something called as a kernel function and finds an optimal hyperplane to create a decision boundary with the maximum possible margin. Projecting the training data into a higher-dimensional space makes it easier to find an optimal hyperplane for classification. Different kernel functions available in SVM include Linear, Polynomial, and Radial Basis Function (RBF) kernel. To find the optimal decision boundary SVM uses support vectors that are data points that lie on the maximized margin [16, p. 1].

Figure 2-4 adapted from [17, p. 155] presents an intuition behind a binary classification case using SVM. The solid black line in the center is the best hyperplane that separates the classes C1 and C2. The best hyperplane is found by optimizing for the maximum margin distance (best margin) with the help of the support vectors that are represented by dotted lines in the figure.

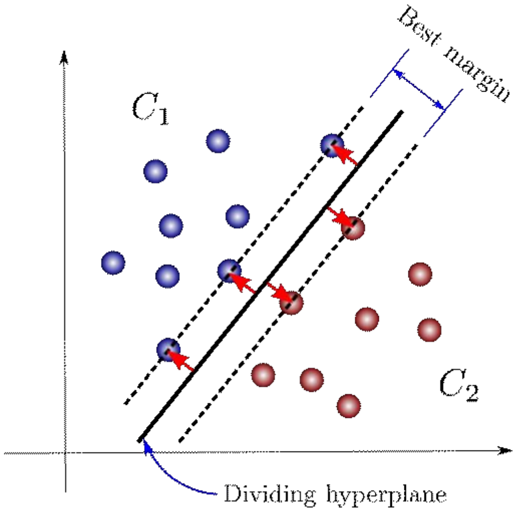


Figure 2-4: Intuition behind SVM in a binary classification case [17, p. 155]

SVM was initially developed for classification problems but was later extended to suit regression problems. SVR operates with the same logic as SVM but rather than finding a decision boundary with maximum possible margin SVR focuses on finding an approximate function to minimize the error of the loss function. SVR tries to find a decision boundary depending on the defined loss function that ignores errors that are located within a particular distance of the true value. The distance is denoted with a variable ϵ. Hence, SVR does not care about the whole training data and depends on a subset of training data to make predictions. The ϵ variable dictates the tolerance for error in SVR. The concept of kernels also holds true for SVR and nonlinear data is mapped into a data of higher dimensional space to make it linear [17, pp. 155-156].

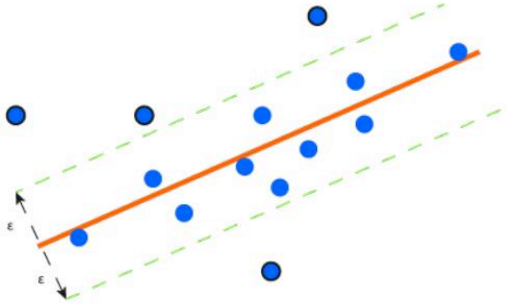


Figure 2-5: Intuition behind a two-dimensional SVR [17]

In Figure 2-5 [17, p. 157], the orange line represents the function built by SVR and the green dotted lines form the ϵ tube (margin). The points outside the ϵ tube, highlighted with black circles are the support vectors in SVR. The best fit hyperplane will try to have the maximum number of data points from the training data inside the ϵ tube.

The various parameters that can be tuned in SVR are as follows [16, pp. 19-23]:

- Kernel - Linear, Polynomial or Radial Basis Function Kernel

- C – The penalty factor to the error of the loss function (regularization parameter). A higher C penalizes the error more and could lead to low generalization (high overfitting) of the function built by SVR. A lower C penalizes the error loss.
- Size of the ϵ tube (margin)
- Gamma, if the Kernel is Radial basis function (RBF)

2.3 Concepts of interpretability

This section discusses various concepts of interpretability like its definition, importance, types, and scope. The section also discusses two model agnostic local interpretability techniques called LIME and SHAP. The section concludes by shedding light on various techniques for the evaluation of interpretability.

2.3.1 Definition and importance of interpretability

Miller [3] in his research work defines interpretability as “the degree to which a human can understand the cause of a decision”. Another definition of interpretability offered by Kim et al. [18] in their research work is that “Interpretability is the degree to which a human can consistently predict a model’s output”.

Since machine learning and data science are seeping into every domain of life, a lot of research in the field of interpretability in machine learning has been happening to enable humans to thoroughly understand the reasons behind the classification or regression outputs of these machine learning models.

It has been recognized that prediction accuracy is an incomplete metric to judge the performance of a machine learning model and a deeper human friendly understanding about the elements of the data that lead to prediction is important. Interpretability helps humans to better understand the following elements of the decision making of a machine learning model [1, pp. 3-4].

- It helps humans understand about the fairness of a machine learning model. For example, the interpretability of a model that approves or rejects a loan application at a bank would help the humans understand if the model is or is not discriminating based on gender or race.
- Interpretability ensured that the privacy is not violated in any way and the sensitive information in the data is protected.
- Interpretability helps in understanding the reliability and robustness of a machine learning algorithm. In a robust machine learning model, the prediction does not drastically change if a small amount of noise is added to the input data.
- One of the most important implications of interpretability is that it generates trust by enabling humans to understand the reasons behind decisions made by a machine learning model.

2.3.2 Types of interpretability methods and tools

The types of interpretability methods can be divided into various categories on the basis of how the interpretability of a machine learning model is obtained. The interpretability can be obtained by using machine learning models that are less complex making them easily interpretable. This kind of interpretability is referred to as intrinsic interpretability and examples of intrinsically interpretable models include linear regression, logistic regression, and short decision trees. The second way to obtain interpretability is by using interpretation methods after the model has been trained. This kind of interpretability is called model agnostic or post hoc interpretability [4, pp. 25-26].

It is also important to understand the difference between the kinds of interpretability tools based on interpretability methods. Model Specific interpretability tools only work with specific models. Since the methods used to interpret intrinsically interpretable models are limited to these models, they by default fall under the category of model specific interpretability tools. Any interpretability tool that is developed to target a particular model class falls under the category of model specific interpretability tool. For example, an interpretability tool that can just interpret Recurrent Neural Networks would be a model specific interpretability tool. Model Agnostic tools are used to interpret models post training (post hoc) and can be used to interpret any machine learning model. Model agnostic tools do not acquire any internal information about the inner working of the model like its weights or any other structural information for interpretability [4, p. 26].

2.3.3 Global vs local Interpretability

The scope of interpretation from an interpretability method could be Global or local. Global interpretability is achieved when the interpretability method holistically describes the effect of features on the prediction of the target variable [4, pp. 27-28].

Local interpretability is achieved when the interpretability method can describe the effect of feature values from a particular instance of the dataset on the prediction made for that particular instance. A good human understandable output produced after interpreting the model is called an explanation [4, p. 29].

2.3.4 Model agnostic interpretability

Using model agnostic tools for interpreting machine learning models has a very big advantage. Since the interpretability, while using model agnostic tools is not dependent on the type of machine learning model as in the case of model specific interpretability tools, it offers a great amount of flexibility to choose and build any kind of machine learning model. During any machine learning model building process, the target is to get the best model in terms of the model performance evaluation metric. Since intrinsically interpretable models are not capable of picking up complex relationships between the input features and the target variable, they pose a big disadvantage in terms of model performance. Model agnostic interpretability tools help overcome the tradeoff between model performance and interpretability. This leads to the freedom of not having to just use intrinsically interpretable models for interpretability. Another great advantage of using model agnostic interpretability methods is that interpretability of different machine learning models used for the same task can be compared [19].

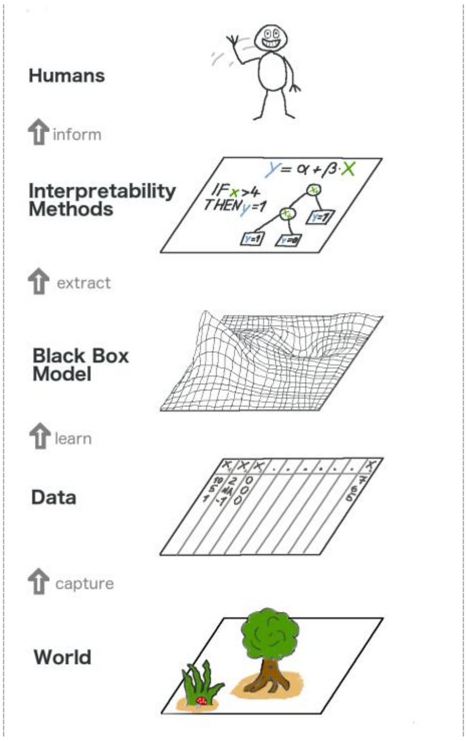


Figure 2-6: The overall picture of model agnostic interpretability [4, p. 142]

The overall picture of model agnostic interpretability is presented in Figure 2-6 was adapted from [4, p. 142].

Molnar [4, pp. 144-200] in his book discusses various model agnostic tools which include partial dependence plot, Individual Condition expectation, accumulated local effects, feature interaction, permutation feature importance, and surrogate methods like LIME and SHAP. A surrogate method builds a surrogate model that is trained on the black-box model to approximate its predictions. The surrogate model is an intrinsically interpretable model like linear regression or decision trees and the interpretability of the black-box model is achieved by interpreting the surrogate model. The goal of a good surrogate model is to approximate the black box prediction model as accurately as possible. Since the surrogate model only requires the prediction function of the black-box model and the data and does not have any information regarding the working of the black-box model, it is model agnostic.

Molnar in [4, p. 201] further summarizes the steps used to build a global surrogate model and approach interpretability using surrogate models.

- A dataset is selected. This dataset could be the same as the one used to train the black-box model or a subset of the original dataset like the validation or test set.
- The predictions for the selected dataset are obtained using the black-box model.
- An interictally interpretable model like linear regression or decision tree is trained on the selected dataset and the predictions from the black-box model. This results in the surrogate model.
- The surrogate model is then interpreted to approximately explain the effect of various input features on the target prediction of the black-box model.

Since the surrogate model is trying to approximate the black-box model and not make predictions using the original data, the prediction quality of the black box model influences the quality of interpretations obtained from the surrogate model. If the black-box model is performing poorly then the conclusions drawn about the black box model in form of interpretations using the surrogate model would also be poor [4, p. 202].

2.3.5 Local interpretable model agnostic explanations (LIME)

Ribeiro et al. [2, p. 1135] in their research work proposed a local surrogate model called LIME to explain the prediction on a single instance of data by a black box machine learning model. In comparison to global surrogate models which focus on explaining the holistic effect of features on the predictions made by the black-box model, a local surrogate model like LIME focuses on explaining individual predictions.

LIME follows the following general approach to explain individual predictions [4, p. 207].

- After the instance whose prediction needs to be explained is provided to LIME, it permutes (perturbs) the dataset to create new sample data.
- Corresponding predictions for the new sample data are obtained using the black-box model.
- This new sample data is weighted with respect to its proximity to the instance whose prediction needs to be explained.
- A weighted interpretable model like linear regression is trained on the sample data and the corresponding predictions.
- The trained interpretable model is interpreted to explain the individual prediction.

The local surrogate model should be good at approximating the black-box model locally but does not necessarily have to be good at approximating the model behavior globally.

Figure 2-7 from the paper on LIME [2, p. 1138] presents a toy example to highlight the intuition behind LIME.

The blue/pink color represents the complex decision boundary of the black box prediction function. The dark red cross is the instance that needs to be explained. LIME creates new sample data and obtains its predictions using the black box model which are then weighted with respect to their proximity to the instance that needs to be explained. These weights are represented by the size in the image. The dashed line represents the obtained surrogate model like linear regression that can be interpreted to explain the prediction [2, pp. 1137-1138].

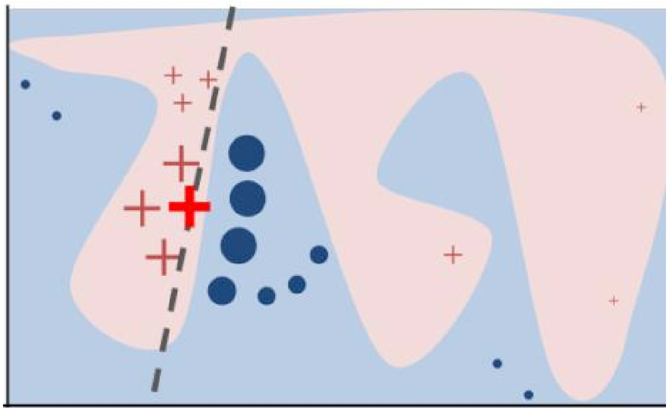


Figure 2-7: The intuition behind LIME highlighted for a black box model for binary classification [2]

Mathematically LIME's explanation of a local an individual prediction can be expressed in the form of Equation **2-1** [4, p. 206]:

$$\text{explanation}(x) = \arg\min_{g \in G} \quad L(f, g, \pi_x) + \Omega(g)$$

Equation 2-1 [4, p. 206]

The explanation for an individual prediction instance x is obtained by a local surrogate model like linear regression represented by g. The explanation function minimizes the loss denoted by L which is a measure of how close the explanation obtained by the surrogate model is to the prediction of the original black box model f while keeping the complexity of the surrogate model $\Omega(g)$ low. Model complexity can be controlled by the user by limiting the number of features that the surrogate model can use. All possible explanations generated in form of surrogate models are represented by G. π_x is the proximity measure to define the size of the neighborhood around x. The technical details behind how the size of the neighborhood is decided can be found in [4, pp. 209-210].

2.3.6 Shapley Additive explanations (SHAP)

SHAP (SHapley Additive exPlanations) [14] is a surrogate model approach to interpret black box models. SHAP is a method based on a concept from cooperative game theory called Shapley Values [20, p. 307]. SHAP offers local interpretability with the shapley value-based method to explain the cause of individual predictions and also offers global interpretability based on the addition of shapley values from individual predictions. In order to understand SHAP, it is critical to first understand the concept of Shapley values.

Shapley values is a technique for fairly distributing the payout among different players involved in a co-operative game depending on their contribution to the total payout [20, pp. 307-308].

In terms of relation to machine learning, the game is the task of predicting the value of the target variable using a single instance of feature values from the dataset. The players are the feature values of that instance from the dataset that collaborate in coalition to predict the value of the target variable, the prediction is the payout, and the gain is the predicted value subtracted by the average prediction of the model over all the instances of the data from the dataset [4, p. 219].

Consider the following example to understand the shapley value intuitively. If there is a machine learning model that is predicting the salary of an employee at a company based on four features namely age, gender, years of experience, and distance from the workplace. One particular prediction instance in the dataset where the age is 30 years, gender is male, years of experience is 8 years, distance from the workplace is 5 km, and predicted salary on the basis of these features is 60,000 is taken into consideration for interpretation. To calculate the shapley value for age – 30 years in this particular case, the salary would be predicted for each of the feature coalition given below, with and without the feature age – 30 years. The difference between the predicted salary with and without the feature age - 30 years for each of the coalition would be the marginal contribution. The shapley value would be the average of all marginal contributions. All possible coalitions are as follows.

- No feature values
- Gender - Male
- Years of experience - 8 years
- Distance from the workplace - 5 km
- Gender - Male + Years of experience - 8 years
- Gender - Male + Distance from the workplace - 5 km
- Years of experience - 8 years + Distance from the workplace - 5 km

- Gender - Male + Years of experience - 8 years + Distance from the workplace - 5 km

The feature values of the features not in the coalition can be replaced with 0 to make the prediction. Putting the value to 0 means that the feature is not present.

Mathematically, the shapley values estimation formula as defined by Lundberg and Lee in [14] and reiterated by Molnar in his book [4, p. 226] is as follows (Equation 2-2 [4, p. 226] [4, p. 226]).

$$\phi_j(val) = \sum_{S \subseteq \{x_1,\ldots,x_p\} \setminus \{x_j\}} \frac{|S|!\,(p-|S|-1)!}{p!}\left(val\left(S \cup \{x_j\}\right) - val(S)\right)$$

Equation 2-2 [4, p. 226]

ϕ_j is the shapley value for a particular feature value j from one prediction instances. S is the subset of features used by the model, x the feature value vector of the prediction instance that needs to be explained and p is the number of features in the prediction instance. $val_x(S)$ signifies the prediction of feature values in S marginalized over features not included in S. This can be seen from Equation **2-3** [4, p. 226].

$$val_x(S) = \int \hat{f}(x_1, \ldots, x_p)\,dP_{x \notin S} - E_X\left(\hat{f}(X)\right)$$

Equation 2-3 [4, p. 226]

The concept of SHAP is built on shapley values and SHAP explains the prediction made in the particular instance but measuring the contribution of each feature variable on the prediction. SHAP computes the shapley values and post that the one addition it brings to the table is by representing the shapley value explanation as an additive feature attribution model, a linear model. This approach of SHAP is called Kernal SHAP and it connects LIME and shapley values. The explanation produced by Kernal SHAP can be mathematically represented by Equation **2-4** [14, p. 2].

$$g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j z_j'$$

Equation 2-4 [14, p. 2]

The model made by SHAP that produces the explanation is g, the coalition vector is $z' \in \{0,1\}^M$. M is the maximum coalition size and ϕ_j is the feature attribution for a particular feature j, its shapley value. The entry of 1 in the coalition vector means that a particular feature is present and an entry of 0 signifies that the particular feature is absent.

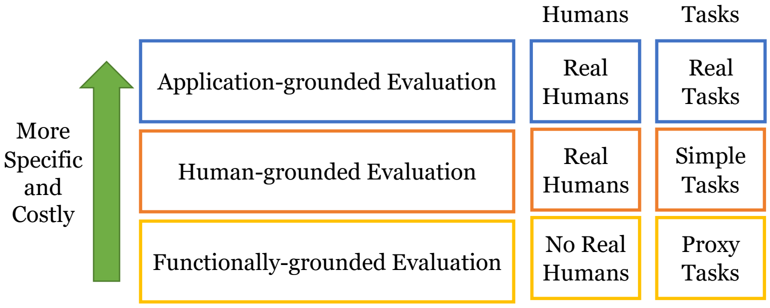
2.3.7 Evaluation of interpretability

Until now, there is no clear consensus on how to evaluate interpretability [4, p. 30]. However, initial research by Doshi-Velez and Kim [1, pp. 4-6] suggests the following ways to evaluate interpretability.

