UNIVERSITAT POLITÈCNICA DE CATALUNYA

BACHELOR'S THESIS

Feature selection in machine learning with Google's PageRank

Author:
Josep MEDIALDEA ROSALES

Tutor:
Enrique ROMERO MERINO

Facultat d'Informàtica de Barcelona
Bachelor's degree in Computer Science
Mention in Computing
Department of Computer Science

June 25, 2021

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Abstract

Facultat d'Informàtica de Barcelona

Department of Computer Science

Bachelor of Computer Science

Feature selection in machine learning with Google's PageRank

by Josep MEDIALDEA ROSALES

Since the development of the famous algorithm behind Google's Search Engine, PageRank has been used in many other fields beyond the web. These fields go from neuroscience to literature.

This thesis elaborated in the Universitat Politècnica de Catalunya wants to port the PageRank algorithm to another field completely different from the web. More specifically, it will develop a feature selection algorithm based on PageRank and study its performance.

Feature selection in machine learning is one of the most relevant problems in the field of Data Science. This project will study the feature selection problem from an unconventional approach: Google's PageRank algorithm.

Acknowledgements

I want to thank Enrique, the director of this thesis. Once I had to choose this thesis topic, I asked him if he wanted to be my director. He is the one who came up with the idea of using the PageRank algorithm for feature selection. Thanks to him, I have learned a lot during the development of this thesis.

I would also like to thank all the incredible teachers and classmates I met during my bachelor's degree at the Universitat Politècnica de Catalunya.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and scope

This chapter introduces the topic of this project and contextualizes it defining the most relevant concepts related to it, describing the problem that the project has to solve, and determining its scope.

1.1 Concept definition

To begin with, we need to define some fundamental concepts needed to understand the project and follow the explanations easily.

1.1.1 Machine Learning

Machine Learning is an application of Artificial Intelligence [1]. Like all the other applications of AI, Machine Learning tries to give computers characteristics of human behavior. In this case, Machine Learning provides systems the ability to automatically learn tasks and improve from experience without being explicitly programmed. The primary aim is to allow computers to learn a specific task without the need for human intervention or assistance and adjust actions accordingly.

To achieve this, the process usually starts with observations or data such as examples to look for patterns in data and make decisions in the future based on the examples provided.

There are two[1] main categories of Machine Learning algorithms:

- **Supervised machine learning algorithms:** this category of algorithms uses labeled examples. This means that they are using information learned in the past. With this information, algorithms generate inferred functions to make predictions about the output data. After sufficient training, the system can provide targets for any new input. Because this category of learning algorithms has already labeled data, they can compare their output with the correct one to modify their models accordingly.
- **Unsupervised machine learning algorithms:** in contrast to supervised learning algorithms, this category of algorithms does not use previously labeled or classified data. Unsupervised learning studies hidden structures from unlabeled data and infer a function to describe it.

This project focuses on supervised machine learning algorithms.

[1] Semi-supervised and Reinforcement algorithms will not be explained.

1.1.2 Dataset

A dataset is a collection of data that is often represented by a matrix where rows represent instances (or observations) and columns represent variables (or properties) of the instances. Variables can be classified into two different groups based on the values that they can take:

- **Numerical variables:** this type of variables take numerical values (e.g., Height).
- **Categorical variables:** this type of variables take a value from a finite set of possible ones (e.g., Gender). Each of the possible values that the variable can take is also known as category or label.

Datasets are used to train Machine Learning models. Supervised machine learning algorithms can also use them to compare their results with the correct ones (i.e., the ones in the dataset).

1.1.3 Classifier

Classifiers are predictive models whose goal is to map input variables (i.e., observations) to discrete output variables (i.e., categorical variables) [2]. For example, we could design a classifier that receives relevant information from a patient (e.g., Age, Gender, Height, Weight, Blood pressure, Heartbeats per minute…) as an input and output if the person is likely to suffer a Heart disease. The output of a classifier is usually known as the target variable.

There are a lot of different Machine Learning algorithms that can be used to generate classification models. Here is a list of some of the most popular ones:

- **Decision Tree**
- **Naive Bayes**
- **Artificial Neural Networks**
- **k-Nearest Neighbors (KNN)**
- **Support Vector Machines (SVM)**

1.1.4 Feature selection

In Machine Learning, feature selection is the process of selecting a subset of variables from a dataset to generate a model [3]. Feature selection can be used for these reasons:

- **Model simplification:** if we reduce the number of features it will be easier for the user to interpret it.
- **Shorter training time:** commonly, the training time of a Machine Learning model grows very fast because of the dimensionality[2] of the dataset. Feature selection is one method of dimensionality reduction that can significantly reduce training time.
- **Overfitting reduction:** feature selection helps reducing overfitting in Machine Learning.

[2] number of variables in the dataset.

Selecting a subset of features from a dataset is not a trivial thing to do. A feature selection method can not select features randomly. Feature selection tries to find redundant or irrelevant features that can be removed from the set of features without incurring much loss of information.

1.1.5 PageRank

PageRank is the famous algorithm behind the Google search engine. It was presented by Larry Page and Sergey Brin back in 1998 in a paper called "The Anatomy of a Large-Scale Hypertextual Web Search Engine" [4]. The purpose of the algorithm was to measure the importance of website pages. According to Google:

> PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites.

Nowadays, PageRank is not the only algorithm used by Google to order search results, but it is the first algorithm that was used by the company, and it is the best known.

The original PageRank algorithm receives an unweighted directed graph as an input and returns the PageRank score of each node. The Google search engine represents the web as a graph where each node is a website page and the edges represent the links between each of them. This graph was used as the input of the PageRank algorithm to calculate the importance of each webpage.

The PageRank score of a node p_i is calculated as follows:

$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)} \tag{1.1}$$

Where N is the number of nodes in the graph, $M(p_i)$ is the set of nodes that link to p_i , $L(p_j)$ is the out-degree of the node p_j and d is called the dumping factor (usually set to 0.85).

There is a generalization of the PageRank algorithm called Weighted PageRank [5], which accepts weighted graphs. This algorithm calculates the score of a node p_i as follows:

$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j) * weight((p_j, p_i))}{L(p_j)} \tag{1.2}$$

Where N is the number of nodes in the graph, $M(p_i)$ is the set of nodes that link to p_i , $L(p_j)$ is the weighted out-degree of the node p_j and d is the dumping factor.