1. Application-grounded evaluation: In application level (grounded) evaluation, the explanations generated after interpretability are tested by the domain experts/end user. The quality of the evaluations is evaluated with respect to the end task. For example, if the explanations for a machine learning model tasked with performing diagnosis of a particular disease are generated, they would be evaluated by doctors performing the diagnosis. If the domain expert is able to understand and explain the decision made by the model, it can be considered a good interpretation. This evaluation technique is considered the best because of two reasons. Firstly, it directly evaluates the explanations with respect to the final objective/task of the ML system. Secondly, it involves domain experts. However, conducting

application-level evaluation is very costly in terms of the time and effort involved. This makes the application-level evaluation very difficult.

2. Human-grounded evaluation: In Human level (grounded) evaluation, the explanations are presented to ordinary people for evaluation. During this evaluation, the ordinary people are subjected to simple experimental tasks. Human-level evaluation is an attractive choice because it reduces the cost of experimentation and is also simpler to conduct as compared to application-level evaluation. The human-level evaluation also drastically increases the size of the subjects as it is conducted with ordinary humans. Human-level evaluation is a metric that depends on the quality of the generated explanation and is independent of the prediction accuracy of the underlying machine learning model. Some examples of possible experiments include:
   - Forward simulation experiment where ordinary humans are presented with an input to the ML model and its explanation. Using the input and explanation, humans are asked to correctly simulate the output of the ML model.
   - Counterfactual simulation experiment where the ordinary humans are presented with an input to the ML model along with its output and explanation. The humans are asked about possible changes that can be made to the model's input to change its prediction to a desired output.
3. Functionally-grounded evaluation: Function level (grounded) evaluation does not require humans and is well suited for models that have been already evaluated by humans. They are cheaper than human-level evaluation in terms of time, effort, and cost. Function level evaluation involves a proxy task based on some formal definition of interpretability to evaluate the quality of explanations. Function level evaluation does seem very attractive because it is inexpensive, but it is extremely challenging to determine the proxy tasks for evaluation.



	Humans	Tasks
Application-grounded Evaluation	Real Humans	Real Tasks
Human-grounded Evaluation	Real Humans	Simple Tasks
Functionally-grounded Evaluation	No Real Humans	Proxy Tasks

Figure 2-8: Different techniques of evaluating interpretability [1, p. 2]

Figure 2-8 from [1, p. 2] summarizes the various evaluation methods of interpretability along with their attributes.

2.4 Related Work

This section discusses the work done in two research papers related to this project work.

2.4.1 Model agnostic interpretability in financial time series with SHAP

Mokhtari et al. [21, pp. 168-170] have approached the interpretability of financial time series using SHAP. The problem tackled in this paper was a classification problem and both binary class and multi-class classification models were built using SVM, XGBoost, Random Forest, k Nearest Neighbors (kNN), and LSTM. The predictions made were interpreted using SHAP to understand the most important features for the prediction and to understand the contribution of the new dataset in comparison to the old dataset for the prediction task. kNN and SVM had the best performance on their dataset and SHAP was useful in drawing crucial insights about the prediction.

The research work done in [21, p. 171] only focuses on the global interpretability aspect of interpretability. Moreover, this research work only experiments with SHAP as an interpretability technique. The biggest limitation of the research work is that no methods were applied to evaluate the quality of the explanations produced after interpretability.

2.4.2 Interpretability in a time series forecasting problem using SHAP

García et al. [22, p. 7] in their research used multivariate time series data about various atmosphere related factors like wind speed, solar radiation, temperature, humidity, etc. to predict the NO_2 concentrations in the city of Madrid using LSTM. The predictions were then interpreted using SHAP to understand the overall effect of features on particular prediction instances. SHAP was also used to obtain the feature importance to understand the overall impact of each feature on the prediction of NO_2 concentrations.

The research work done in [22, p. 6] produces both local and global explanations for its time series forecasting model using SHAP. However, a big limitation of the work is that the quality of the explanations produced (local and global) was not evaluated.

3 Methodology

This chapter is divided into two major sections. The first section is the choice of method section which discusses the decision-making process behind the chosen methodology. The second section is the application of method section which discusses how different methods were applied to solve the research problems.

3.1 Choice of Method

The reasoning behind the chosen methodology is presented as follows:

1. An empirical research was conducted during the project to address the research questions as it would have been extremely hard to address the research questions from a theoretical point of view due to the high complexity of machine learning and interpretability methods involved. The first step involved understanding the sales processes of the company and the data streams that encapsulate the sales activity of the company. A qualitative analysis was conducted in form of interviews to achieve the aforementioned objectives. Interview is a common data collection method used in research that helps the researcher get a thorough understanding of the problem and interviewee's point of view [12]. The methodology of conducting interviews best aligned with the objective of obtaining the required domain knowledge to prepare the list of sales activity related data streams. The next step involved procuring the data from the data streams and cleaning them to create usable features for machine learning. After the features were created, an exploratory data analysis was conducted to understand the features, which is a commonly used step for any machine learning project.

2. After understanding the features, it was decided to approach the time series forecasting part of the project as a supervised learning problem using machine learning. This decision was based on two reasons. The first reason, as mentioned by Bwonlee in [6, p. 14] is that using machine learning algorithms for time series forecasting gives a very big advantage over conventional time series forecasting algorithms. The advantage is that machine learning algorithms can capture nonlinear relationships between the feature variables and the target variable whereas traditional time series forecasting techniques are capable of only capturing linear relationships between the features and the target variable. Machine learning algorithms can also be used with both univariate and multivariate time series data. Brownlee also highlighted in [6, p. 15] that classical machine learning algorithms are incapable of automatically detecting the temporal dependencies in data while making predictions. That is why it becomes important to lag the time series manually (approaching time series forecasting as supervised learning) during forecasting or classification tasks with machine learning. An exception to this is an advanced deep learning algorithm like Long short-term memory (LSTM).

   The second reason is that making a prediction on time series data is just one part of this project. The main focus lies in interpreting the model built on time series data to draw valuable insights. As found during the literature survey in [4, p. 191], various model agnostic interpretability techniques are also incapable of automatically understanding the temporal dependencies of the time series data. As shown by Mokhtari et al. [21, pp. 168-170], even if an advanced deep learning algorithm like LSTM capable of detecting temporal dependencies is used, model agnostic interpretability methods would be unable to understand these temporal dependencies while generating explanations. That is why Mokhtari et al. [21, p. 170] created lag variables for their time series manually before feeding it to LSTM.

3. After preparing the dataset for forecasting it was fed into the support vector regression algorithm. The choice of the machine learning model to be used was based on the amount of data available. The size of the dataset was not huge enough, so only classical ML model algorithms were considered to make predictions during this project. As noted by Chniti et al. in [13, p. 80], a deep learning algorithm like LSTM due to its state-of-the-art architecture is considered perfect for dealing with sequential data like time series. Since LSTM requires large amounts of data, they were not used during this study. Moreover, the comparative study by Chniti et al. [13, pp. 82-83] compares the performance of LSTM and a powerful classical ML algorithm, SVR for time series forecasting. The study found that the performance of the SVR model was comparable and even better than LSTM in some cases. SVR was the algorithm of choice for time series forecasting during this study project.

4. Thereafter, the predictions made by the SVR model were interpreted using LIME and SHAP interpretability methods to understand the influence of features on the predictions at a local level. LIME and SHAP are two of the most novel and state of the art model agnostic interpretability techniques [23]. Molnar in [4, p. 215] discusses that the biggest advantage of model agnostic techniques like LIME and SHAP which makes them very promising is that they are feature attribution methods. A feature attribution method presents the explanation of a complex (black box) machine learning model in terms of the contribution of features leading to prediction. Molnar further discusses that the explanations of feature attribution methods are extremely human friendly and even make sense to lay humans. Molnar also highlights that solid theoretical foundations in game theory (shapley values) is one of the biggest advantages of SHAP. This really ensures that prediction is distributed fairly among features.

5. The last task was the evaluation of the explanations produced after interpreting the SVR model using SHAP and LIME. As found during the literature survey, the scientific community has not yet agreed on a standard way to evaluate the interpretability [4, p. 30]. However, the initial research by Doshi-Velez and Kim [1, p. 4] suggests three ways of evaluating interpretability. These are application grounded evaluation, human grounded evaluation, and functionally grounded evaluation. As further discussed by Doshi-Velez and Kim, application grounded evaluation is very costly in terms of time and effort. This is because application grounded evaluation is conducted with domain experts and with respect to the real task. Designing an experiment for application grounded evaluation was infeasible during this degree project due to constraints of time and resources. It was further discussed by Doshi-Velez and Kim that although functionally grounded is cheaper in time and effort, it is suited for models that have already been evaluated by humans.

   Due to the aforementioned reason, human grounded evaluation of interpretability seemed the most appropriate method and was chosen. Human grounded evaluation of interpretability involves ordinary humans and a simple task [1, p. 5]. The human evaluation of interpretability for time series, to the best of the author's knowledge, is still unexplored. Due to this reason, there was no direct point of reference for designing the human evaluation experiment for interpretability. However, other research work related to the human evaluation of interpretability was thoroughly explored to get a good understanding about possible experiments for human evaluation. Nguyen [8] conducted a forward prediction (simulation) task for human evaluation. Nguyen during her research worked with a binary classification case where the model predicts the sentiment of the movie review, positive or negative based on the text of the review. During the forward prediction task, Nguyen presented ordinary humans with the input to the model that is the text of the review and the explanation of the model. Based on the input and explanation, the humans were asked to predict the output of the model, negative or positive sentiment. Hase and Bansal [9] also worked on the movie review text dataset with output labels as positive or negative. Hase and

Bansal tried to improve the design of the forward prediction(simulation). The idea of using the forward prediction task for this project work was rejected because it seems reasonable to use it in a binary classification setting but using it in a regression(forecasting) setting seemed unreasonable. It would be very unreasonable to ask ordinary humans to predict a numerical output on the basis of input feature and explanation as there are infinite numbers in the number system.

Malhi et al. [24, pp. 129-130] conducted human evaluation of interpretability for an ML model built for approving or rejecting a bank loan application using various attributes of the loan applicants. The researchers here employed a verification task for the human evaluation of interpretability where they presented the humans with an input, an output (approve or reject), and an explanation. Based on the input, output, and explanation, the humans were asked to agree or disagree with the output of the model.

It was realized that a variation of verification task could be used for human evaluation of interpretability in the case of a time series forecasting problem. The human evaluation study in [24, pp. 134-140] formed the basis for the human evaluation study conducted during this thesis.

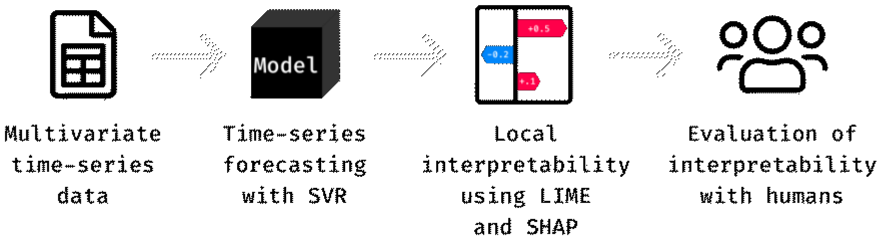

Model
+0.5
-0.2
+.1
Multivariate time-series data
Time-series forecasting with SVR
Local interpretability using LIME and SHAP
Evaluation of interpretability with humans

Figure 3-1: Overview of the methodology

Figure 3-1 presents an overview of the methodology used to address the research problems.

3.2 Application of Method

This section discusses the application part of the chosen methodology. The topics span across interviews to build the dataset for ML, dataset preparation, time series forecasting using SVR, generating local explanations using LIME and, SHAP and human evaluation of interpretability.

3.2.1 Interviews to build the dataset for machine learning

The first step undertaken was to understand the sales process of the company by conducting interviews. The motivation behind conducting the interviews was as follows:

- Understanding the sales process of the company.
- Identifying the data streams in the company that contain data of various sales related activities.
- Obtain the domain knowledge required to build useful features from the identified data streams.

The company encourages employees to log data about various sales related activities in the company but due to self-organized culture of the company, the members of the sales teams do not necessarily log all the data about the sales process. Another important objective of these meetings was also to understand the sales activity related data that mandatorily gets logged by the sales teams throughout the company.

3.2.1.1 Choice of participants

The interviews were conducted with sales professionals, sales team leads, and the executives that are involved in the sales processes. The interview process consisted of participants with at least one year of sales experience in the company. It was done to make sure they thoroughly understood the sales process of the company.

3.2.1.2 Structure of the interview

The interviews were conducted in a semi structured format as it enables the interviewee to answer a particular set of questions but also enables them to share their opinions, thoughts and experiences freely [12]. The interviews began with a fixed set of questions regarding the sales process, the type of tools used during the process, the kind of data is logged in the company's system during the sales process, and the frequency of data logging. Due to the semi structured format of the interview, all interviewers shared their thoughts on the sales tools that are mandatorily used by most sales professionals across all the offices of the company and lead to reliable data streams that can be used for the project.

A total of 8 interviews were conducted and the specifics of the interview are below. The names of the people have been anonymized due to privacy reasons. Table 3.1 presents the statistics of the interview process conducted.

Table 3.1: Statistics of the interviews to understand the sales process

Role at the Company	**Time of employment at the company**	**Interview time**
Sales professional 1 in Helsinki office	2 years	30 mins
Sales professional 2 in Helsinki office	1 year	30 mins
Senior sales manager in Helsinki office	3 years	45 mins
Sales team lead in Helsinki office	5 years	30 mins
Sales professional 3 in Berlin office	3 years	45 mins
Senior strategy executive in Helsinki office	6 years	1 hour
Senior executive in Helsinki office	19 years	1 hour

3.2.2 Dataset Preparation

This section discusses the 3 steps followed for building the relevant dataset for the time series forecasting model.

3.2.2.1 Building the dataset

The understanding gained from the interviews in section 3.2.1 was the basis for building the relevant dataset for the ML forecasting model. The exact relevance of the feature set built, and the data streams used with respect to the sales process in the company have been discussed in the results section This section focuses on the method used for building the relevant sales activity dataset. Figure 3-2 illustrates the process holistically.

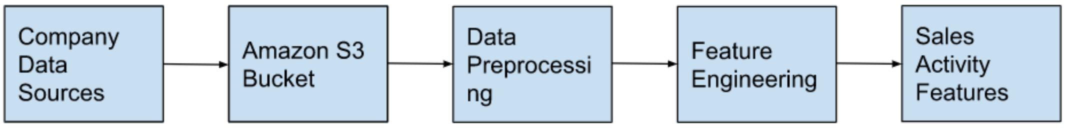


Figure 3-2: Holistic view into the process of building the dataset

The process was completed using the following steps:

1. Collecting the data from various data streams – Most of the data from different company-wide data streams is parsed and stored into the company's data lake hosted on Amazon Web Services (AWS)'s Simple Storage Space (S3) bucket.

    - The first step involved procuring the names of the people who are involved in the sales at the company. The company has a centralized human resources database that contains the names of the people who are currently employed by the company alongside their other details like email, phone number, pay scale, time of employment, and competence/role of the employee in the company. This data was procured from the company's data lake in Comma Separated Values (CSV) format and then the names of the employees with sales as a competence in the competence field of the dataset were extracted. However, the problem was that this dataset only contained the names of the active employees of the sales teams and did not contain any information about the employees who have left the company. In order to make up for that and to complete the list of people with sales as a competence in the company, another database with information about the past employees of the company was manually procured from the human resources department. This database was made available in an excel spreadsheet format. It was then used to extract the names of the company's past employees with sales competence. The final list of the employees with sales competence was then built. It contained the names of both the current and previous employees of the company.
    - The data about the top fifty clients of the company in the last ten years was procured. The basis for selecting the top fifty clients was the total billing amount generated by the clients. The billing records of the clients are stored in the financial database of the company. The data in the financial database of the company is manually updated on a monthly basis and pushed to the data lake of the company. The client billing data procured from the data lake in the CSV format contained the following data fields: name of the client and billing amount.

- The company uses Google calendar for booking internal and external meetings. The data about meetings from Google's calendar API is parsed and stored in the company's data lake on a daily basis. The google calendar API offers a very exhaustive list of fields generated while creating a calendar invite. The calendar data was obtained from the data lake in CSV format. The data in the following fields was extracted from the calendar data: name of the organizer, the start date of the meeting, the title of the meeting, and the description text of the meeting.
- As a standard practice in any company, employees are required to mark the hours they spend working. The company's data lake pulls this data from the hour making software at the company on a daily basis. The dataset was procured and contained the following fields: name of the employee, task completed, and time taken to complete the task.
- The company uses a Customer Relationship Management (CRM) software called Hubspot to record the journey of sales deals. The data of the CRM software was procured from the data lake in CSV format. The data of the CRM software contained multiple data fields out of which two fields called time of the deal and the status of the deal was extracted.
- The company's sales employees have credit cards which are used for various company-related expenses. The expenses of the credit card are manually logged and stored by the employees in a centralized excel file at the end of every month. The excel files are processed and made available on the company's data lake. The data was obtained from the data lake in the CSV format and contained the following fields: name of the employee, transaction amount, and description of the transaction.

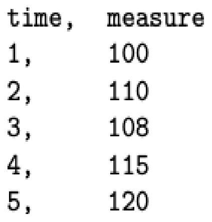
2. Data preprocessing – The data was made error-free for the machine learning algorithm by removing rows and columns with empty or corrupted values. The formatting of the data time data was fixed, and it was standardized to eastern european time (EET). Date time columns were moved to the index location in time series for easy manipulation of data. The irregular time series were converted into regular time series by aggregating the data to a monthly level.
3. Feature engineering – Feature engineering is the process of creating usable features for a machine learning model using domain knowledge. The domain knowledge collected during the interviews was used to create usable sales activity features for machine learning. These features were built using the initial data procured from different data streams of the company in step 1. The features built during the feature engineering process represented information on a statistical level. All the private information related to employees was anonymized as per GDPR regulations. It was thoroughly ensured that features don't track the activity of any employee on an individual level.

3.2.2.2 Feature scaling

The second feature scaling step followed was to normalize the values of the features using the min-max scaler. Min-max scaler was used to ensure that some feature values that have a significantly higher magnitude than others do not get higher importance by the machine learning algorithm while training. The standard setting of the min-max scaler was used, and the original values were scaled down a range of values between 0 to 1.

3.2.2.3 Sliding window method for creating lag variables

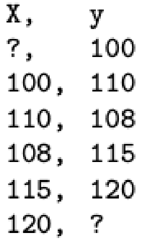
The scaled dataset prepared in section 3.2.2.2 was converted to suit a supervised learning problem. Approaching the time series forecasting as a supervised learning problem gives the freedom to use a wide range of linear and nonlinear supervised learning algorithms. During the process of making predictions on time series data, the input features from previous time steps like t-1, t-2, t-3, etc. are also fed into the supervised learning algorithm to predict the target variable. The supervised learning algorithm cannot access the previous time steps automatically, so the data frame needs to be adjusted to suit the prediction problem. This was done using the sliding window method [6]. Consider the following example to understand the sliding window method better.



```
time,  measure
1,     100
2,     110
3,     108
4,     115
5,     120
```

Figure 3-3: A simple univariate time series

Figure 3-3 adapted from [6] represents a simple univariate time series where a quantity represented by measure is recorded against time. The time in this univariate time series can be referenced to as the current time step, t. Let's assume that the objective is to perform unit step forecasting that uses the value of measure from the previous time step, t-1 to predict the value of measure at the current time step, t. The current time step, t is the future time step for, t-1. In order to achieve this using a supervised learning algorithm, the data would have to be reorganized in the way presented by Figure 3-4 adapted from [6]. The measure from the previous time step represented by X is now the input value for the current time step which forms the target variable represented by y.

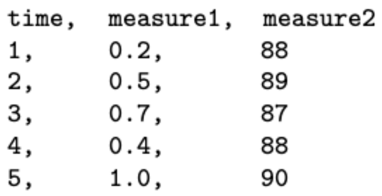


```
X,    y
?,    100
100,  110
110,  108
108,  115
115,  120
120,  ?
```

Figure 3-4: Univariate time series lagged by one-time step

The distinguishing factor of time series is that they are observations ordered by time and by using the sliding window method, the order of the time series remains maintained. It can also be observed from the image above that after applying the sliding window method, there is no previous time step observation X for the target variable y in the first instance. There is also no current time step target variable for X in the last data instance and these two rows of data must be removed before feeding the dataset into the machine learning algorithm.

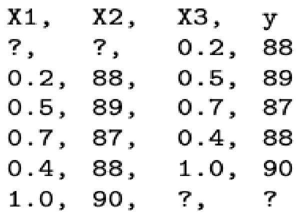
A multivariate time series has multiple observations recorded for the same time and is much more complex than univariate time series. The sliding window method can be extended to multivariate time series to prepare a multivariate time series for a supervised learning algorithm. The process can be better understood with the following example.



```
time,  measure1,  measure2
1,     0.2,       88
2,     0.5,       89
3,     0.7,       87
4,     0.4,       88
5,     1.0,       90
```

Figure 3-5: A multivariate time series

Figure 3-5 adapted from [6] represents a multivariate time series where two observations, measure 1 and measure 2 were recorded against time. Let's assume, the goal is to predict the value of measure 2 at the current time step using the values of measure 1 and measure 2 from the previous time steps and the value of measure 1 from the current time step. The multivariate time series data frame can be reorganized for the supervised learning model as represented by Figure 3-6 adapted from [6].



```
X1,  X2,  X3,  y
?,   ?,   0.2, 88
0.2, 88,  0.5, 89
0.5, 89,  0.7, 87
0.7, 87,  0.4, 88
0.4, 88,  1.0, 90
1.0, 90,  ?,   ?
```

Figure 3-6: Multivariate time series lagged by one-time step

X1 is the previous time step variable for measure 1 and X2 is the previous time step variable for measure 2. X3 is the current time step value for measure 1 and y is the target variable which contains the current time step values from measure 2 that need to be predicted. The values X1, X2, and X3 form the input variable for the supervised machine learning model and y forms the target variable. The instances in the dataset that do not contain values for either of the variable X1, X2, X3, or y would have to be removed before feeding the dataset into the machine learning algorithm.

The above examples only show the usage of the sliding window for reorganizing the dataset to build feature values from one previous time step. However, the window size of the sliding window method can be adjusted to include data from various previous time steps, t-1, t-2, t-3, and so on.

The raw dataset was readjusted to form a supervised learning problem using the sliding window method during this project. The initial dataset consists of multiple sales activities of the company (multivariable time series data) and the corresponding deals closed on a monthly level. The dataset was readjusted to use the deals closed in a particular month as the target variable which can be referenced to as current time step, t. The corresponding features were formed by the sales activity of the current month, t, and the lagged variables of the sales activity from previous months, t-n where n represents the number of months from previous time steps. In time series terminology, the dataset has been reframed for a single step forecasting using multivariate time series with different lags. Figure 3-7 summarizes the framework of the supervised learning problem formed for this project.

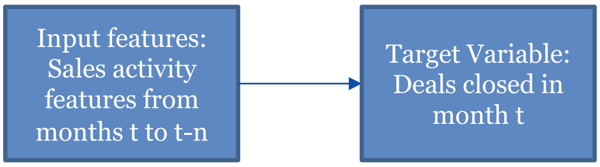

Input features: Sales activity features from months t to t-n
Target Variable: Deals closed in month t

Figure 3-7: Framework of the supervised learning problem for this project

The following examples can be used to solidify the understanding:

Example 1: When the dataset is lagged by 1 month, the data instance from 31-07-2017 would have deals closed from 31-07-2017 as the target variable. The sales activity features from 31-07-2017 and 30-06-2017 would form the input feature variables.

Example 2: When the dataset is lagged by 2 months, the data instance from 31-07-2017 would have the deals closed from 31-07-2017 as the target variable. The sales activity features from 31-07-2017, 30-06-2017, and 31-05-2017 would form the input variables.

Example 3: When the dataset is lagged by 3 months, the data instance from 31-07-2017 would have the deals closed from 31-07-2017 as the target variable. The sales activity features from 31-07-2017, 30-06-2017, 31-05-2017, and 31-04-2017 would form the input variables.

The same logic can be extended for identifying the target variable and input features for the data instance from 31-07-2017 as the lag of the month's increases.

3.2.3 Time series forecasting using support vector regression

Time series forecasting was performed on the data set using support vector regression in the following way.

1. Creating the training and test dataset– During the project, five different datasets containing lag variables from 1 to 5 were created using the sliding window method. The dataset with lag 1 contained input features from time instance t and time instance t-1. The dataset with lag 2 contained input features from time instance t, time instance t-1, and time instance t-2. Similarly, the dataset with lag 3 contained input features from time instance t, time instance t-1, time instance t-2, and time instance t-3. The same logic can be extended to understand the structure of the datasets with lags 4 and 5. After the datasets with different lags were created, each dataset was split into training and test set. It is a common practice to shuffle the data before creating the training and test set. However, this methodology does not work with time series data as the order of sequencing needs to be preserved while creating training and test set from a time series dataset. 80% of the data was used for training and 20% was used for testing.
2. Tuning the hyperparameters of the SVR model – The hyperparameters of the SVR model were thoroughly tuned while the training process for each of the five datasets with different lag variables. The hypermeters namely C, epsilon, gamma, and kernel were trained using the technique of GridsearchCV. The default cross-validation parameter is the 3-step cross-validation, but this shuffles the order of the instances during the training process. This kind of technique does not work for time series where the instances are time-dependent making their order significantly important. In order to overcome this limitation of normal cross-validation, a time series cross-validation was used. The default cross-validation was replaced by a 4 step time series cross-validation in GridsearchCV. A list of various hyperparameters was passed to GridsearchCV and the best performing hyperparameters for the SVR model trained on each of the five datasets were obtained. The hyperparameters that were passed to GridsearchCV are as follows.

Table 3.2: The set of hyperparameters passed to GridsearchCV

Kernel	Linear, RBF
C	0.1, 1.5, 10, 25, 50
Gamma	1e-2, 1e-3, 1e-4, 1e-5
Epsilon	0.1, 0,2, 0.3, 0.5

3. Performance Evaluation – The performance of the SVR models built on the training data for each of the five datasets with different lag variables was evaluated on the test dataset. This was done by using the mean absolute percentage error as the performance metric.

   Mean absolute percentage error is one of the most popular methods used metric to evaluate time series predictions. Moreover, another advantage of the mean absolute percentage error is that it is intuitive and easily understandable by humans [25].

3.2.4 Local explanations with SHAP and LIME

The best performing model based on the least mean absolute percentage error was interpreted using the python-based implementation of Kernal SHAP and LIME. The SHAP python-based implementation [26] was published by the original author of the paper Scott Lundberg. The LIME python-based implementation [27] was also published by the original author of the paper Marco Ribeiro. Ten prediction instances from the test dataset of the best performing model were chosen and the explanation (local interpretability) for each of them was obtained using both SHAP and LIME.

The explanation with LIME was generated using the following steps.

1. The LIME explainer was prepared by passing all the values from the training dataset. The parameters used while preparing the LIME explainer were mode and feature names. The mode was set to regression as the machine learning problem in this project was of regression type. The name of the features from the training dataset was passed to the parameter feature names.
2. The prepared LIME explainer was then used to produce the explanation for the prediction instances from the test dataset. Two parameters were passed to the explainer. The first one was the feature values of the prediction instance that needed to be explained. The second parameter was the trained machine learning model along with the predict attribute.
3. A human friendly explanation was then displayed using LIME explainer's pyplot figure function.

The explanation with SHAP was generated using the following steps.

1. The SHAP kernel explainer was prepared by passing the trained machine learning model with the predict attribute and all the values from the training dataset.
2. The values from the test dataset were passed through the prepared SHAP kernel explainer. This helped to obtain the SHAP values for all the features in each prediction instance of the test dataset.
3. A human-friendly explanation was displayed using the SHAP's summary plot attribute. SHAP's summary plot attribute produced the explanation for the required prediction instance from the test data and simultaneously displayed it in form of a human-understandable plot. Two parameters were passed to SHAP's summary plot attribute to produce the explanation.

The first parameter was the SHAP values of the prediction instance that needed to be explained (the values were obtained in step 2). The second parameter was the feature values for prediction instance that needed to be explained.

3.2.5 Human evaluation of interpretability

A human-computer interaction (HCI) study was conducted to evaluate the quality of the explanations produced by LIME and SHAP. The goal of the study was to evaluate if the explanations aid humans in understanding the decision-making process of the underlying machine learning model.

3.2.5.1 Structure of the study

To conduct the study, three different interactive web-based applications were built. The first application was used to evaluate the explanation generated by LIME, the second one was used to evaluate the explanation generated by SHAP and the third application acted as a baseline where no explainable AI (noXAI) technique was used. The study was conducted with a total of 60 human participants (application users), 20 for LIME, 20 for SHAP, and 20 for noXAI application. The optimal number of participants for each application was based on the research work in [24, p. 137]. Each application presented every participant with 10 cases of different prediction instances from the test dataset. The details are as follows:

1. The first web application consisted of the input to the machine learning model, output/prediction of the model, and explanation generated by LIME. The input during this project was the sales activity features with lags on a monthly level and the output was the sales deals closed on a monthly level. A total of 20 users were asked to interact with the application. Each user was presented with 10 different cases of input, output, and the corresponding LIME explanation. At the end of each case, the users were asked if the explanation helped them in understanding the prediction in a more understandable way. The response of the users was recorded as a binary, yes or no.
2. The second web application consisted of the input to the machine learning model, output/prediction of the model, and explanation generated by SHAP. A total of 20 users were asked to interact with the application. Each user was presented with 10 different cases of input, output, and the corresponding SHAP explanation. At the end of each case, the users were asked if the explanation helped them in understanding the prediction in a more understandable way. The response of the users was recorded as a binary, yes or no.
3. The third web application consisted of the input to the machine learning model, output, and no explanation (no XAI). A total of 20 users were asked to interact with the application. In the no XAI application, the users were presented with 10 different cases of input and output but no explanation about the decision-making process of the underlying model was given. At the end of each case, the users were asked if they were able to understand the prediction. The response of the users was recorded as a binary, yes or no.

The normalized version of the input data was presented to all the 60 participants due to privacy reasons. The 10 prediction instances (cases) were shown in the same order with their respective LIME, SHAP, and noXAI explanation to avoid any kind of bias that could have been caused due to their order.

3.2.5.2 Hypotheses testing

The aim of the study this HCI study was to evaluate the following hypothesis:

1. **H_a**: The number of times the study participant is able to understand the prediction with LIME explanation would be greater than no explanation. (LIME > noXAI)
2. **H_b**: The number of times the study participant is able to understand the prediction with SHAP explanation would be greater than no explanation. (SHAP > noXAI)
3. **H_c**: The number of times the study participant is able to understand the prediction with LIME explanation would be greater than SHAP explanation. (LIME > SHAP)

The hypotheses testing was conducted to check if the null hypotheses ($H_{ao,}$ $H_{bo,}$ H_{co}), the negation of the aforementioned three hypotheses could be rejected.

3.2.5.3 Choice of participants

No particular control was exercised over the study participants. The only thing that was ensured was that a set of 20 unique participants interacted with the LIME, SHAP, and noXAI applications, bringing the total count of the participants for the study to 60. This was done to ensure that the results for each application were unbiased as seen in [8, p. 1074], [9, pp. 5544-5545] and [24, p. 137].

3.2.5.4 Protocol of the study

The study was conducted remotely, and the following protocol was followed during the study:

1. The links of the applications were shared with the different sets of users. The introductory page of the applications thoroughly discussed the time series forecasting case along with all the relevant instructions related to the tasks that had to be performed by the users.
2. Additional questions and queries regarding the cases were answered over a call/text. However, it was made sure that no additional information about the LIME and SHAP explanations was shared even when the users had queries about them. Doing so would have defeated the purpose of the study and could have added bias in the user's mind. In any human evaluation of interpretability study, it is important for the users to be able to understand the explanations on their own. This can be seen from [8, pp. 1074-1075], [9, p. 5545] and [24, p. 138].
3. After the instructions, the users went through the 10 cases in their respective LIME, SHAP, and noXAI applications. At the end of each case in the LIME and SHAP applications, the users were asked if the explanation helped them in understanding the prediction in a more understandable way and the response was recorded as a yes or no. At the end of each case in the no XAI application, the users were asked if they were able to understand the prediction made and the response was recorded as a yes or no.
4. After the users went through all the cases in the application, they were presented with a questionnaire that had additional questions about their demographic, assessment of the explanations, and user experience with the application.

Figure 3-8 summarizes the structure of the study.

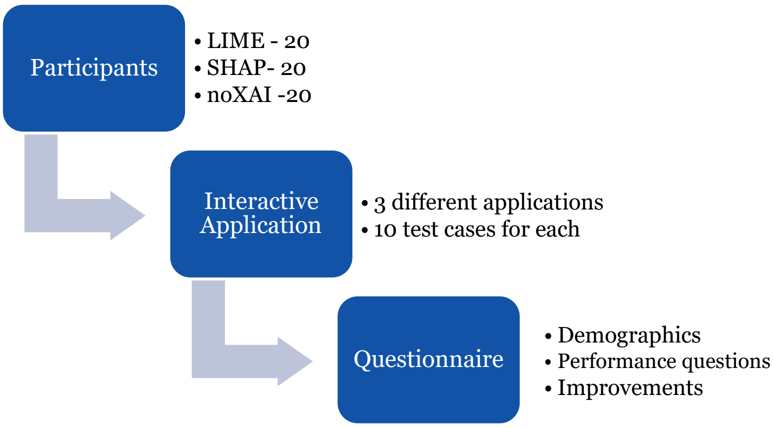


Figure 3-8: **Structure of the HCI study for human evaluation of interpretability**

3.2.5.5 Choice of questions

The choice of the questions for the questionnaire was adapted from [24, pp. 138-139]. The following questions were asked about the demographic in all three applications:

1. Age – Input: number
2. Gender – Input [multiple choice]: male, female, other
3. Highest educational qualification – Input [multiple choice]: pre-high school, high school, bachelor or equivalent, master or equivalent, Ph.D. or higher
4. Background in science, technology, engineering, or mathematics (STEM) – Input [Boolean]: yes or no.

The following questions were asked about the user's assessment of the explanations in the LIME and SHAP applications:

1. The users were asked if they had heard about explainable machine learning – Input [Boolean]: yes or no
2. The users were asked to describe explainable machine learning if they answered the previous question with a yes – Input: short text
3. The users were asked to rate their satisfaction level with the explanations - Input [Likert scale]: scale of 0-5
4. The users were asked if the explanations were good enough for them to trust the predictions - Input [Boolean]: yes or no
5. The users were asked to suggest possible improvements to the explanations that can help improve their understanding – Input: short text

The following questions were asked about the user experience in all three applications:

1. The users were asked to rate their satisfaction level with the user experience - Input [Likert scale]: scale of 0-5
2. The users were asked to suggest possible improvements to the user experience – Input: short text.

The following questions were asked about explanations only in the noXAI application:

1. The users were asked if could trust the predictions without appropriate explanations – Input [Boolean]: yes or no
2. The users were asked if the predictions would be more satisfying/trustable if they were supported by explanations: - Input [Boolean]: yes or no
3. The users were asked to suggest the kind of explanations that could be helpful – Input: short text

3.2.5.6 Method for hypotheses analysis

In order to understand the usefulness of the explanations for the human participants, comparative tests were conducted between the three different sets of application users (SHAP, LIME, and noXAI). The comparative tests were done for the testing of the aforementioned hypotheses. The hypotheses testing was conducted using a statistical two-sample t-Test. The t-Test was used to calculate the p-value and the p-value was used to reject or accept the null hypothesis.