1.2 Problem to be solved

One of the most important things when training a Machine Learning model is the data that is being used. Nowadays, we are surrounded by a large amount of data. Hence, it is often the case where we want to train a model with a dataset containing a significant amount of variables (i.e., features). In this case, probably, using all the features in the dataset is not a good idea. Training a model with a dataset that has a lot of variables increments significantly both the training time of the model and the probability of overfitting. Given the fact that some of the variables could be irrelevant or redundant compared with the other ones, perhaps we could remove them without losing relevant information for the model. This is known as the feature selection problem.

As mentioned before, the feature selection problem consists of selecting a subset of variables from a dataset. This selection takes into consideration the relevance and redundancy of each of the features in the dataset to keep those features that give most of the information relevant for the future model. The goal of this project is to develop a feature selection algorithm based on Google's PageRank.

The PageRank algorithm calculates the importance of each node in a graph. On the web, the algorithm is used to calculate the importance of each website. Hence, the graph used by the Google search engine represents each website page as a node and the edges represent the links between them. Since its presentation back in 1998, PageRank has been used in many different fields beyond the web [6]. These fields go from Biology to Literature.

This project wants to explore the possible modifications needed to use PageRank as a feature selection algorithm. The main idea is to build a graph where the nodes represent the variables of a dataset and use it as an input for PageRank to obtain the score (i.e., the relevance) of each variable.

1.3 State of the Art

In this section, there is a brief explanation of the feature selection algorithms that are currently used.

We can classify feature selection algorithms in four different groups based on their evaluation criteria:

- **Filter methods:** these methods are used during the preprocessing of the data and are independent to the training algorithm used. They usually have low computational complexity. This kind of feature selection algorithms defines a score for each of the variables in the dataset based on their correlation, PCA[3], variance, etc.
- **Wrapper methods:** these methods take into account the accuracy of the model to evaluate the performance of the features. Since every time that the algorithm wants to do an evaluation it needs to train the model, the time complexity is very high.
- **Embedded methods:** these methods are specific to each of the different training algorithms (e.g., Decision Trees).
- **Hybrid methods:** these methods were created as a combination of the filter methods and the wrapper methods.

The feature selection algorithm that will be developed in this project will be a filter method.

[3] Principal Component Analysis

1.4 Stakeholders

This project is intended for researchers in the field of data science and specifically machine learning. It is also addressed to the users of those fields who want to use or study alternative methods of feature selection.

The results of this project could potentially benefit data scientists that need a fast and simple method of feature selection.

1.5 Objectives

The main goal of this project is to develop a feature selection algorithm based on PageRank. This objective will be divided into several steps:

- **Study possible graph representations of a dataset:** design different graph representations of a dataset. All versions will represent the features of the dataset as nodes in the graph but they will differ in the way that they represent the relations (i.e., edges) between each feature (i.e., node).
- **Implement the methods to represent datasets as graphs:** implementation of the code that transforms a dataset into a graph representation that will be the input of the PageRank algorithm.
- **Implement the PageRank algorithm:** once we can transform a dataset into a graph we need to implement the actual PageRank algorithm to calculate how important each feature in the dataset is.
- **Implement an algorithm to evaluate subsets of features:** we need to develop an algorithm capable of evaluating the subset of variables selected.
- **Evaluation of the feature selection algorithm:** we will evaluate the algorithm proposed and compare the results obtained with other feature selection algorithms.

1.6 Requirements

These are the requirements needed to meet the quality standards of this project:

- The algorithm proposed should have a reasonable time complexity.
- The code must be easy to understand to be used by other users.
- Modular implementation of the software. This will make upgrades and modifications accessible.
- Limited use of external libraries.
- The feature selection method should perform coherently. The subset of features selected can not be random and must follow any of the characteristics that we expect from a good feature selection.

1.7 Possible requirements and obstacles

Here there is a list of the possible obstacles and risks that may occur during the realization of this project:

- **Disease or injury:** usually this is an uncommon event but given the fact that this project is done during the COVID-19 pandemic it must be taken into consideration.
- **Errors during the design of the algorithm:** they can cause unexpected behaviors. Since the design of the algorithm is one of the first steps of the project this type of error can be severe.
- **Errors during the implementation of the algorithm:** they may appear during the code implementation of the algorithm.
- **Errors during the evaluation of the obtained results:** this kind of error is very important because it can potentially lead the project to wrong conclusions.
- **Hardware failure:** they are unpredictable but their consequences can be mitigated easily if the project is sufficiently backed up.

1.8 Methodology

This section defines the methodology and the tools that will be used during the project.

1.8.1 Working methodology

The working methodology that we chose was Scrum [7]. Scrum is an *agile* methodology that breaks the work into goals that can be completed within time-boxed iterations (called *sprints*) no longer than one month and most commonly two weeks. At the end of the *sprint*, the team meets to demonstrate the work done and decide the work that will be done during the next sprint.

We will apply this methodology by meeting every two weeks via Google Meet. We chose this method because it allows us to stay in touch frequently to clarify doubts and to make sure that both the author and the tutor are on the same page.

1.8.2 Development tools

These are the tools that will be used during the project:

- **Programming language:** the chosen programming language will be Python. The decision was easy because at the moment it is the most used programming language when it comes to data science.
- **External libraries:** the main external library that will be used is Pandas. Pandas is a Python library used to interact with datasets. It has methods to read and write the dataset and to make queries.
- **Document preparation:** this document will be elaborated using the LaTeX software system. LaTeX is widely used in academia for the communication and publication of scientific documents in many fields including mathematics, computer science, engineering...

Chapter 2

PageRank Feature Selection algorithm

This chapter is the most theoretically important of the whole thesis since it will detail the PageRank-based feature selection algorithm around which the whole thesis is based.

First of all, we will explain the logic we have followed to adapt the performance of the PageRank algorithm to the attribute selection problem. For this purpose, we will also provide pseudocode that summarizes in a general way the algorithm operation.

Secondly, once we have defined the algorithm in a general way, we will list and explain the algorithm's parameters, which allows the user to tune the algorithm's operation according to his needs.

Finally, we will describe the different configurations of the algorithm (combinations of parameters) that we will test during the experimentation phase of the thesis.

2.1 Algorithm description

As we have already mentioned, we will start by describing the PageRank-based feature selection algorithm, which we will refer to as PRFS from now on.