A t-Test is a statistical test developed in 1908 to test if there is a significant difference between two datasets by comparing their mean distribution. A t-test is a very common test for hypothesis testing. The t-test is used to calculate the t-statistic using the mean, variance, and size of the sample datasets involved. Every t-statistic value calculated from a t-test has a corresponding p-value. The

Equation **3-1** represents the mathematical formula for calculating the t statistic value [28].

$$t = (m_A - m_B)/\sqrt{(S_A^2/n_A + S_B^2/n_B)}$$ **Equation 3-1 [28]**

The symbols m_A and m_B represent the means of the two sample datasets A and B, respectively. The symbols $S_A^2$and S_B^2 represent the variance of the two sample datasets whereas n_A and n_B represent the size of the two sample sets A and B, respectively.

The p-value approach is the most popular approach for hypothesis testing. It has been commonly used for hypothesis testing in the human evaluation of interpretability studies as seen in [8, p. 1076], [9, p. 5545] and [24, p. 141].

During hypothesis testing, a p-value is derived from the test statistic calculated using the collected sample datasets, under the assumption that the null hypothesis is true. The p-value is a probability measure that measures if the statistical difference between the sample datasets was a result of random chance. The p-value is evidence against the null hypothesis and is used to indicate the level of statistical significance. The p-value is measured between 0 and 1. A small p-value provides stronger evidence against the null hypothesis and indicates that there is a small probability that the results derived from the sample datasets occurred by chance. Scientifically, it is impossible to guarantee 100% non-randomness in the results derived from sample datasets during hypothesis testing. This is addressed by setting a significance (cutoff) level denoted by α. The results from a hypothesis testing are considered statically significant at a p-value of less than or equal to 0.05 (α). This implies that the probability of the results derived from hypothesis testing being random is less than 5%. The value of α is a design choice and is selected by the person who designs the hypothesis. Other acceptable choices of α are 0.10 where the results are considered marginally significant and 0.01 where the results are considered highly significant. This implies that the probability of the results derived from hypothesis testing being random is less than 1% and 10%, respectively. According to the Neyman-Pearson theory of hypothesis testing, this is known as the type 1 error rate [29].

The significance level (α) selected for hypotheses testing during this project was 0.05 and 0.01. The correlation between the demographics of the participants and their ability to understand the prediction in the LIME, SHAP, and noXAI case studies was also measured using the spearman correlation coefficient.

Spearman correlation was chosen because it captures the monotonic relationship between the variables instead of just a linear relationship and also works well with categorical variables like gender [30].

Figure 3-9 summarizes the method for hypotheses analysis.

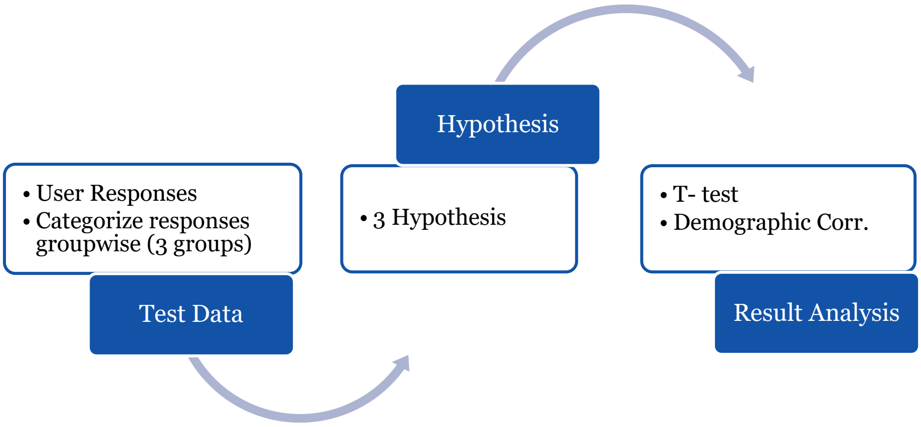


Figure 3-9: Analysis method used for hypotheses evaluation

3.2.6 Software technologies

The code for this project involved the use of two programming languages, Python 3.7.6 and Structured Query Language (SQL) version 15.0.

SQL was used to procure the data from the company servers stored in Amazon Web Services (AWS) cloud service called S3. The interface used to interact with the S3 is called AWS Athena. Athena is a tool that provides the ability to interact and procure the data in a S3 using SQL.

Python was used for data cleaning, data preparation, machine learning implementation, and interpreting the machine learning model. The integrated development environment (IDE) used to write the code was Jupyter Notebook.

The exact python libraries that were used during this project are as follows:

- Numpy – Library for performing mathematical operations on the data
- Pandas – Library for data preprocessing and data analysis
- Scikit-learn – Library for machine learning
- Matplotlib and Seaborn – Libraries for graphical visualizations and plots
- SHAP – Library used for SHAP interpretability
- LIME – Library used for LIME interpretability
- SPSS – Software for statistical analysis
- Google Docs – Human evaluation of interpretability

4 Results and Analysis

This chapter presents the results and analysis. The chapter is broadly divided into five sections: interview results, dataset preparation, time series forecasting with SVR, interpretability with LIME and SHAP and human evaluation of interpretability.

4.1 Interview results

This section discusses the results of the interviews conducted to build the feature dataset for time series forecasting.

4.1.1 The sales process of the company

The sales process of the company is fairly streamlined and universal for every sales employee of the company. The sales employees constantly reach out to new and old clients to enquire about their business needs and pitch new services that could potentially be beneficial for them. The process involves setting up a lot of meetings. The initial meetings are organized over google meet with a purpose of understanding business needs or pitching new services. If the initial meetings with the clients are fruitful, detailed proposals for clients are prepared by having internal discussions with designers, developers, managers, data scientist, etc. The final negotiations of the sales deals involve physical meetings with the clients. The journey of every sales deal is recorded in the company's CRM, Hubspot.

4.1.2 Data streams

The following data streams discovered during the interview process contain data related to company's sales activities:

- Google calendar – All the company meetings (internal and external) are booked using google calendar.
- Employee competence data – The employee competence data contained information about the job role (competence) of the employees.
- Revenue data – The revenue data contained information about the revenue generated by the company from various clients.
- Credit card data – Contains information about the transactions executed using the company credit cards.
- Company time logger – As a standard practice in the company, employees are required to mark the hours they spend working along with task they work on.
- Customer relationship management (CRM) tool – The CRM tool used by the company is Hubspot. A CRM tool is used to manage analyze the relationships with existing and potential clients.

The data streams contained ten years' worth of data, from 2010 – 2019.

4.1.3 Feature building on the basis of domain knowledge

The following features were build using the domain knowledge obtained during the interview process:

- Meetings by sales employees – All the meetings organized/attended by the employees with sales-based roles/competence. The primary reason behind adding this feature was that meetings by sales employees are acting as a proxy for the amount of effort being into procuring/closing sales deals.
- Meetings of sales employees with top clients - All the meetings organized/attended by the sales employees involving top 50 clients of the company. It was realized during the interview that the 80% of the company's revenue is recurring and comes from the top 50 clients. That is why it was it was integral to track the sales activities involving top clients.
- Meetings about top clients in the company - Meetings about top clients in the company is a feature the encapsulates the internal meetings conducted in the company about the top 50 clients. As mentioned earlier, most of the company revenue comes from the top 50 clients. Internal meetings about top clients essentially track new ideas and proposals being prepared to pitch to top clients.
- Physical meeting of sales employees with client – Physical meetings with clients indicate that a sales deal is in the final stages of being closed. They were extracted using the credit card data.
- Time spent in meetings by sales employees – The time spent in the meetings by employees with sales- based roles is another great proxy for the amount of effort being poured into procuring/closing sales deals.
- Deals closed – The target variable for the ML model was obtained from the CRM (Hubspot).

All the features were accumulated on a monthly level and formed a regular time series. All the features represented information on a statistical level. All the private information related to employees was anonymized as per GDPR regulations. It was thoroughly ensured that features don't track the activity of any employee on an individual level.

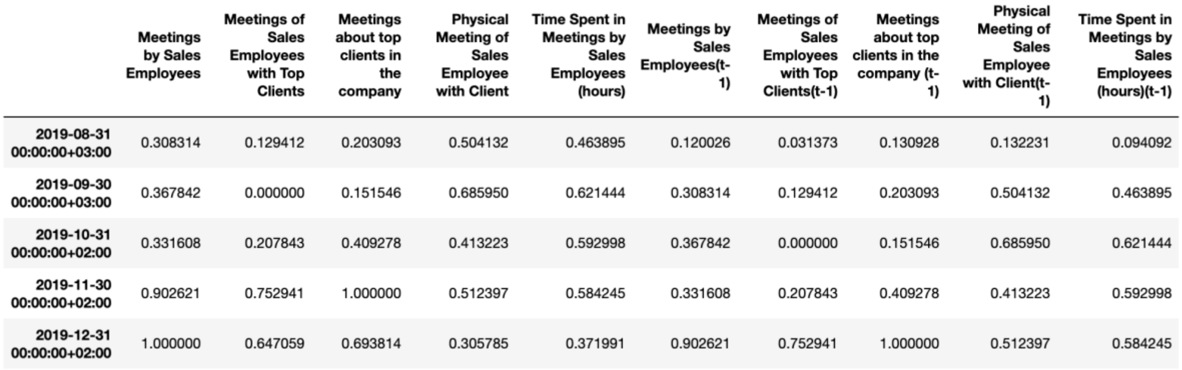
Table 4.1 summarizes the features used for time series forecasting. The first column contains the name of the features, the second column contains the data sources of the company utilized to create the features and the third column contains the category of the feature variable.

Table 4.1: Features for time series forecasting

Feature	**Source**	**Category**
Meetings by sales employees	Google calendar, Employee competence	Numerical Variable
Meetings of sales employees with top clients	Google calendar, Revenue data	Numerical Variable
Meetings about top clients in the company	Google calendar	Numerical Variable
Physical meeting of sales employees with client	Credit card data	Numerical Variable
Time spent in meetings by sales employees	Company time logger	Numerical Variable
Deals closed	Company CRM (Hubspot)	Numerical Variable

4.2 Dataset preparation

The final datasets (with lag 1 to 5) for time series forecasting were prepared in three steps: building the raw dataset on the basis of the interview results, feature scaling, and creating lag variables using the sliding window method. Figure 4-1 depicts a sample of five data instances from the dataset with a lag of 1.



	Meetings by Sales Employees	Meetings of Sales Employees with Top Clients	Meetings about top clients in the company	Physical Meeting of Sales Employee with Client	Time Spent in Meetings by Sales Employees (hours)	Meetings by Sales Employees(t-1)	Meetings of Sales Employees with Top Clients(t-1)	Meetings about top clients in the company (t-1)	Physical Meeting of Sales Employee with Client(t-1)	Time Spent in Meetings by Sales Employees (hours)(t-1)
2019-08-31 00:00:00+03:00	0.308314	0.129412	0.203093	0.504132	0.463895	0.120026	0.031373	0.130928	0.132231	0.094092
2019-09-30 00:00:00+03:00	0.367842	0.000000	0.151546	0.685950	0.621444	0.308314	0.129412	0.203093	0.504132	0.463895
2019-10-31 00:00:00+02:00	0.331608	0.207843	0.409278	0.413223	0.592998	0.367842	0.000000	0.151546	0.685950	0.621444
2019-11-30 00:00:00+02:00	0.902621	0.752941	1.000000	0.512397	0.584245	0.331608	0.207843	0.409278	0.413223	0.592998
2019-12-31 00:00:00+02:00	1.000000	0.647059	0.693814	0.305785	0.371991	0.902621	0.752941	1.000000	0.512397	0.584245

Figure 4-1: A sample from the dataset with a lag of 1

As shown in Figure 4-1 the final dataset with a lag of 1 contains the features from time instance t (or t-0) and the features from time instance t-1 (lag of 1), represented with an additional t-1 in the column name. For example, the data instance with index "30-09-2019" contains feature values from September 2019 and the feature values from August 2019 (represented with t-1 in column name). Similarly, the data instance with index "31-10-2019" contains feature values from October 2019 and the feature values from September 2019. This helps in capturing the temporal relationships between the feature values and target variable while training the machine learning model. It can also be seen from Figure 4-1 that the feature values were normalized using feature scaling. The same logic can be extended to visualize the datasets with a lag of 2,3,4 and 5.

4.3 Time series forecasting with SVR

This section presents the results of time series forecasting with support vector regression.

Table 4.2: Performance summary of support vector regression models

Lag in the dataset	**Mean absolute percentage error**	**Best hyperparameters**
1	11.32 %	C:1.5, epsilon: 0.2, gamma: 0.01, kernel: RBF
2	10.72 %	C:1.5, epsilon: 0.1, gamma: 0.1, kernel: RBF
3	**9.29 %**	**C:1.5, epsilon: 0.1, gamma: 0.1, kernel: RBF**
4	11.24 %	C:10, epsilon: 0.1, gamma: 0.01, kernel: RBF
5	11.41 %	C:1.5, epsilon: 0.1, gamma: 0.1, kernel: RBF

Table 4.2 summarizes the performance of time series forecasting models built with SVR. The first column presents the amount of lag in the dataset used to train SVR, the second column presents the mean absolute percentage error on the test dataset and the third column presents the best set of hyperparameters for that particular model.

As shown in Table 4.2, the machine learning model trained using the dataset with a lag of 3 (presented in bold) was the best performing model with a mean absolute percentage error of 9.29 %. The best model was further interpreted using the model agnostic interpretability techniques, LIME and SHAP.

4.4 Interpretability with LIME and SHAP

This section presents the explanations generated as a result of interpreting the best performing SVR model with LIME and SHAP. Figure 4-2 depicts the LIME explanation for one prediction instance from the test dataset. Figure 4-3 depicts the SHAP explanation for the same prediction instance. The prediction instance is the number of sales deals closed in December 2018.

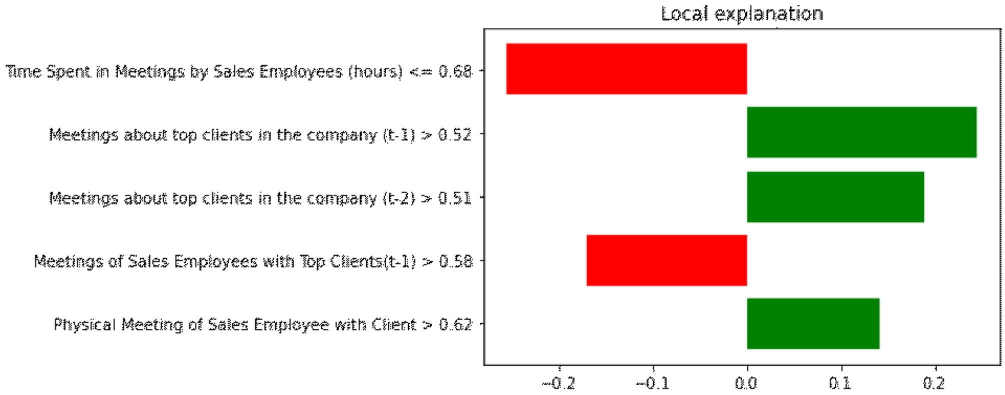


Figure 4-2: An example of explanation generated by LIME

The x-axis of the LIME explanation (Figure 4-2) represents the magnitude of feature impact on the model output/prediction and the y-axis represents the name of the feature variables. The first attribute of the LIME explanation worth noticing is that the features are arranged (top to bottom) in the descending order of their importance (positive or negative). The explanation only contains the top 5 features. The red-colored bar signifies that a particular feature had a negative impact on the model output and the green colored bar signifies that a particular feature had a positive impact on the model output. The length of the bar is representative of the magnitude of impact. The LIME explanation also tries to provide a reasoning behind the positive or negative impact of a feature on the model output. This reasoning is in the form of cut off values that is auto generated by LIME and can be seen in the y-axis along with the names of the feature variables.

The LIME explanation in Figure 4-2 is conveying the following information. The most important feature contributing towards the prediction of sales deals closed in December 2018 is “Time Spent in Meetings by Sales Employees (hours)”. The feature had a very high negative impact because its value (normalized) was less than 0.68.

The second most important feature is “Meetings about top clients in the company” held in November 2018 (represented with t-1). This feature had a positive impact on the predicted number of sales deals closed in December 2018 because its value (normalized) was greater than 0.52.

Similarly, the third most important feature is “Meetings about top clients in the company” held in October 2018 (represented with t-2). The feature had a positive impact on the predicted number of sales deals in November 2018 because its value (normalized) was greater than 0.51.

However, sometimes all elements of the LIME explanation may not be easily interpretable, at least by lay humans. For example, the fourth most important feature according to LIME explanation is “Meetings of Sales Employees with Top Clients” held in October 2018 (represented with t-1). According to LIME, the feature had a negative impact on the predicted number of sales deals closed in December 2018 because its value (normalized) was greater than 0.58. It seems difficult to interpret the fact that a greater number of meetings with top clients led to a negative impact. Perhaps, an explanation of this sort might make more sense to domain experts.

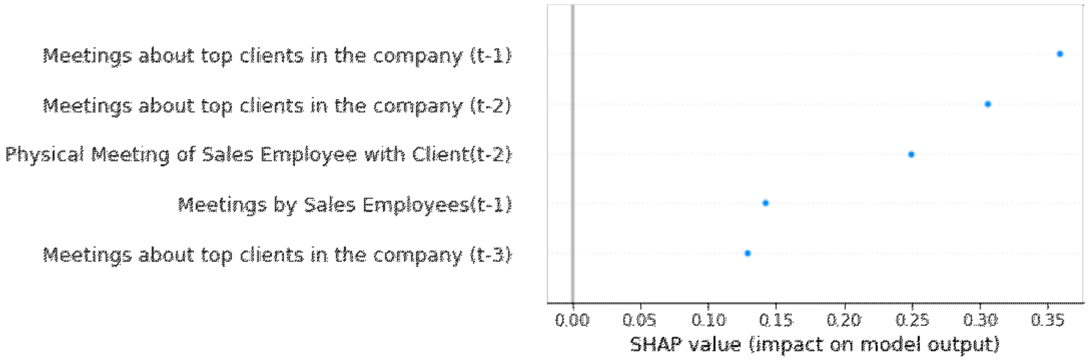


Figure 4-3: An example of explanation generated by SHAP

As shown in Figure 4-3, the x-axis of SHAP explanation also represents the magnitude of feature impact on the model output/prediction which in the case of SHAP is measured in terms of shapley values. The explanation only contains the top 5 features. The y-axis represents the name of feature variables. The features are arranged (top to bottom) in the descending order of their importance. Unlike LIME, SHAP explanations do not show any reasoning behind the positive or negative impact of a feature on the model output. The magnitude of the impact is visually displayed with the help of dots in SHAP explanation as opposed to bars in LIME. Since LIME and SHAP are two different algorithms, the explanations produced by them for the same prediction instance also differ slightly. According to the SHAP explanation, the top 5 features only had a positive impact on the prediction as opposed to the LIME explanation where 2 out of the top 5 features were shown to have a negative impact. 3 out of the top five features from SHAP and LIME explanations are the same whereas 2 are different.

According to SHAP explanation, the most important feature contributing towards the prediction of sales deals closed in December 2018 is “Meetings about top clients in the company” held in November 2018 which is different from the most important feature according to the LIME. According to LIME explanation, this is the second most important feature.

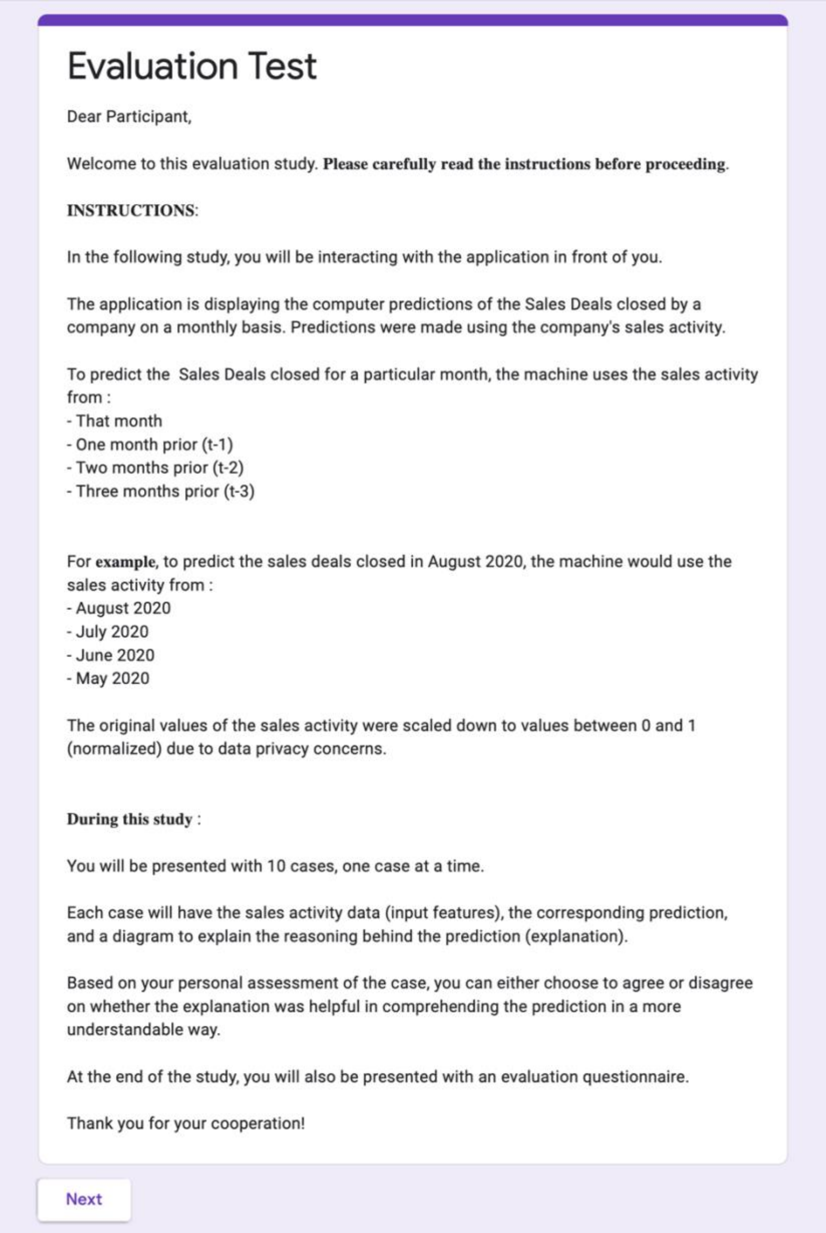
The second most important feature contributing towards the prediction of the sales deals closed in November 2018 is “Meetings about top clients in the company” held in October 2018 which is the same as the second most important feature according to the LIME.

4.5 Human evaluation of interpretability

This section presents and analyses the results of the human evaluation of interpretability studies.

4.5.1 LIME, SHAP, and noXAI applications for human evaluation

This section presents the LIME, SHAP, and noXAI applications built for human evaluation of interpretability. Figure 4-4 depicts the instructions presented to the participants at the beginning of the evaluation test in the LIME and SHAP applications.



Evaluation Test

Dear Participant,

Welcome to this evaluation study. **Please carefully read the instructions before proceeding**.

INSTRUCTIONS:

In the following study, you will be interacting with the application in front of you.

The application is displaying the computer predictions of the Sales Deals closed by a company on a monthly basis. Predictions were made using the company's sales activity.

To predict the Sales Deals closed for a particular month, the machine uses the sales activity from :
- That month
- One month prior (t-1)
- Two months prior (t-2)
- Three months prior (t-3)

For **example**, to predict the sales deals closed in August 2020, the machine would use the sales activity from :
- August 2020
- July 2020
- June 2020
- May 2020

The original values of the sales activity were scaled down to values between 0 and 1 (normalized) due to data privacy concerns.

During this study :

You will be presented with 10 cases, one case at a time.

Each case will have the sales activity data (input features), the corresponding prediction, and a diagram to explain the reasoning behind the prediction (explanation).

Based on your personal assessment of the case, you can either choose to agree or disagree on whether the explanation was helpful in comprehending the prediction in a more understandable way.

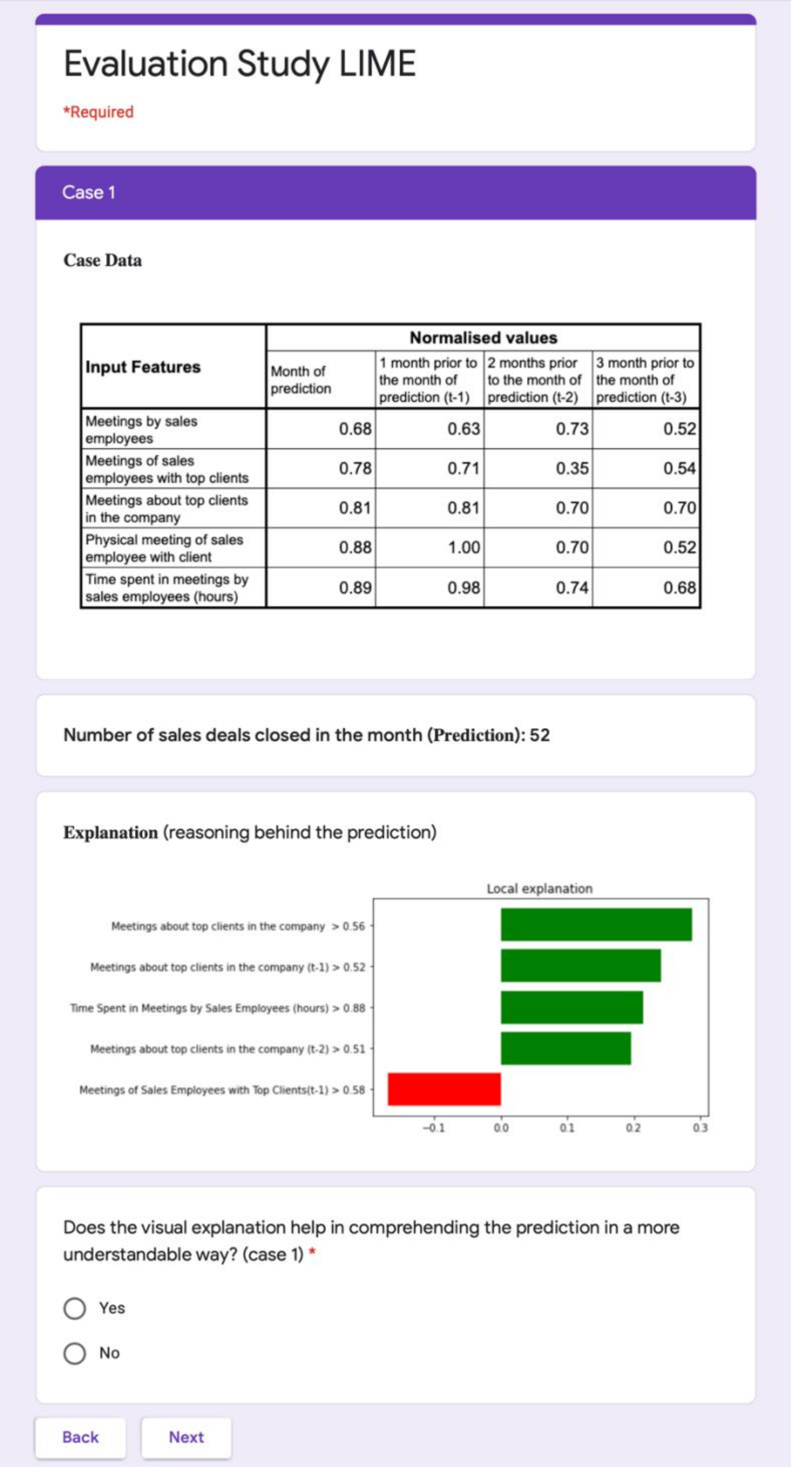
At the end of the study, you will also be presented with an evaluation questionnaire.

Thank you for your cooperation!

Figure 4-4: Instructions presented to the human participants in the LIME and SHAP applications

Figure 4-5 depicts one case from the LIME application. As shown in the image, the participants were presented with normalized input features, the prediction/output of the model, and the LIME explanation to highlight the reasoning behind the prediction made by the machine learning model.

The participants were asked if the explanation helped them in understanding the prediction in a more understandable way, at the end.



Evaluation Study LIME

*Required

Case 1

Case Data

Input Features	**Normalised values**			
	Month of prediction	1 month prior to the month of prediction (t-1)	2 months prior to the month of prediction (t-2)	3 month prior to the month of prediction (t-3)
Meetings by sales employees	0.68	0.63	0.73	0.52
Meetings of sales employees with top clients	0.78	0.71	0.35	0.54
Meetings about top clients in the company	0.81	0.81	0.70	0.70
Physical meeting of sales employee with client	0.88	1.00	0.70	0.52
Time spent in meetings by sales employees (hours)	0.89	0.98	0.74	0.68

Number of sales deals closed in the month (**Prediction**): 52

Explanation (reasoning behind the prediction)

Local explanation
Meetings about top clients in the company > 0.56
Meetings about top clients in the company (t-1) > 0.52
Time Spent in Meetings by Sales Employees (hours) > 0.88
Meetings about top clients in the company (t-2) > 0.51
Meetings of Sales Employees with Top Clients(t-1) > 0.58
−0.1
0.0
0.1
0.2
0.3

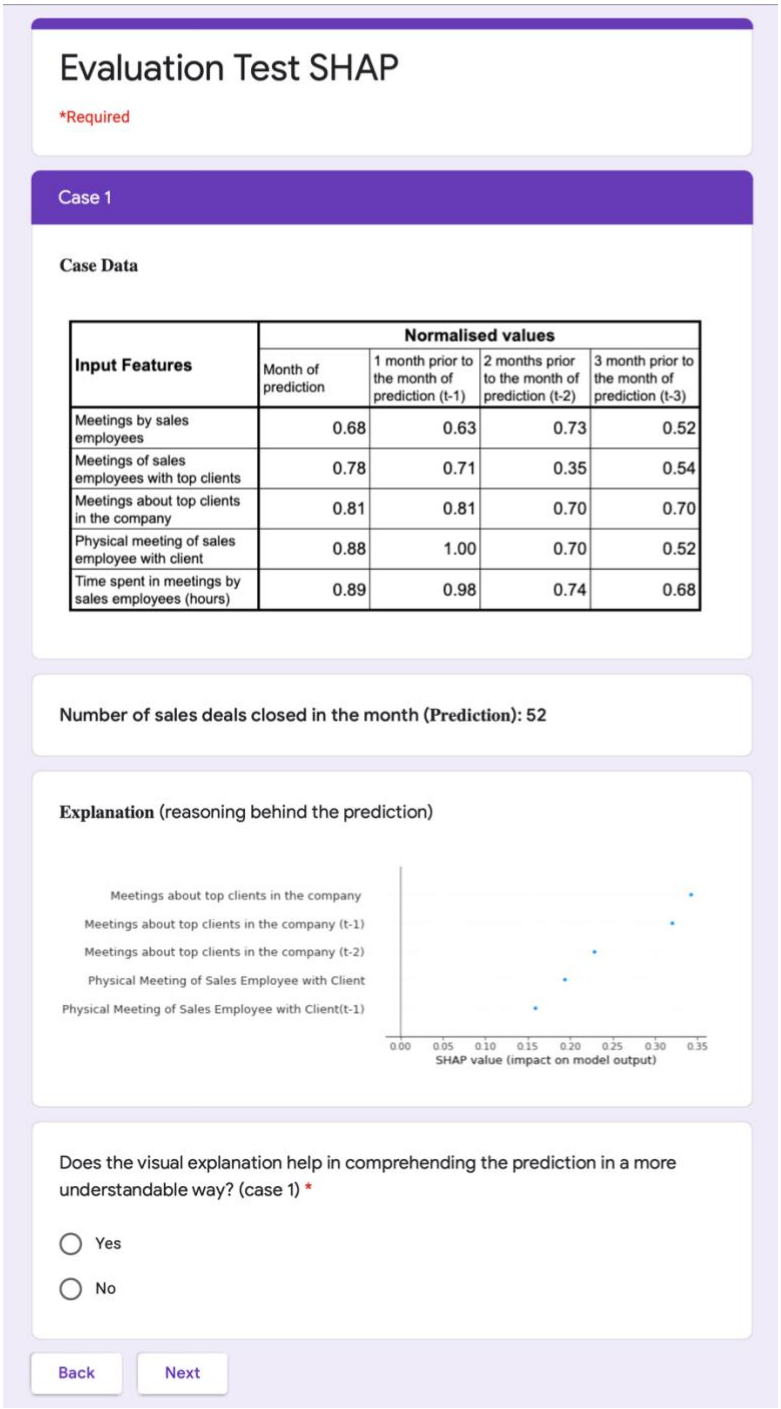
Does the visual explanation help in comprehending the prediction in a more understandable way? (case 1) *

○ Yes

○ No

Figure 4-5: One case instance from the LIME application

Figure 4-6 depicts one case from the SHAP application. As shown in the image, the participants were presented with normalized input features, the prediction/output of the model, and the LIME explanation to highlight the reasoning behind the prediction made by the machine learning model. The participants were asked if the explanation helped them in understanding the prediction in a more understandable way, at the end.