First, let us briefly recall the problem we want to solve: selecting features in a dataset used to train a predictive model. The problem is easy to define formally:

The input to the problem will be a dataset D that, in this thesis, we will interpret as a matrix of size $ns \times (nf + 1)$ where ns represents the number of samples in the dataset and nf represents the number of features. The extra column in the dataset is the target variable to be predicted using predictive models. Since the feature selection algorithm we propose is a filter method, the problem will consist of assigning a score to each of the attributes in the dataset so that we can subsequently produce a ranking. The ideal objective is that the ranking produced by the algorithm places the most "important" attributes in higher positions (later, we will define what we mean by important) so that the user can keep only the attributes that he/she considers necessary.

Having defined the problem we are facing more formally, let us recall what the PageRank algorithm consists of and how it can solve the attribute selection problem. The PageRank algorithm can be interpreted as a graph centrality algorithm. The algorithm receives a graph as input and assigns a score to every one of its nodes.

With all this in mind, it is possible to intuit the functioning of the algorithm we have designed. The algorithm has two distinct phases: first, it will build a graph representing the input dataset in which each of the features will be represented by a node, and second, once the graph has been constructed, it will calculate the score of each of the nodes using the PageRank algorithm and will order the features based on the score that the PageRank algorithm has assigned to their nodes.

We can consider another version of the algorithm that, apart from receiving the dataset as input, also receives an integer n representing the number of features we want to select. Instead of returning the ranking of all the features, this version will return a set with the first n features.

Here there is the pseudocode of the last version of the algorithm:

Algorithm 1: PRFS algorithm

```
Input:
$D$ - Dataset
$n$ - Number of features that the user wants to select
$params$ - Parameters of the algorithm
Output:
$FS$ - Subset of $n$ features

$G := DatasetGraphRepresentation(D, params)$;
$PR := CalculatePageRankScores(G)$;
$F := GetFeaturesOrderedByPageRankScore(D,PR)$;
$i := 0$;
$FS := []$;
while $i < n$ do
    $FS.append(F[i])$;
    $++i$;
end
```

We can observe in the pseudocode that the algorithm also receives the user's parameters. Therefore, in the following subsection, we will detail the parameters of the algorithm and their functionality.

2.2 Parameters

As seen in the proposed pseudocode, the algorithm receives some parameters used to create the graph. The parameters determine how the graph that will represent the input dataset is constructed.

Before listing and defining in detail each one of the parameters, let us stop and think about what characteristics the graph representing the dataset should have so that, using the logic of the PageRank algorithm, the scores that the algorithm assigns to the feature nodes meet the objective we want to achieve.

To do so, let us analyze how the graph used by PageRank is constructed when used by search engines. In search engines, PageRank is used to score every web page that is available on the Internet. The algorithm represents the web as a graph where the nodes represent web pages, and the edges represent links between them. Thus, we can visualize the graph as if pages were transmitting scores between them.

We have transferred this logic to the feature selection problem. The graph with which we will represent the data set will represent every feature in the form of nodes that transmit scores to each other. Thus, the algorithm will have to do the following:

- **Reward relevance:** each feature will transmit a better score to those features that are relevant. It will give more points to those features that have information related to the target variable.
- **Penalize redundancy:** each feature will give less score to those features that contain redundant information. Likewise, it will give less score to those features with information already contained in other features.

Features will penalize and reward the aspects we have just mentioned by assigning a weight to the edges that connect them with the rest of the features.

Having explained the logic that the graph representing the input dataset must follow, we will mention the parameters that will define its shape. The parameters are the following:

- **Graph type (or model):** parameter that determines how the graph's adjacency matrix will be constructed and what shape the graph will have.
- **Alpha function** (α)**:** the function that will be used to calculate the metric in charge of representing the relevance of the features. Because the relevance of a feature depends on the information it shares with the target variable, the alpha function will calculate the relationships between the features and the target variable.
- **Beta function** (β)**:** the function that will be used to calculate the metric in charge of representing the non-redundancy of the features. As the redundancy of a feature depends on the information it shares with the other features, the beta function will be used to calculate the relationship between the features.
- **Weight** (w)**:** a real number that will give more importance to the relevance or non-redundancy. It will weigh the values calculated by the alpha function and those of the beta function.

Having listed the parameters, we will explain in detail each of them separately in the following sections.

2.2.1 Graph models

We have designed two different graph models that the user can use to represent the input dataset.

Feature graph

The first graph model is called the feature graph. The feature graph model is a complete weighted directed graph with the same number of nodes as features in the dataset. Each node represents a feature, and it is connected to the rest of the nodes. The weight of every edge in the graph is defined as follows:

Given $G = (V, E)$ the feature graph of a dataset D with n samples represented as an $n \times m$ matrix with columns $\{f_1, ..., f_{m-1}, t\}$ where $\forall i \in \{1, ..., m-1\},\ f_i$ represents a feature and t represents the target variable that the user wants to predict, $\forall f_i f_j \in E,\ weight(f_i f_j) = \alpha(f_j, t) + w * \beta(f_i, f_j)$.

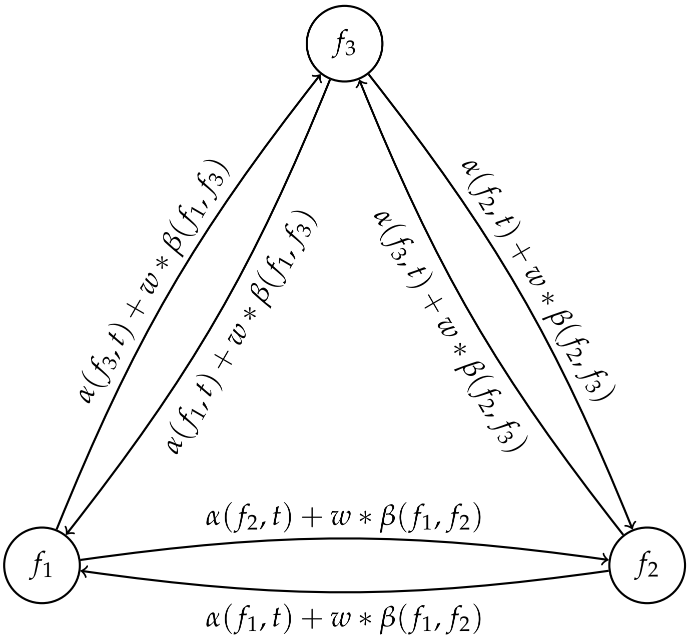

f_3
f_1
f_2
$\alpha(f_3,t)+w*\beta(f_1,f_3)$
$\alpha(f_1,t)+w*\beta(f_1,f_3)$
$\alpha(f_3,t)+w*\beta(f_2,f_3)$
$\alpha(f_2,t)+w*\beta(f_2,f_3)$
$\alpha(f_2,t)+w*\beta(f_1,f_2)$
$\alpha(f_1,t)+w*\beta(f_1,f_2)$

FIGURE 2.1: Feature graph of a dataset with three features.