Evaluation Test SHAP

*Required

Case 1

Case Data

Input Features	Normalised values			
	Month of prediction	1 month prior to the month of prediction (t-1)	2 months prior to the month of prediction (t-2)	3 month prior to the month of prediction (t-3)
Meetings by sales employees	0.68	0.63	0.73	0.52
Meetings of sales employees with top clients	0.78	0.71	0.35	0.54
Meetings about top clients in the company	0.81	0.81	0.70	0.70
Physical meeting of sales employee with client	0.88	1.00	0.70	0.52
Time spent in meetings by sales employees (hours)	0.89	0.98	0.74	0.68

Number of sales deals closed in the month (**Prediction**): 52

Explanation (reasoning behind the prediction)

Meetings about top clients in the company
Meetings about top clients in the company (t-1)
Meetings about top clients in the company (t-2)
Physical Meeting of Sales Employee with Client
Physical Meeting of Sales Employee with Client(t-1)
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
SHAP value (impact on model output)

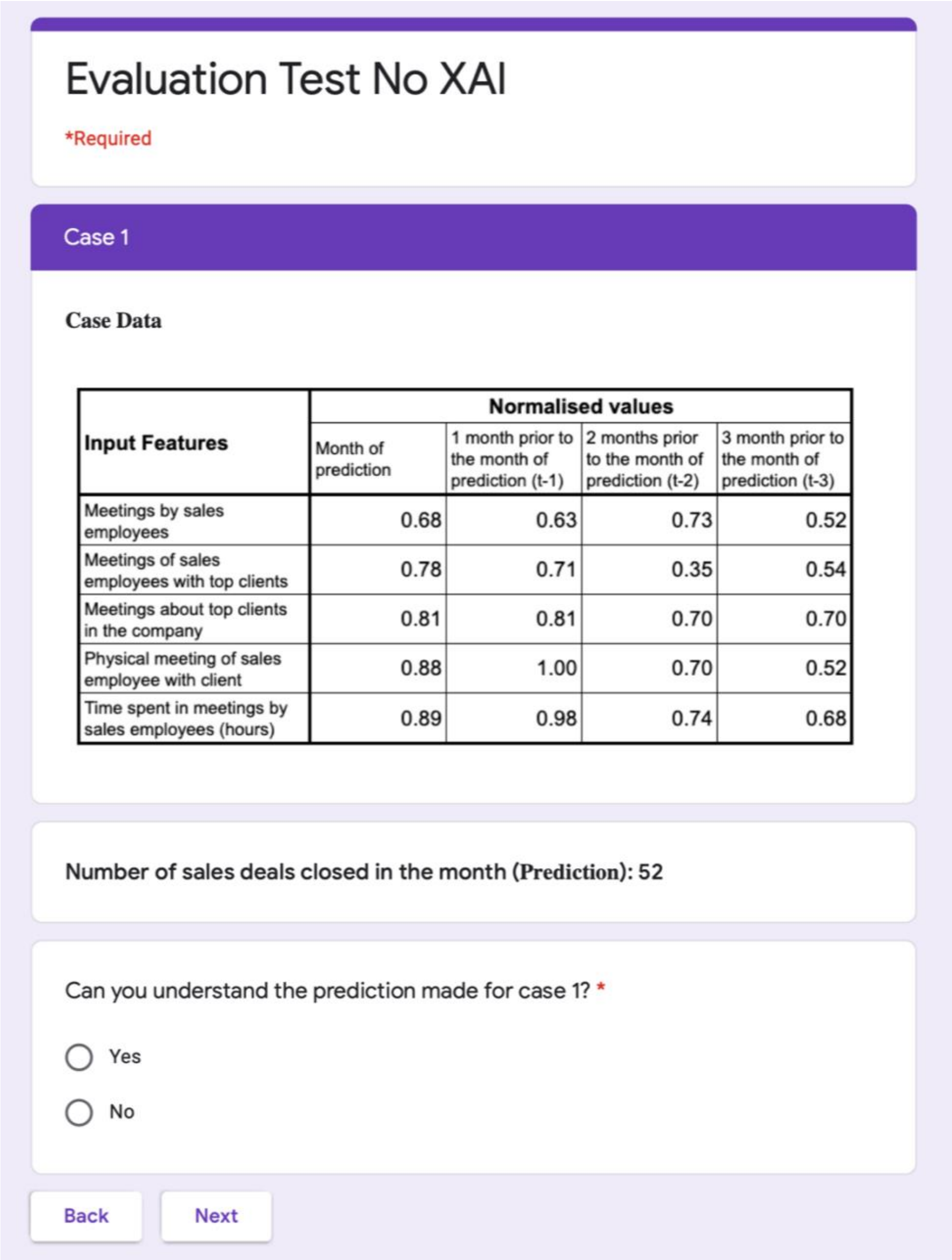
Does the visual explanation help in comprehending the prediction in a more understandable way? (case 1) *

○ Yes

○ No

Figure 4-6: One case instance from the SHAP application

Figure 4-7 depicts one case from the noXAI application. As shown in the image, the participants were presented with normalized input features and the prediction/output of the model. The participants were asked if they were able to understand the prediction, at the end.



Evaluation Test No XAI

*Required

Case 1

Case Data

Input Features	Normalised values			
	Month of prediction	1 month prior to the month of prediction (t-1)	2 months prior to the month of prediction (t-2)	3 month prior to the month of prediction (t-3)
Meetings by sales employees	0.68	0.63	0.73	0.52
Meetings of sales employees with top clients	0.78	0.71	0.35	0.54
Meetings about top clients in the company	0.81	0.81	0.70	0.70
Physical meeting of sales employee with client	0.88	1.00	0.70	0.52
Time spent in meetings by sales employees (hours)	0.89	0.98	0.74	0.68

Number of sales deals closed in the month (**Prediction**): 52

Can you understand the prediction made for case 1? *

○ Yes

○ No

Figure 4-7: One case instance from the noXAI application

4.5.2 Demographic analysis

This section presents the demographic analysis of the human evaluation of interpretability study participants.

Table 4.3: Demographic of study participants

Methods	**Total**	**Gender**			**Highest Degree**			**STEM**		**Age (years)**
		Male	Female	Others	Master's	Bachelor's	High School	Yes	No	
noXAI	20	12	8	0	13	7	0	12	8	20 (2), 22, 24, 25 (2), 26 (2), 29 (1), 30 (1), 31 (2), 33, 34(2), 38, 41, 47, 50, 51
LIME	20	11	8	1	8	10	2	17	3	18, 20, 21, 23 (3), 24 (5), 25(2), 27, 28 (2), 29, 31, 32, 55
SHAP	20	15	5	0	10	7	3	15	5	21 (2), 22 (2), 24 (2), 25 (3), 26 (4), 27 (3), 28, 29, 34, 35

Table 4.3 summarizes the demographics of the study participants of LIME, SHAP, and noXAI studies.

As deduced from Table 4.3, 10% of the participants for the no XAI study were aged equal to or less than 20 years, 40% were aged between 21 - 30 years, 30% were aged between 31 - 40 years, 15% were aged between 41 – 50 years and the rest 5% were aged between 51 – 60 years.

Out of the 20 participants, 60% of the study participants for the no XAI evaluation study were male and the rest 40% were female. 65% of the participants had a master's or equivalent degree and 35% had a bachelor's degree as their highest educational qualification. 60% of the participants had an educational background in STEM and 40% did not.

Whereas for the LIME study, 10% of the participants were aged equal to or less than 20 years, 75% were aged between 21 - 30 years, 10% were aged between 31 - 40 years, and the rest 5% were aged between 51 – 60 years.

Out of the 20 participants, 55% of the study participants for the LIME evaluation study were male, 40% were female and 5% identified as non-binary. 40% of the participants had a master's or equivalent degree, 50% had a bachelor's degree and 10% had high school as their highest educational qualification. 85% of the participants had an educational background in STEM and 15% did not.

For the SHAP study, 90% of the participants were aged between 21 - 30 years and the rest 10% were aged between 31 - 40 years.

Out of the 20 participants, 75% of the study participants for the SHAP evaluation study were male and the rest 25% were female. 50% of the participants had a master's or equivalent degree, 35% had a bachelor's degree and 15% had high school as their highest educational qualification. 75% of the participants had an educational background in STEM and 25% did not.

From the overall analysis it can be deduced that for noXAI and LIME case studies, the gender ratio of the participants was almost equal. However, for the SHAP case study, the ratio was skewed more towards the male gender. It can also be seen that for the noXAI case, the majority of participants were split between the age bracket of 21-30 and 31-40 years. However, for the LIME and SHAP cases, the majority of participants were from the age bracket of 21-30 years. The majority of participants for all three studies had a bachelor's or master's degree. The majority of participants for all three studies were from STEM background.

4.5.3 Results of the human evaluation study

This section presents the results of the human evaluation study. Table 4.4 depicts the sum, mean, and median of the responses collected from the participants during the study.

Table 4.4: Results of the human evaluation study

	Statistical Measure	**LIME**	**SHAP**	**noXAI**
Yes	Sum	86	82	7
	Mean	**4.30**	**4.10**	**0.35**
	Median	5	5	0
No	Sum	114	118	193
	Mean	5.70	5.90	9.59
	Median	5	5	10

During the LIME study, for 86 out of 200 cases, the participants said that the LIME explanation was helpful in understanding the prediction. During the SHAP study, for 82 out of 200 cases the participants said that the SHAP explanation was helpful in understanding the prediction. Whereas, during the study where no explanation was shown to the participants, for only 7 out of 200 cases the participants said that they were able to understand the prediction.

Similarly, it can be seen from Table 4.4 that for 114 out of 200 cases the participants said that the LIME explanation was not helpful in understanding the prediction. During the SHAP study, for 118 out of 200 cases the participants said that the SHAP explanation was not helpful in understanding the prediction. In comparison to SHAP and LIME, during the noXAI study, for 193 out of 200 cases the participants said that they were not able to understand the prediction.

It is very evident from the results presented in Table 4.4 that the LIME and SHAP explanations greatly helped the study participants in understanding the prediction of the machine learning model over the baseline case where no explanation was presented. The LIME explanation helped the participants in understanding the prediction for 4.30 out of 10 cases on average. The SHAP explanation helped the participants in understanding the prediction for 4.10 out of 10 cases on average. In comparison to LIME and SHAP cases, the participants of the noXAI study were able to understand the prediction for 0.35 cases out of 10 on average, which is negligible. The results of the human evaluation study align very well with the assumptions made while designing the hypotheses $\mathbf{H_a}$ and $\mathbf{H_b}$.

It can also be seen from the mean/median of the "yes" responses presented in Table 4.4 that the LIME and SHAP explanations provided an almost equal amount of help to human participants in understanding the predictions. However, the LIME explanation does seem to be a little more helpful by a very small margin. This implies that the results support the hypothesis $\mathbf{H_c}$ to a very little extent.

4.5.4 Hypotheses analysis

This section presents the results of the two-sample t-tests along with the hypotheses analysis.

Table 4.5: Hypothesis analysis. Significance: *p < 0.05, **p < 0.01

t-test	Hypothesis	p-value (two tailed)	p-value (two tailed)
LIME > noXAI	H_a	**0.00008***	**0.00008****
SHAP > noXAI	H_b	**0.001***	**0.001****
LIME > SHAP	H_c	0.867	0.867

Table 4.5 presents the p-values derived from the t-tests done for hypotheses testing. The significance level for the hypothesis testing was set at 0.05 and 0.01. The p-values that are statistically significant at the level of $p < 0.05$ and $p < 0.01$ are in bold.

On the basis of the p-values presented in Table 4.5, the null hypothesis H_{a0} and H_{b0} can be rejected and the hypothesis (alternate) H_a and H_b can be accepted. However, on the basis of the calculated p-value, there was a failure to reject the null hypothesis H_{c0} and accept the hypothesis H_c. It is hard to generalize the results due to the small sample size.

As presented in Table 4.4, the mean/median number of cases where LIME and SHAP explanations were helpful are almost equal. However, the mean number of cases where LIME explanation was helpful (4.30 out of 10) is slightly greater than the mean number of cases where the SHAP explanation was helpful (4.10 out of 10), so it supports the hypothesis H_c to a little extent. Increasing the sample size for the test could help reach more statistically significant results regarding hypothesis H_c.

Table 4.6: Two sample t-Test assuming unequal variance (LIME, noXAI)

	LIME	noXAI
Mean	4.30	0.35
Standard Deviation	3.495	1.137
Variance	12.221	1.292
Observations	20	20
t Stat	4.805	
P(T<=t) two-tail	**0.00008**	

Table 4.6 presents the results of the two-sample t-test performed to test the validity of the hypothesis H_a. Participant responses that said that they were able to understand the prediction during the LIME and noXAI studies made up the two samples for this t-test. The mean, standard deviation, and variance elaborate the distribution difference between the two samples. The mean for the LIME case study indicates that the LIME explanation helped the participants in understanding the prediction in 4.30 out of 10 cases on average. Whereas for the noXAI case study, the participants were only able to understand the prediction in 0.35 (practically 0) out of 10 cases on average.

The t stat represents the t statistic and P(T<=t) represents the p-value obtained as a result of the test for hypothesis H_a. Since the p-value obtained from the test is significant at $p < 0.01$, it indicates that there is less than 1% probability that the results of the test are random.

Table 4.7: Two sample t-Test assuming unequal variance (SHAP, noXAI)

	SHAP	**noXAI**
Mean	4.10	0.35
Standard Deviation	3.972	1.137
Variance	15.777	1.292
Observations	20	20
t Stat	4.059	
P(T<=t) two-tail	**0.001**	

Table 4.7 presents the results of the two-sample t-test performed to test the validity of hypothesis H_b. Participant responses that said that they were able to understand the prediction during the SHAP and noXAI studies made up the two samples for this t-test. The mean, standard deviation, and variance elaborate the distribution difference between the two samples. The mean for the SHAP case study indicates that the SHAP explanation helped the participants in understanding the prediction in 4.10 out of 10 cases on average.

The t stat represents the t statistic and P(T<=t) represents the p-value obtained as a result of the test for hypothesis H_b. Since the p-value obtained from the test is significant at $p < 0.01$, it indicates that there is less than 1% probability that the results of the test are random.

Table 4.8: Two sample t-Test assuming unequal variance (LIME, SHAP)

	LIME	**SHAP**
Mean	4.30	4.10
Standard Deviation	3.496	3.972
Variance	12.221	15.777
Observations	20	20
t Stat	0.169	
P(T<=t) two-tail	**0.867**	

Table 4.8 presents the results of the two-sample t-test performed to test the validity of hypothesis H_c. Participant responses that said that they were able to understand the prediction during the LIME and SHAP studies made up the two samples for this t-test. The mean, standard deviation, and variance elaborate the distribution difference between the two samples.

The t stat represents the t statistic and P(T<=t) represents the p-value obtained as a result of the test for hypothesis H_c. The p-value obtained from the test is greater than 0.05 and not significant at a significance level of 0.05 and 0.01.

4.5.5 Demographic correlations

This section presents the results of the demographic correlation analysis performed to measure the correlation between the demographics of the participant and their ability to understand the prediction in the LIME, SHAP, and noXAI case studies. The correlations are presented in Table 4.9.

Table 4.9: Demographic correlations. Significance: *p < 0.05

	Age	**Gender**	**Education**	**STEM**	**Knowledge about XAI**
noEXP (correlation)	0.169	-0.342	0.307	0.342	-
noEXP (p- value)	0.476	0.140	0.187	0.140	-
LIME (correlation)	**-0.476**	**-0.497**	-0.165	**0.479**	0.172
LIME (p- value)	0.034*	0.026*	0.488	0.033*	0.468
SHAP (correlation)	-0.261	0.021	-0.290	-0.373	-0.090
SHAP (p-value)	0.266	0.931	0.215	0.105	0.706

There were no significant correlations between most demographic attributes of the participants and their ability to understand the prediction.

However, some moderately high correlations were discovered between the age, gender, and STEM attributes of the participants and their ability to understand the prediction in the LIME case study. The correlations are presented in bold in Table 4.9. The analysis of the significant correlations is as follows:

1. A negative correlation was discovered between age and the ability of the participants to understand the predictions using the LIME explanation. This implies that as age increases, the understanding of the prediction decreases.
2. A negative correlation was discovered between gender and the ability of the participants to understand the predictions using LIME explanation. This implies that a higher understanding of the predictions correlates with gender male.
3. A positive correlation was discovered between STEM background and the ability of the participants to understand the predictions using LIME explanation. This implies that a higher understanding of the predictions correlates with participants having a STEM background.

The p-values in Table 4.9 depict the statistical significance of the correlations. A low p-value indicates a high statistical significance. The correlation between age, gender, and STEM background of the participants and their ability to understand the predictions using LIME explanation is statistically significant at $p < 0.05$.

It does seem quite logical that younger participants were able to make more sense out of the predictions using LIME explanation, possibly due to their affinity with computer algorithms and applications. It also makes sense that the LIME explanation was more helpful to participants with STEM background in understanding the predictions. This could be because people with STEM background are more used to graphs and numbers. The ability to understand the predictions using LIME has a correlation with gender male. This is possible because of the fact that there are more males in STEM-related fields.

4.5.6 Analysis of subjective ratings, trust, and user experience

This section presents the analysis of subjective ratings of the explanations, users experience ratings of the three studies and the trust in the predictions.

4.5.6.1 Subjective ratings

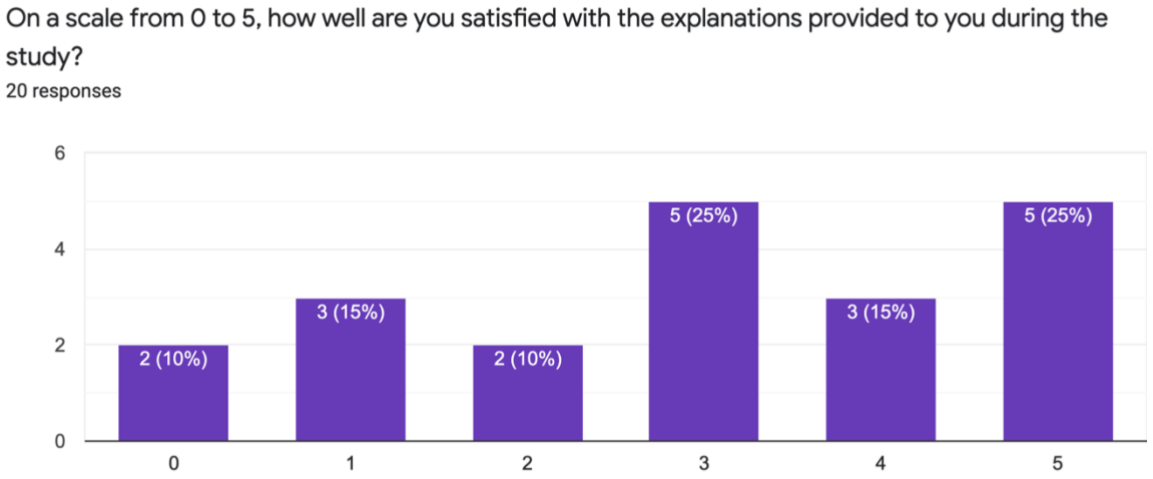
The participants of the LIME and SHAP studies were asked to rate their satisfaction level with LIME and SHAP explanations on a Likert scale of 0 to 5.

Table 4.10: Subjective ratings of the explanations

Statistical Measure	**LIME**	**SHAP**
Mean	**2.95**	**2.6**
Median	3	2.5
Standard deviation	1.70	1.46

Table 4.10 presents the mean, median, and standard deviation of the subjective ratings given by the participants for LIME and SHAP explanations. It can be observed that the mean/median satisfaction level with LIME explanations was higher than the mean/median satisfaction level of the SHAP explanations. This very well aligns with the results of the human evaluation study and the assumptions made while designing H_c. The participants found LIME explanations more helpful than SHAP explanations.

Figure 4-8 depicts the distribution of the satisfaction ratings given by the participants for the LIME explanations.



On a scale from 0 to 5, how well are you satisfied with the explanations provided to you during the study?

20 responses

Figure 4-8: User satisfaction ratings for LIME study

Figure 4-9 depicts the distribution of the satisfaction ratings given by the participants for the SHAP explanations.