In this graph model $\forall f_i \in V$ the PageRank score of f_i is defined as follows:

$$PR(f_i) = \frac{1-d}{|V|} + d \sum_{f_j \neq f_j} \frac{(\alpha(f_i,t) + w * \beta(f_i,f_j)) * PR(f_j)}{\sum_{f_j \neq f_j} \alpha(f_i,t) + w * \beta(f_i,f_j)} \tag{2.1}$$

Feature+Target graph

The second graph model is called the feature+target graph. The feature+target graph model is a complete weighted directed graph with the same number of nodes as features in the dataset and one additional node representing the target variable. The weight of every edge in the graph is defined as follows:

Given $G = (V, E)$ the feature+target graph of a dataset D with n samples represented as an $n \times m$ matrix with columns $\{f_1, ..., f_{m-1}, t\}$ where $\forall i \in \{1, ..., m-1\}$, f_i represents a feature and t represents the target variable that the user wants to predict,

- $\forall f_i f_j \in E,\ weight(f_i f_j) = w * \beta(f_i, f_j).$
- $\forall f_i t \in E,\ weight(f_i t) = 1.$
- $\forall t f_i \in E,\ weight(t f_i) = \alpha(f_i, t).$

In this graph model $\forall f_i \in V$ the PageRank score of f_i is defined as follows:

$$PR(f_i) = \frac{1-d}{|V|} + d \sum_{f_j \neq f_j} \frac{w * \beta(f_i,f_j) * PR(f_j)}{\sum_{f_j \neq f_j} w * \beta(f_i,f_j)} + \frac{\alpha(f_i,t) * PR(t)}{\sum_{f_j} \alpha(f_j,t)} \tag{2.2}$$

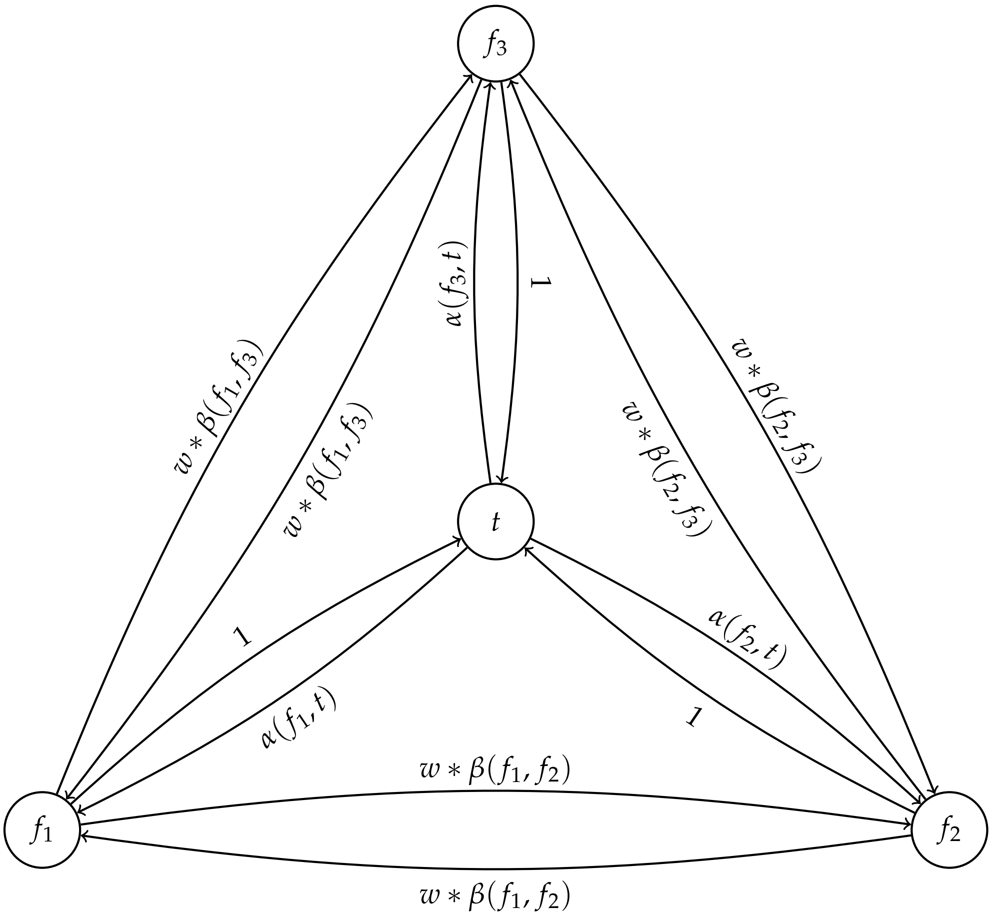


FIGURE 2.2: Feature+Target graph of a dataset with three features.

2.2.2 Alpha functions

The alpha function parameter of the PRFS algorithm is used to calculate the metric that evaluates the relationship between the features and the target variable (i.e., the relevance of each feature). The parameter can be any function $f\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ that receives the values of one feature and the values of the target variable as parameters and returns a real number. The return value of the alpha function should be directly proportional to the relevance of the feature. Here there is a list of the alpha functions that we propose as parameters for the PRFS algorithm:

Correlation

The first alpha function that we thought about is the absolute value of the Pearson Correlation coefficient. It is defined as follows:

$$correlation(f_i, t) = \left| \frac{\text{Cov}(f_i, t)}{\text{Var}(f_i) * \text{Var}(t)} \right| \tag{2.3}$$

Pearson correlation measures the linear relationship between two continuous variables and has a value between 1 and -1. In other words, the Pearson Correlation Coefficient measures the relationship between 2 variables via a line [8]. When the correlation coefficient is closer to value 1, it means there is a positive relationship between the variables. A positive relationship indicates an increase in one variable associated with an increase in the other. On the other hand, the closer correlation coefficient is to -1 would mean there is a negative relationship which is that the increase in one variable would result in a decrease in the other. If the variables are independent, then the correlation coefficient is close to 0, although the Pearson correlation can be small even if there is a strong relationship between the two variables. We take the absolute value of the Pearson Correlation because we are only interested in the amount of correlation between the feature and the target variable.

Spearman Correlation

The second alpha function is very similar to the first one. In fact, the only thing that changes is the correlation coefficient. This version uses the Spearman correlation coefficient instead of the Pearson's [9] [10]:

$$spearman\_correlation(f_i, t) = \left| \frac{\sum_r (f_{i_r} - \overline{f_i})(t_r - \overline{t})}{\sqrt{\sum_r (f_{i_r} - \overline{f_i})^2 \sum_r (t_r - \overline{t})^2}} \right| \tag{2.4}$$

Where r is the rank of each of the ordered values of f_i and t. The fundamental difference between the two correlation coefficients is that the Pearson coefficient works with a linear relationship between the two variables. In contrast, the Spearman Coefficient works with monotonic relationships as well. A monotonic relationship is a relationship that does one of the following:

1. As the value of one variable increases, so does the value of the other variable.
2. As the value of one variable increases, the other variable value decreases.

However, not exactly at a constant rate, whereas in a linear relationship, the rate of increase/decrease is constant.

Mutual Information

The Mutual Information [11] between two random variables measures non-linear relations between them. Besides, it indicates how much information can be obtained from a random variable by observing another random variable.

It is closely linked to the concept of entropy. This is because it can also be known as the reduction of uncertainty of a random variable if another is known. Therefore, a high mutual information value indicates a large reduction of uncertainty, whereas a low value indicates a small reduction. If the mutual information is zero, that means that the two random variables are independent.

The following formula shows the calculation of the mutual information for two discrete random variables:

$$mutual\_information(f_i, t) = \sum_{y \in t} \sum_{x \in f_i} p_{f_i,t}(x, y) * log \left(\frac{p_{f_i,t}(x, y)}{p_{f_i}(x) p_t(y)} \right) \tag{2.5}$$

Where p_{f_i} and p_t are the marginal probability density functions and $p_{f_i,t}$ the joint probability density function.

As explained before, it is related to entropy. This relation is shown in the following formula:

$$mutual\_information(f_i, t) = H(f_i, t) - H(f_i|t) - H(t|f_i) \tag{2.6}$$

Entropy H measures the level of expected uncertainty in a random variable. Therefore, $H(f_i)$ is approximately how much information can be learned of the random variable f_i by observing just one sample.

$$H(f_i) = -\sum_{x \in f_i} P(f_i = x) * log(P(f_i = x)) \tag{2.7}$$

The joint entropy measures the uncertainty when considering together two random variables.

$$H(f_i, t) = -\sum_{x \in f_i} \sum_{y \in t} P(f_i = x, t = y) * log(P(f_i = x), P(t = y)) \tag{2.8}$$

The conditional entropy measures how much uncertainty has the random variable f_i when the value of t is known.

$$H(f_i, t) = -\sum_{x,y \in f_i,t} P(x, y) * log(P(x|y)) \tag{2.9}$$

Accuracy

This last alpha function is a little bit different than the usual alpha functions. It is as simple as calculating the accuracy of a Naive Bayes classifier using only one feature.

$$accuracy(f_i, t) = NBscore(f_i, t) \tag{2.10}$$

When this alpha function is used, the algorithm can be considered a wrapper feature selection algorithm because it uses a predictive model to evaluate features. We will use the Naive Bayes classifier, but we could use any other classifier. We will use this classifier because it is simple (has very few parameters) and fast. The logic behind this alpha function is that the most relevant features should get higher accuracy.

2.2.3 Beta functions

The beta function parameter of the PRFS algorithm is used to calculate the metric that evaluates the relationship between each pair of features (i.e., the redundancy of each feature). The parameter can be any function $f \colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ that receives the values of two features as parameters and returns a real number. The return value of the beta function should be directly proportional to the non-redundancy of the features between them. Here there is a list of the alpha functions that we propose as parameters for the PRFS algorithm:

Uncorrelation

The uncorrelation beta function is the inverse of the correlation alpha function between the pair of features.

$$uncorrelation(f_i, f_j) = \left| \frac{\mathrm{Cov}(f_i, f_j)}{\mathrm{Var}(f_i) * \mathrm{Var}(f_j)} \right| \tag{2.11}$$

This function is the same as the correlation alpha function but inverted. This way, the function will return higher values (i.e., closer to 1) when the features are not linearly correlated and return lower values (closer to 0) when the features are linearly correlated.

Spearman Uncorrelation

The spearman uncorrelation beta function is the inverse of the spearman correlation alpha function between the pair of features.

$$spearman\_uncorrelation(f_i, f_j) = \left| \frac{\sum_r (f_{i_r} - \overline{f_i})(f_{j_r} - \overline{f_j})}{\sqrt{\sum_r (f_{i_r} - \overline{f_i})^2 \sum_r (f_{j_r} - \overline{f_j})^2}} \right| \tag{2.12}$$

This function is the same as the spearman correlation alpha function but inverted. This way, the function will return higher values (i.e., closer to 1) when the features have a low Spearman correlation coefficient and return lower values (closer to 0) when the features have a high Spearman correlation coefficient.

Accuracy

This beta function is very similar to the accuracy alpha function. In this case, the beta accuracy function returns the difference between using only one feature and both.

$$accuracy(f_i, f_j) = max(NBScore(\{f_i, f_j\}, \{t\}) - NBScore(\{f_i\}, \{t\}), 0) \tag{2.13}$$

The logic behind this alpha function is that less redundant features should add more accuracy to the predictive models obtained using both features. In this case, the Naive Bayes is very convenient because the model assumes independence between variables, and those variables that are dependent (i.e., redundant) obtain worse results.

2.3 Parameter configurations

This section will define the parameter configurations that we will use during the experimentation. Here we can see a list of the combination of graph models, alpha functions, and beta functions chosen:

- Feature graph, correlation and uncorrelation.
- Feature graph, spearman_correlation and spearman_uncorrelation.