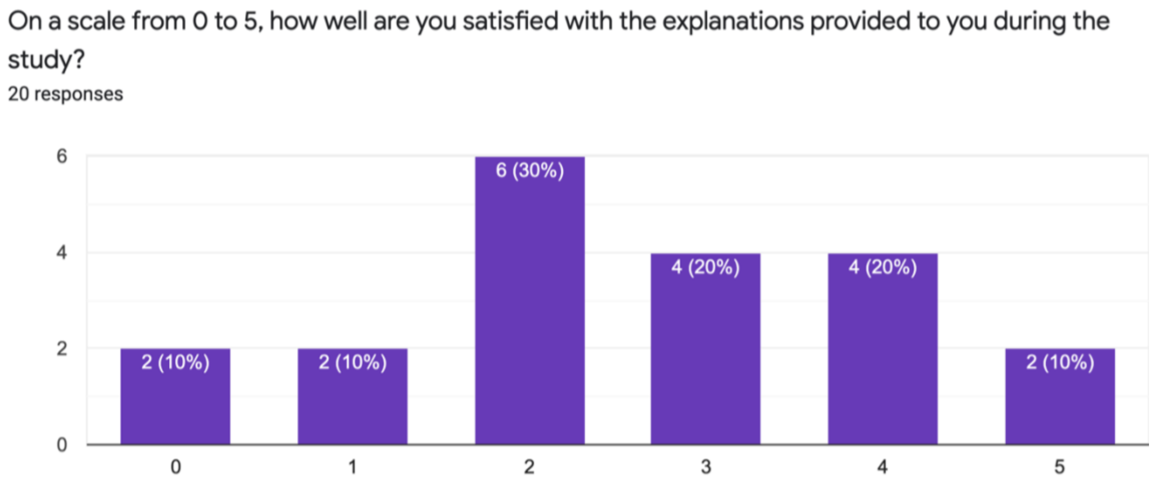


Figure 4-9: **User satisfaction ratings for SHAP study**

4.5.6.2 Trust analysis

The participants of the LIME and SHAP user studies were asked if the explanations were good enough for them to trust the predictions. Figure 4-10 depicts the trust analysis for the LIME explanations. As seen from the figure, 55% of participants felt that the LIME explanations were good enough for them to trust the predictions.

Figure 4-11 depicts the trust analysis for the SHAP explanations. As seen from the figure, 40% of participants felt that the SHAP explanations were good enough for them to trust the predictions.

The results of the trust analysis are consistent with the subjective rating of explanations presented in Table 4.10 and the results of the human evaluation of interpretability presented in table Table 4.4. LIME explanations helped the participants in understanding the predictions for a greater number of cases on average and also had a higher average satisfaction rating as compared to SHAP. Humans have a tendency to trust something more if they understand is better.

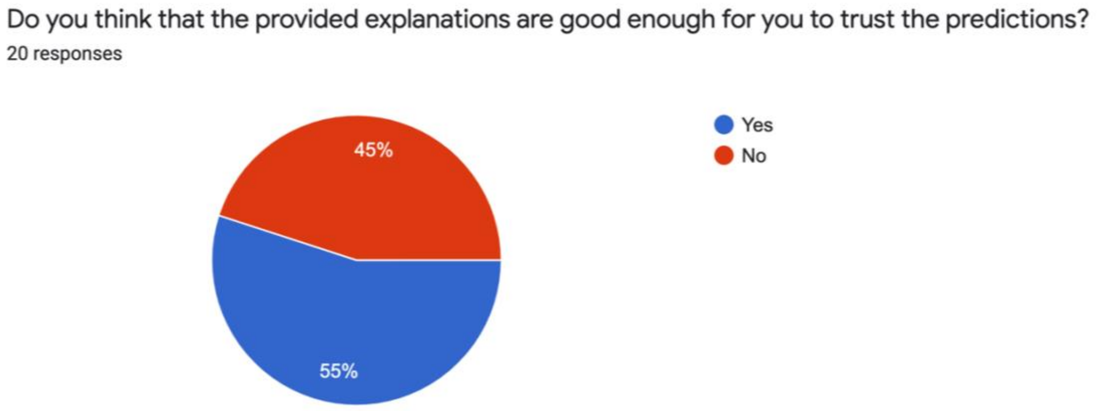


Figure 4-10: **Trust analysis for LIME explanations**

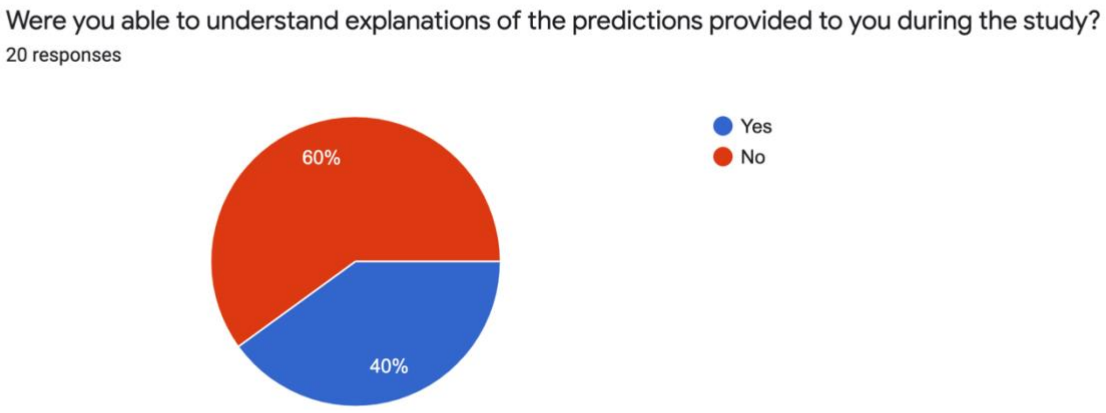


Figure 4-11: Trust analysis for SHAP explanations

The participants of the no XAI user study were asked if the predictions would be more trustable if explanations for the predictions were provided along with them. As depicted in Figure 4-12, 95% of participants answered with a "yes". This clearly validates the need for having explanations for the predictions made by machine learning models, be it using SHAP, LIME, or any other technique.

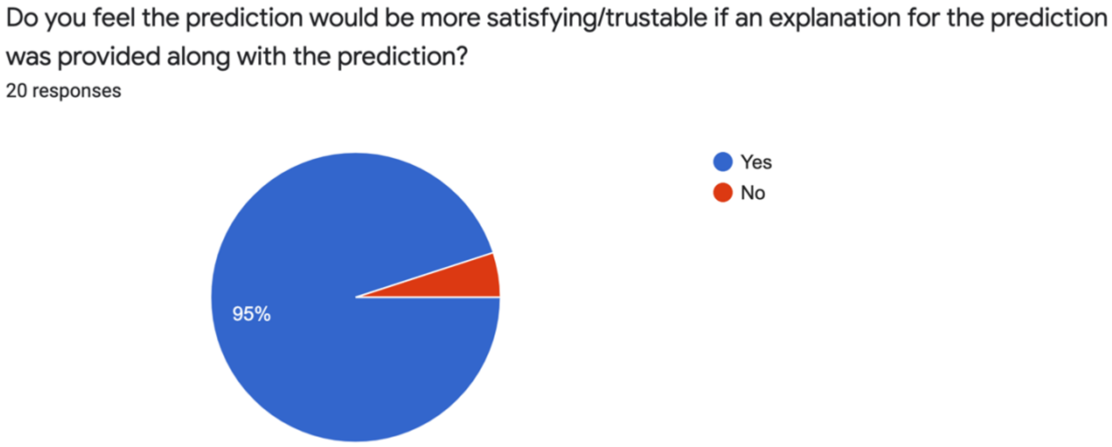


Figure 4-12: Trust analysis in noXAI study

4.5.6.3 User experience analysis

This section presents the user experience analysis of the LIME, SHAP, and noXAI studies. The study participants were asked to rate the user experience of the studies on a Likert scale of 0 to 5. The mean, median, and standard deviation (statistical summary) of the user experience ratings are presented in Table 4.11.

Table 4.11: User experience ratings of the studies

Statistical Measure	LIME	SHAP	noXAI
Mean	**3.60**	**3.05**	**3.95**
Median	4.0	3.50	4.0
Standard deviation	1.53	1.50	0.80

It can be seen from Table 4.11 that LIME received an average user experience rating of 3.60 which is greater than the average user experience rating of SHAP. These results support the results of the human evaluation study (see Table 4.4) and the results of the subjective ratings (see Table 4.10) which are also inclined towards LIME in comparison to SHAP. Logically, the difference in the user experience of the studies was heavily influenced by the LIME, SHAP, and no explanation settings because every other aspect of the studies was kept exactly the same.

The noXAI study received the highest average user experience rating of 3.95. This could possibly be attributed to the fact that the noXAI study did not have any kind of explanation for the predictions. This in turn reduced the complexity of the study and made the user experience better as compared to SHAP and LIME user studies.

Figure 4-13 depicts the distribution of the user experience ratings given by the participants for the LIME study.

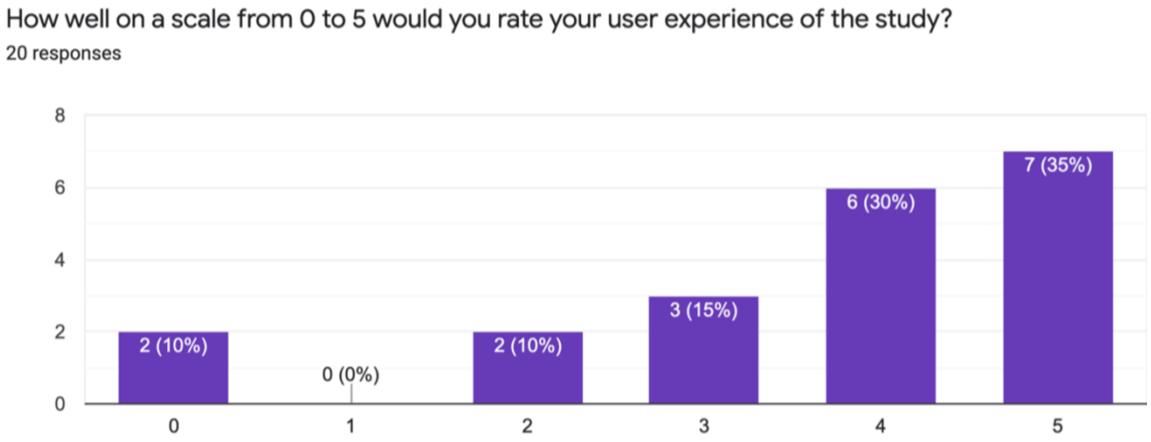


Figure 4-13: User experience ratings for LIME study

Figure 4-14 depicts the distribution of the user experience ratings given by the participants for the SHAP study.

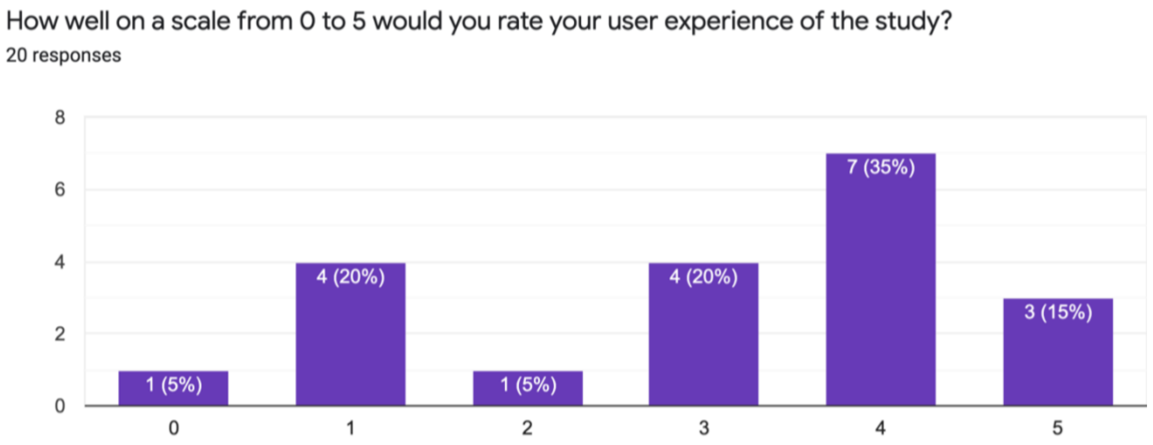


Figure 4-14: User experience ratings for SHAP study

Figure 4-15 depicts the distribution of the user experience ratings given by the participants for the noXAI study.

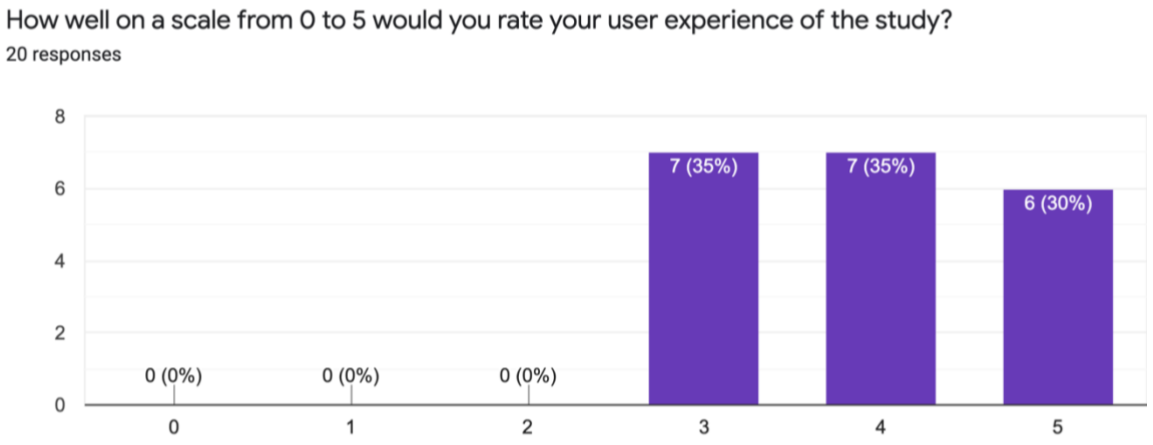


Figure 4-15: User experience ratings for noXAI study

4.5.7 Qualitative analysis of the user feedback

This section discusses the qualitative feedback received from the participants about the explanations and user experience with the LIME, SHAP, and noXAI user studies.

4.5.7.1 Feedback regarding LIME explanations

Some of the study participants felt totally oblivious towards the LIME explanations. Some of the comments made in this respect were: “I did not get the point of this” and “I'm unable to understand the graph so maybe make it better and understandable”. However, a lot of suggestions were given regarding possible improvements that can be made to the explanations. A lot of participants stated that the explanations should be made informative. The improvement suggestions included defining the x and y-axis in a better way, adding a legend, defining the algorithm behind the explanations, and adding informative examples. Some participants stated that the bar charts are probably not the form of visual explanation. Feedback comments like “Bar graphs aren’t the right method for such representation” and “Use pie charts instead” highlight this. One particular participant stated that incorporating global explanations could be helpful as well. The same participant also made a comment about the non-intuitiveness of some elements of the LIME explanation by stating that: “The explanations seemed unintuitive in some cases. For example, why would more meetings in the previous month have a negative contribution to the predicted value? Additional explanations could help with that”.

4.5.7.2 Feedback regarding the user experience of the LIME study

Most of the participants felt that the user experience of the study was good. “It is up to the required level”, “User experience is smooth” and “No improvements required” were some of the comments given by the participants. A few participants found the normalized values used in the study confusing. They mentioned that they would be more comfortable with unnormalized values. One such remark was “Why the data is so confusing like 0.78 or 0.34 use numbers without decimals may be. Make it easier and simpler to understand”. Some participants also recommended adding more information related to the study and its actual used case for a better context. Few more comments like “There was no frame of reference to judge the predictions. Maybe, mention average sales per month so that reader can try to understand why the model predicted high or low in that scenario” and “Input features can be specified better” also emphasized on providing more context about the actual use of such an application. One participant also asked to make the LIME application more interactive by possibly adding hover data access on the bar plots. One participant also commented that perhaps the names of the features can also be improved and made more understandable.

4.5.7.3 Feedback regarding SHAP explanations

Some of the participants indicate in their feedback that they were able to correctly comprehend SHAP values as a measure of the impact of feature values on the model output. For example, “Also, intuitive explanation on SHAP value, if not mathematical, would be helpful although I thought it might be something similar to the coefficient in regression model” and “It was a little hint that SHAP value is correlation with output when the negative values were introduced and the predictions showed a dip”. One participant gave clear feedback on why the SHAP explanation was not understandable by stating that “It isn't clear that the dot (value in shap) is a multiplier/weight/coefficient/correlation factor. So, the natural tendency is to try to match the values from the matrix to the graph.” A major feedback regarding the explanations was about using bars or lines instead of dots to represent the SHAP values. A few participants also recommended color-coding the negative and positive SHAP values in the explanation for increasing understandability. Some participants mentioned in their feedback that explanations can be improved by adding the label to the x-axis and making the label of the y-axis bigger.

4.5.7.4 Feedback regarding the user experience of the SHAP study

Most of the participants felt that the user experience was good. Some comments highlighting this were: “The test is visually simple. There is nothing to distract, and aesthetics are also simple yet engaging” and “no improvements needed. A very popular feedback regarding the user experience was about including examples regarding the task before introducing the actual cases. Some comments made in this regard were: “I don't know if the experiment supports it but an example case (already solved with tips to solve) would have been helpful before solving 10 cases without any prior experience in the Explainable Machine Learning and “provide video examples”. Another popular feedback was that the participants found the normalized feature values difficult to comprehend and perhaps the use of unnormalized values would improve the intuitiveness. Two participants suggested that since the input features are normalized, presenting the output in a normalized form would be good for the sake of consistency. Some other comments regarding the user experience were “Graphs could be larger and more contrasted”, “Make it a bit more visually appealing and “Try to present the 10 cases in a more comprehensive manner, allow for their distinction from one another”. Some participants also said that more information about the term SHAP values, the algorithm, and feature values should be shared in the introduction section. One particular participant said that the design of the applications should be changed so that the participants don’t have to scroll too much and the input features, prediction, and the explanation of one case are visible simultaneously.