- Feature+Target graph, correlation, uncorrelation.
- Feature+Target graph, spearman_correlation, spearman_uncorrelation.
- Feature+Target graph, mutual_information, uncorrelation.
- Feature+Target graph, accuracy, accuracy.

When it comes to the weight functions, we will experiment with values 0.2, 0.5, and 0.8. These values have only an exploratory purpose of seeing the effects of the weight parameter.

Chapter 3

Datasets

In this chapter, we will explain the datasets used during the experimentation of the Page Rank Feature Selection algorithm developed in this thesis. We will divide the datasets into two distinct groups: synthetic datasets and real datasets.

3.1 Synthetic datasets

The first group of datasets that we are going to describe is synthetic datasets. Synthetic datasets will be created by ourselves with artificial data. They allow us to easily interpret the experimentation results with the algorithm because we already know useful information about the dataset beforehand. For each synthetic dataset class, we are going to produce different versions. Each version will be created adding/removing redundant and irrelevant variables to/from the original dataset.

3.1.1 Dice dataset

The first synthetic dataset that we created is called the Dice dataset. Each sample in the dataset contains four relevant variables representing four dice.

Variable name	**Domain**
dice1	$\{1,2,3,4,5,6\}$
dice2	$\{1,2,3,4,5,6\}$
dice3	$\{1,2,3,4,5,6\}$
dice4	$\{1,2,3,4,5,6\}$
target	$\{0,1,2,3\}$

The target variable is calculated as follows:

$$target(dice1, dice2, dice3, dice4) = \begin{cases} 3, & \text{if } dice1 + dice2 + dice3 + dice4 \geq 17 \\ 2, & \text{if } dice1 + dice2 + dice3 + dice4 \geq 14 \\ 1, & \text{if } dice1 + dice2 + dice3 + dice4 \geq 11 \\ 0, & \text{otherwise} \end{cases}$$

The dataset is very easy to create because we only need to generate three random integers between one and six and calculate the target variable for each sample. All versions of the Dice dataset will contain 2000 samples.

Version 1

The version adds one redundant variable and four irrelevant variables to the original dataset.

Variable name	Variable type	Domain
dice1	Relevant variable	$\{1,2,3,4,5,6\}$
dice2	Relevant variable	$\{1,2,3,4,5,6\}$
dice3	Relevant variable	$\{1,2,3,4,5,6\}$
dice4	Relevant variable	$\{1,2,3,4,5,6\}$
d1_d2_sum	Redundant variable	$\{4,...,24\}$
i1	Irrelevant variable	$\{1,2,3,4,5,6\}$
i2	Irrelevant variable	$\{1,2,3,4,5,6\}$
i3	Irrelevant variable	$\{1,2,3,4,5,6\}$
i4	Irrelevant variable	$\{1,2,3,4,5,6\}$
target	Target variable	$\{0,1,2,3\}$

Redundant variable d1_d2_sum is calculated as follows:

$$d1\_d2\_sum = dice1 + dice2$$

We create all irrelevant variables in this version and the following versions of the Dice dataset the same way we create the relevant variables: generating a random number between one and six. The only important difference is that we do not consider the value of the irrelevant variables when determining the target variable.

Version 2

The version adds two redundant variables and eight irrelevant variables to the original dataset.

Variable name	Variable type	Domain
dice1	Relevant variable	$\{1,2,3,4,5,6\}$
dice2	Relevant variable	$\{1,2,3,4,5,6\}$
dice3	Relevant variable	$\{1,2,3,4,5,6\}$
dice4	Relevant variable	$\{1,2,3,4,5,6\}$
d1_d2_sum	Redundant variable	$\{4,...,24\}$
d3_d4_sum	Redundant variable	$\{4,...,24\}$
i1	Irrelevant variable	$\{1,2,3,4,5,6\}$
i2	Irrelevant variable	$\{1,2,3,4,5,6\}$
i3	Irrelevant variable	$\{1,2,3,4,5,6\}$
i4	Irrelevant variable	$\{1,2,3,4,5,6\}$
i5	Irrelevant variable	$\{1,2,3,4,5,6\}$
i6	Irrelevant variable	$\{1,2,3,4,5,6\}$
i7	Irrelevant variable	$\{1,2,3,4,5,6\}$
i8	Irrelevant variable	$\{1,2,3,4,5,6\}$
target	Target variable	$\{0,1,2,3\}$

Redundant variables d1_d2_sum and d3_d4_sum are calculated as follows:

$$d1\_d2\_sum = dice1 + dice2$$

$$d3\_d4\_sum = dice3 + dice4$$

Version 3

The version adds two redundant variables (the same as version 2) and forty irrelevant variables to the original dataset.

Variable name	Variable type	Domain
dice1	Relevant variable	$\{1,2,3,4,5,6\}$
dice2	Relevant variable	$\{1,2,3,4,5,6\}$
dice3	Relevant variable	$\{1,2,3,4,5,6\}$
dice4	Relevant variable	$\{1,2,3,4,5,6\}$
d1_d2_sum	Redundant variable	$\{4,...,24\}$
d3_d4_sum	Redundant variable	$\{4,...,24\}$
i1...i40	Irrelevant variables	$\{1,2,3,4,5,6\}$
target	Target variable	$\{0,1,2,3\}$

3.1.2 Sklearn dataset

The second synthetic dataset that we have created is called the Sklearn dataset. We can generate this dataset using the $make\_classification$ function included in the Scikit Learn Python library [12]. This function initially creates clusters of points normally distributed about vertices of an n_informative-dimensional hypercube and assigns an equal number of clusters to each class. It introduces interdependence between these features and adds various types of further noise to the data. These are some of the most important parameters that the $make\_classification$ function receives:

- **n_samples:** The number of samples.
- **n_features:** The total number of features. These comprise n_informative informative features, n_redundant redundant features, n_repeated duplicated features and n_features-n_informative-n_redundant-n_repeated useless features drawn at random.
- **n_informative**: The number of informative features. Each class is composed of a number of gaussian clusters each located around the vertices of a hypercube in a subspace of dimension n_informative. For each cluster, informative features are drawn independently from N(0, 1) and then randomly linearly combined within each cluster in order to add covariance. The clusters are then placed on the vertices of the hypercube.
- **n_redundant:** The number of redundant features. These features are generated as random linear combinations of the informative features.
- **n_classes:** The number of classes (or labels) of the classification problem.

Version 1

This version of the Sklearn dataset will be generated with the following parameters:

- **n_samples:** 2000.
- **n_features:** 15.
- **n_informative**: 5.
- **n_redundant:** 5.

- **n_classes:** 2.

Variable name	Variable type	Domain
relevant1..relevant5	Relevant variable	$\mathbb{R}$
redundant1..redundant5	Redundant variable	$\mathbb{R}$
irrelevant1..irrelevant5	Irrelevant variable	$\mathbb{R}$
target	Target Variable	$\{0,1\}$

Version 2

This version of the Sklearn dataset will be generated with the following parameters:

- **n_samples:** 2000.
- **n_features:** 20.
- **n_informative**: 5.
- **n_redundant:** 5.
- **n_classes:** 2.

Variable name	Variable type	Domain
relevant1..relevant5	Relevant variable	$\mathbb{R}$
redundant1..redundant5	Redundant variable	$\mathbb{R}$
irrelevant1..irrelevant10	Irrelevant variable	$\mathbb{R}$
target	Target Variable	$\{0,1\}$

Version 3

This version of the Sklearn dataset will be generated with the following parameters:

- **n_samples:** 2000.
- **n_features:** 50.
- **n_informative**: 5.
- **n_redundant:** 5.
- **n_classes:** 2.

Variable name	Variable type	Domain
relevant1..relevant5	Relevant variable	$\mathbb{R}$
redundant1..redundant5	Redundant variable	$\mathbb{R}$
irrelevant1..irrelevant40	Irrelevant variable	$\mathbb{R}$
target	Target Variable	$\{0,1\}$

3.2 Real datasets

3.2.1 Heart Disease UCI dataset

This dataset contains information from 300 UCI patients such as their age, resting blood pressure, sex... The target variable refers to the presence of heart disease in the patient.

The dataset was developed by the Cleveland Clinic Foundation, the Zurich University Hospital, the Basel University Hospital, and the Hungarian Institute of Cardiology.

3.2.2 Titanic dataset

This dataset consists of the information of the 2224 passengers and crew from the famous RMS Titanic.

For each passenger, we have much information such as their age, gender, socioeconomic class... The target variable represents whether the passenger survived or not.

Chapter 4

Algorithm evaluation

Once we have defined the Page Rank Feature Selection algorithm and presented the datasets that we selected, we will describe the methods that we used to evaluate the algorithm. The algorithm evaluation process can be divided into four steps:

1. **Experimentation**
2. **Data extraction**
3. **Graph generation**
4. **Conclusions**

This chapter will describe steps 1 to 3, and we will dedicate a whole chapter to the conclusions later.

4.1 Experimentation

In this section, we will explain the first step of the algorithm evaluation process. We suppose that the final version of the PRFS algorithm is already implemented. The experimentation consists of executing the feature selection algorithm with the selected datasets using different parameter configurations. As the result of the experimentation, we will obtain the feature ranking for every combination of datasets and parameter configurations.

4.2 Data extraction

This section describes the data that we extract after the experimentation. After the execution of the feature selection algorithm, we obtain a ranking of the features. We thought about how we could evaluate if a given ranking of features was good or not. We know that a good ranking keeps the best features (relevant) in higher positions and the worst (irrelevant) features in the lower positions. Considering this, we thought that we could evaluate the ranking by evaluating the incremental subsets of features that we obtain if we start by selecting the first feature in the ranking, then the two best features, continuing until we finish with all the features in the ranking selected. For example, if we execute the PRFS on a dataset $D = \{f_1, f_2, f_3, f_4, f_5, t\}$ and we obtain the ranking $[f_2, f_5, f_3, f_4, f_1]$ the incremental subsets of features would be $\{\{f_2\}, \{f_2, f_5\}, \{f_2, f_5, f_3\}, \{f_2, f_5, f_3, f_4\}, \{f_2, f_5, f_3, f_4, f_1\}\}$. For every set we calculate the following data:

- **Relevant variables:** Number of relevant variables in the subset.

- **Redundant variables:** Number of redundant variables in the subset.
- **Irrelevant variables:** Number of irrelevant variables in the subset.
- **Naive Bayes accuracy:** Accuracy of a Naive Bayes model trained using the features in the subset.
- **Decision Tree accuracy:** Accuracy of a Decision Tree model trained using the features in the subset.
- **Support Vector Classifier accuracy:** Accuracy of a Support Vector Classifier model trained using the features in the subset.

It is important to note that the classification of variables into relevant, redundant, and irrelevant will only be done in the synthetic dataset. Because of that, we will not calculate the number of relevant, redundant, and irrelevant variables selected in the subsets of the subsets features of real datasets. It is also important to note that we will not spend time selecting the best parameters of the classifier models because our goal is not to solve the classification problem. We only want to compare the accuracy obtained using every subset of features to evaluate how the accuracy evolves when we incrementally add features from the ranking. This is why we used the default parameters of each model defined in the SciKit Learn library.

4.3 Graph generation

In the previous section, we have defined the data extraction process. This section will define how we are going to represent the extracted data graphically.

For each combination of dataset and parameter configuration, we will generate two graphs:

- **Accuracy graph:** This graph will illustrate how the accuracy evolves when we increase the number of features selected. The horizontal axis will represent the number of features selected, and the vertical axis will represent the accuracy obtained. There will be three plots for each figure, one for every different classifier.
- **Feature type graph:** This graph will illustrate how the feature type count evolves when we increase the number of features selected. The horizontal axis will represent the number of features selected, and the vertical axis will represent the type count. There will be three plots for each figure, one for every different type of feature.

All graphs will be generated using the famous matplotlib library. They can be found in the results chapter.

Chapter 5

Conclusions

This chapter summarizes the conclusions that we have extracted from the results of the experimentation in this thesis.

Feature selection is a very modern field of study that is growing very fast. This thesis has only covered a very small fraction of the field of feature selection. Even though we have experienced how necessary it is to keep working on improving the feature selection techniques because, as we can observe in the results obtained, all datasets contain variables that add noise to the predictive models that worsen the results obtained. Applying feature selection techniques to the datasets is very important because:

- The predictive models obtained will obtain better results because we reduce overfitting.
- We will reduce the size of the training dataset and, consequently, the storing requirements.
- We can visualize patterns in the dataset easily because we remove some of the irrelevant data.
- We reduce both the training and the prediction time of the models and all resources related to them.