4.5.7.5 Feedback regarding explanations in the noXAI study

The participants were asked about what kind of explanations would they find useful. A lot of the participants suggested the explanations should be something visual. Some of the users recommended using some kind of statistical techniques for generating explanations. Whereas some other users suggested explaining the algorithm behind predictions. One user suggested using a linear regression model as it is interpretable.

4.5.7.6 Feedback regarding the user experience of the noXAI study

The majority of the participants said that the user experience was good, and no improvement is needed. Some participants insisted on adding explanations for the predictions to make them more understandable.

5 Conclusions and Future work

This chapter presents the conclusions, limitations, suggestions for future work, and reflections of this project work.

5.1 Conclusions

This work focused on interpreting a machine learning-based time series forecasting model. Another major focus of the work was in evaluating the explanations produced after interpreting the machine learning model.

The first research question posed in this thesis work was**: How to achieve model agnostic interpretability in a time series forecasting problem?**

The answer to the first research question was obtained in the following way. The interpretability (local) of the time series forecasting model was achieved with the help of two of the most popular and reliable model agnostic interpretability techniques, LIME, and SHAP. An important aspect of this research work was to ensure that the temporal dependencies of the time series data are captured in the explanations as no model agnostic interpretability technique is capable of automatically identifying temporal dependencies while producing explanations. Model agnostic interpretability techniques fail to extract the temporal dependencies even if the underlying time series forecasting is built using an advanced algorithm like LSTM which is very popular for capturing the temporal dependencies of time series data during model training. The solution to this research problem was achieved by manually lagging the time series with the help of the sliding window method. The original time series data set was lagged to create 5 different data sets containing lag variables from 1 to 5. The machine learning model trained using the dataset with a lag of 3 was the best performing model with a mean absolute percentage error of 9.29 %. The best performing model was further interpreted using LIME and SHAP. LIME and SHAP were able to successfully capture the temporal dependencies of underlying time series forecasting model after applying the lagging window method. The resulting explanations clearly presented the important features responsible for the prediction along with the time instance (temporal dependency).

The second research question posed in this thesis work was**: How to evaluate the interpretability of a time series forecasting model?**

The answer to the second research question was obtained in the following way. The evaluation of the explanations produced using LIME and SHAP was conducted using the human-grounded evaluation method. Human-grounded evaluation or human evaluation of interpretability involves presenting the explanations to real humans and asking them to perform simple tasks in order to gauge their understanding of the explanations. Their ability to perform these tasks is quantifiably measured using various statistical techniques. The particular kind of human evaluation of interpretability technique found best suited for evaluating the interpretability of a time series forecasting model is called verification technique/task. Until now, no research work has focused on human evaluation of interpretability for time series. Three human evaluation of interpretability studies were designed during this project work. Out of the three studies, the first one focused on LIME explanations, the second one focused on the SHAP explanations and the third one did not contain any explanations and acted as a baseline for the evaluation test. The results clearly proved that the explanations produced by LIME and SHAP greatly helped humans in understanding the predictions made by the machine learning model which clearly aligned with the hypotheses (H_a and H_b) formulated during this project work. The trust analysis also proved that having explanations along with the prediction can massively increase the trust of humans in the predictions made by a machine learning model.

The human evaluation study results also suggested that LIME and SHAP explanations were almost equally understandable with LIME performing better but with a very small margin. This result supported the hypothesis (H_c) formulated during this project work to a little extent. The subjective satisfaction ratings by the participants were higher for LIME. The trust analysis also favored LIME with a higher percentage of people saying that the explanations helped them trust the prediction of the machine learning model. The user experience of the LIME study was rated higher than the SHAP study. All the comparative results between LIME and SHAP are consistent with each other and favored LIME explanations.

The results of demographic correlation analysis show some interesting correlations between the participant's ability to understand the predictions using LIME explanations and their age, gender, and STEM background. Lower understanding of the predictions correlates with higher age. A higher understanding of the prediction correlates with the gender male. A higher understanding of the predictions correlates with the participants having a background in STEM.

The work done during this project can easily be extended to any time series forecasting or classification scenario for achieving and evaluating interpretability. Moreover, this work also forms a good framework for achieving and evaluating interpretability in any machine learning-based regression or classification problem (supervised learning).

5.2 Limitations

The scope of the interpretability was limited to local interpretability and global interpretability was not explored during this project work. However, achieving global interpretability using model agnostic methods is extremely difficult because it is hard for a surrogate model to mimic the full decision boundary of a complex machine learning model, techniques like SHAP do claim that they can produce high-quality global explanations.

Only two model agnostic interpretability techniques, SHAP and, LIME were used during this project. The scope can be extended to other sophisticated feature attribution-based interpretability techniques like CIU [31] and ELI5 [32].

The time spent by the participants on each case was not recorded during the human evaluation studies. The time spent can act as a proxy for the effort made to understand the predictions and the explanations. It can act as a valuable metric of evaluation. Moreover, the complexity of the explanations generated by LIME and SHAP was limited to only the top five most important features. This could be increased to understand the effect of increased complexity on the human understanding of predictions.

Due to constraints of time and resources, the scope of evaluation was limited to human evaluation of interpretability which involves conducting simple experiments with lay humans. The work can be extended to conduct application-grounded evaluation which involves domain experts performing real tasks.

5.3 Future work

The work done in this project can be extended to achieve interpretability using other sophisticated model agnostic interpretability tools like ELI5 and CIU. It would be very interesting to compare the evaluation results of ELI5 and CIU with LIME and SHAP.

During this project work, the explanations were evaluated using human evaluation of interpretability which involves lay humans doing simple tasks. To get a better evaluation of the explanations, they would have to be evaluated using application-grounded evaluation which involves domain experts performing tasks specific to the use of the explanations.

Various control elements of the evaluation studies can be changed for further analysis. One suggestion regarding this is recording the time spent by the participants on each case of the 3 evaluation studies. The time spent could act as a very good proxy for the effort made by the participant to understand the prediction and the underlying explanation. The correlations generated using this could act as a very good evaluation metric and could lead to some interesting insights. Another suggestion is changing the complexity of the explanations. During this work, the size of the explanations was only limited to the top 5 features. It would be interesting to see if increasing the complexity of the explanations like changing the number of features from top 5 to top 10 or 15, affects the ability of the participants to understand the predictions.

Due to paucity of time, the participants for the human evaluation of interpretability study were limited to 60 (20 for each case). It would be logical to validate the results with a larger sample size. Moreover, increasing the sample size could help reach statistically significant results regarding hypothesis H_c (LIME > SHAP).

The type of human evaluation task performed in this project work was a verification task. There are other types of human evaluation tasks like forward prediction and counterfactual prediction which are also unexplored with respect to time series.

The framework used for achieving and evaluating interpretability in this project work focused on a regression setting. However, this work can easily be extended to a classification setting. It would be interesting to compare the human evaluation results of interpretability for a classification setting to those for a regression setting.

References

[1] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608,* 2017. Available: https://arxiv.org/abs/1702.08608

[2] M. T. Ribeiro, S. Singh and C. Guestrin, "Why should I trust you?" Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135-1144. DOI: 10.1145/2939672.2939778

[3] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artificial Intelligence,* vol. 267, pp. 1-38, 2019. DOI: 10.1016/j.artint.2018.07.007

[4] C. Molnar, Interpretable Machine Learning, Lulu. com, 2020. Available: https://christophm.github.io/interpretable-ml-book

[5] P. Esling and C. Agon, "Time-series data mining," *ACM Computing Surveys (CSUR),* vol. 45, pp. 1-34, 2012. DOI: 10.1145/2379776.2379788

[6] J. Brownlee, Introduction to time series forecasting with python: how to prepare data and develop models to predict the future, Machine Learning Mastery, 2017.

[7] B. Hidasi and C. Gaspar-Papanek, "ShiftTree: an interpretable model-based approach for time series classification," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2011, pp. 48-64. DOI: 10.1007/978-3-642-23783-6_4

[8] D. Nguyen, "Comparing automatic and human evaluation of local explanations for text classification," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 1069-1078. DOI: 10.18653/v1/N18-1097

[9] P. Hase and M. Bansal, "Evaluating Explainable AI: Which Algorithmic Explanations Help Users Predict Model Behavior?," *arXiv preprint arXiv:2005.01831,* 2020. Available: https://arxiv.org/abs/2005.01831

[10] H. Yu, L. R. Varshney, H. Taube and J. A. Evans, "Human Evaluation of Interpretability: The Case of AI-Generated Music Knowledge," *arXiv preprint arXiv:2004.06894,* 2020. Available: https://arxiv.org/abs/2004.06894

[11] P. Regulation, "Regulation (EU) 2016/679 of the European Parliament and of the Council," *REGULATION (EU),* vol. 679, p. 2016, 2016. Available: https://dvbi.ru/Portals/0/DOCUMENTS_SHARE/RISK_MANAGEMENT/GDPR_eng_rus.pdf

[12] C. Dearnley, "A reflection on the use of semi-structured interviews," *Nurse researcher,* vol. 13(1), pp. 19-28, 2005. DOI: 10.7748/nr2005.07.13.1.19.c5997

[13] G. Chniti, H. Bakir and H. Zaher, "E-commerce time series forecasting using LSTM neural network and support vector regression," in *Proceedings of the International Conference on Big Data and Internet of Thing*, 2017, pp. 80–84. DOI: 10.1145/2939672.2939778

[14] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in Neural Information Processing Systems,* pp. 4765-4774, 2017. Available: https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf

[15] S.-Y. Shih, F.-K. Sun and H.-y. Lee, "Temporal pattern attention for multivariate time series forecasting," *Machine Learning,* vol. 108, pp. 1421-1441, 2019. DOI: 10.1007/s10994-019-05815-0

[16] S. Gunn, "Support vector machines for classification and regression," *ISIS technical report,* vol. 14, pp. 5-16, 1998. Available: https://see.xidian.edu.cn/faculty/chzheng/bishe/indexfiles/new_folder/svm.pdf

[17] H. Drucker, C. Burges, L. Kaufman, A. Smola and V. Vapnik, "Support vector regression machines," *Advances in neural information processing systems,* vol. 9, pp. 155-161, 1996. Available: https://papers.nips.cc/paper/1996/file/d38901788c533e8286cb6400b40b386d-Paper.pdf

[18] B. Kim, R. Khanna and O. Koyejo , "Examples are not enough, learn to criticize! criticism for interpretability," *Advances in neural information processing systems,* vol. 29, pp. 2280-2288, 2016. Available: https://proceedings.neurips.cc/paper/2016/file/5680522b8e2bb01943234bce7bf84534-Paper.pdf

[19] M. T. Ribeiro, S. Singh and C. Guestrin, "Model-agnostic interpretability of machine learning," *arXiv preprint arXiv:1606.05386,* 2016. Available: https://arxiv.org/pdf/1606.05386

[20] L. Shapley, "A value for n-person games," *Contributions to the Theory of Games,* vol. 2, pp. 307-317, 1953. Available: http://www.library.fa.ru/files/Roth2.pdf

[21] K. E. Mokhtari, B. P. Higdon and Basar Ayse, "Interpreting financial time series with SHAP values," in *Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering*, 2019, pp. 166–172. Available: https://dl.acm.org/doi/abs/10.5555/3370272.3370290

[22] M. Vega García and J. Aznarte, "Shapley additive explanations for NO2 forecasting," *Ecological Informatics,* vol. 56, p. 101039, 2020. DOI: 10.1016/j.ecoinf.2019.101039

[23] Y. Wang, R. Emonet, E. Fromont, S. Malinowski, E. Menager, L. Mosser and R. Tavenard, "Learning Interpretable Shapelets for Time Series Classification through Adversarial Regularization," *arXiv preprint arXiv:1906.00917,* 2019. Available: https://arxiv.org/pdf/1906.00917

[24] A. Malhi, S. Knapic and K. Främling, "Explainable Agents for Less Bias in Human-Agent Decision Making," *International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems,* pp. 129-146, June 2020. Available: https://link.springer.com/chapter/10.1007/978-3-030-51924-7_8

[25] C. Chen, J. Twycross and J. Garibaldi , "A new accuracy measure based on bounded relative error for time series forecasting," *PloS one,* vol. 12, p. e0174202, 2017. DOI: 10.1371/journal.pone.0174202

[26] S. Lundberg, "SHAP python implementation," [Online]. Available: https://github.com/slundberg/shap

[27] M. T. Ribeiro, "LIME gihub implementation," [Online]. Available: https://github.com/marcotcr/lime

[28] T. K. Kim, "T test as a parametric statistic," *Korean journal of anesthesiology,* vol. 68, p. 540, 2015. DOI: 10.4097/kjae.2015.68.6.540

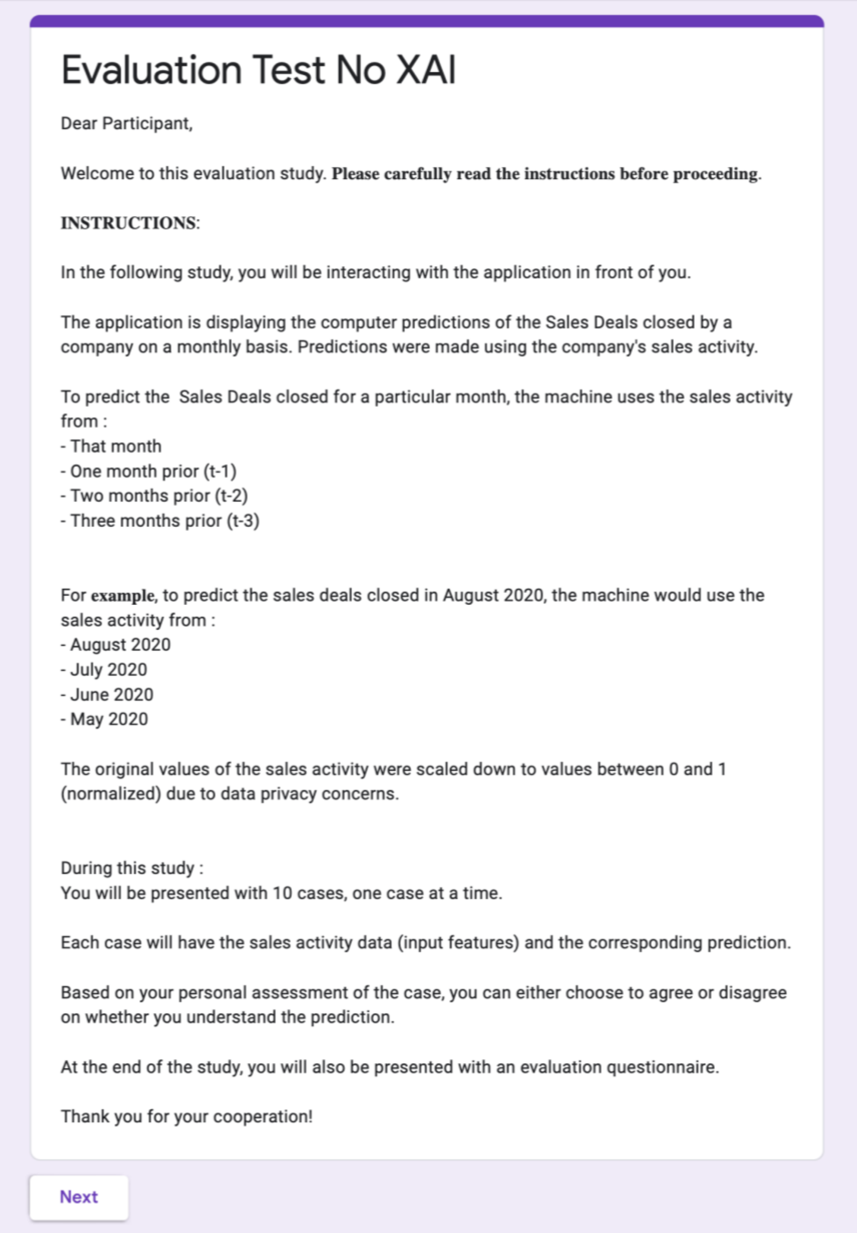
[29] H. J. Hung, R. T. Neill, P. Bauer and K. Köhne, "The behavior of the p-value when the alternative hypothesis is true," *Biometrics,* pp. 11-22, March 1997. DOI: 10.2307/2533093

[30] R. Artusi, P. Verderio and E. Marubini, "Bravais-Pearson and Spearman correlation coefficients: meaning, test of hypothesis and confidence interval," *The International journal of biological markers,* vol. 17, pp. 148-151, 2002. DOI: 10.1177/172460080201700213

[31] K. Främling, "Explaining results of neural networks by contextual importance and utility," in *Proceedings of the AISB'96 conference*, 1996. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.64.7564&rep=rep1&type=pdf

[32] [n.d]., "ELI5 Documentation," 2019. [Online]. Available: https://eli5.readthedocs.io/en/latest/overview.html

Appendix A:



Evaluation Test No XAI

Dear Participant,

Welcome to this evaluation study. **Please carefully read the instructions before proceeding**.

INSTRUCTIONS:

In the following study, you will be interacting with the application in front of you.

The application is displaying the computer predictions of the Sales Deals closed by a company on a monthly basis. Predictions were made using the company's sales activity.

To predict the Sales Deals closed for a particular month, the machine uses the sales activity from :
- That month
- One month prior (t-1)
- Two months prior (t-2)
- Three months prior (t-3)

For **example**, to predict the sales deals closed in August 2020, the machine would use the sales activity from :
- August 2020
- July 2020
- June 2020
- May 2020

The original values of the sales activity were scaled down to values between 0 and 1 (normalized) due to data privacy concerns.

During this study :
You will be presented with 10 cases, one case at a time.

Each case will have the sales activity data (input features) and the corresponding prediction.

Based on your personal assessment of the case, you can either choose to agree or disagree on whether you understand the prediction.

At the end of the study, you will also be presented with an evaluation questionnaire.

Thank you for your cooperation!

Figure A-1: Instructions presented to the human participants in the noXAI application