This thesis has introduced a new feature selection algorithm: the PageRank Feature Selection algorithm. We have put much effort into trying to adapt the PageRank algorithm to the field of feature selection. Our goal has not been easy because the web (the field where the PageRank algorithm belongs) is completely unrelated to feature selection. We are very proud of the results that we have obtained because although there are, for sure, better feature selection algorithms than the PRFS algorithm, we have shown that there is room for new ideas to solve the feature selection problem.

After carefully analyzing the experimentation results, we have observed some positive and negative aspects of the PRFS algorithm. About the positive aspects, we have observed that:

- If we look at the accuracy plots, we can see that, in most cases, the accuracy increases very fast with the first features in the obtained ranking. If we look at the feature type count graphs, we can see that the irrelevant features tend to be concentrated in the last positions in the ranking, whereas the relevant features tend to be in the beginning. This shows that the algorithm produces results that are coherent with the logic that we presented in the description of the algorithm.

- The algorithm is very modular, and there is room for trying a lot of different parameters in the future that may produce even more interesting results.
- The algorithm is very fast. This makes sense because the algorithm is based on the PageRank algorithm designed to work with a huge dataset: the web.

Some of the negative aspects of the PRFS algorithm are:

- The algorithm only produces a ranking of the features but does not produce a specific subset. The ranking itself is very useful because we know that the features in the higher positions are usually the most interesting ones in the dataset. However, the user does not know how many of them are really important. This is a negative aspect of the algorithm compared to other feature selection algorithms that produce a specific subset of features. In the case of PRFS, the user will need to apply other techniques to determine how many of the features in the ranking wants to keep.
- The algorithm oversimplifies redundancy and relevance using the same metric (the result of the selected alpha function and beta function) for all features. It is often the case where one feature is linearly redundant/relevant, and another is non-linearly redundant/relevant.

When it comes to the parameters of the algorithm, we can conclude that:

- It is better to penalize non-relevance than penalizing redundancy. This is accomplished by using weight values lower than 1.
- Both graph types produce very similar results, but the Feature+Target graph has a great advantage in comparison with the Feature graph. In the first graph model, both the alpha function and the beta function values are separated in terms of evaluating the PageRank scores. This is very useful because we can use alpha and beta functions with totally different result ranges, such as the mutual information alpha function and the uncorrelation beta function.
- It is not easy to determine which alpha and beta function to use because it depends a lot on how the features and the target variable are correlated.

Chapter 6

Time management

This project has a total duration of 4 months and 8 days (from February 15th to June 22nd). There is a total of 92 business days in this period. Dedicating an average of 5 hours every day means that the project has an estimated duration of 460 hours. Since we have not confirmed any date to present the project, we assume the project will be presented on June 28th (the closest possible day).

6.1 Resources

Every project has a set of resources associated that are necessary to successfully develop it. These are the resources needed for this particular project:

- **Team [R1]:** we are a 2 person team formed by the author of the thesis and its director.
- **Software [R2]:**
  - **Code editor:** Visual Studio Code is an open-source editor owned by Microsoft. Nowadays, it is one of the most used editors for software development.
  - **Version Control System:** this project will use git. In particular, it will use the Github platform.
- **Hardware [R3]:** Only one computer will be necessary: a Macbook Pro running macOS Catalina.

6.2 Task definition

The following section defines all the tasks included in this project. These tasks were designed taking into consideration the objectives described in the previous chapter. There is a brief description and an estimate duration given for each task (see Table 6.1 and Figure 6.1 for a table task decomposition and Gantt diagram).

6.2.1 Project management

Every project needs proper management to define and organize the work that needs to be done. This is why the Program Management course (GEP) is very important. Some of the tasks defined in the following paragraphs take place during the discourse of the mentioned course.

Introduction and scope

This task defines the context and scope of the project. It also defines the objectives and the working methodology. This task is fundamental because it defines the path that the project will follow. 22 hours will be spent with this task.

Time management

Definition of the tasks included in this project. For each task, we will provide a definition, its dependencies, and its time decomposition. This task was finished in 8 hours.

Budget and sustainability

Very similar to the time management task. It will define the economic aspects of the project and justify its sustainability. Since this is a research project, this task will not be as sophisticated as if it had been a project developed in a company. It will take 6 hours to define the budget of this project.

Report

Elaboration of this document. One of the most significant tasks of the whole project. The thesis report will contain all the information related to the project and, this document is what typically lasts in time. We have estimated a total of 50 hours to finish this task.

Oral presentation

Once the project gets finished, it needs to be presented in a tribunal. This task includes both the preparation of the oral presentation itself and the elaboration of the slides that will be used. This task will need 14 hours.

Meetings

Every two weeks, there will be a meeting between the author and his director. They will be used to discuss the work done and prepare what will be done until the next meeting. The meetings will last 1 hour approximately. Since there are 18 weeks between the beginning and the end of the project, there will be a total of 9 meetings making an approximate total of 10 hours spent on this task.

6.2.2 Preliminary work

Before any implementation, we need to get in touch with the essential aspects related to the project's topic.

Research

To start, we need to get informed about the main topics of this project. Fortunately, some of the concepts have already been learned in courses like Machine Learning (APA) and Big Data (CAIM). We need to understand:

- **Feature selection algorithms:** what are they used for and how they work.

- **PageRank:** how does the algorithm work, where has been used before, and how to implement it.

To understand these concepts a minimum of 45 hours will be needed.

There are a lot of different Machine Learning algorithms that can be used to generate classification models. Here is a list of some of the most popular ones:

State of the art

Explore the state of the art of the feature selection problem. It will be interesting to explore how the feature selection problem is solved nowadays to compare the results obtained with the alternative method proposed in this project and evaluate its performance.

6.2.3 Minimum Viable Product

The main goal of this group of tasks is to develop an early version of the feature selection algorithm proposed.

Design

Once we have sufficient knowledge about how feature selection and PageRank work, we will design a prototype of a feature selection algorithm based on PageRank. This initial prototype must be able to solve the feature selection problem and should define two important things: how to represent the feature graph and how to calculate the PageRank scores of each feature. It does not have to be a definitive solution. We will design this prototype using pseudocode. When we have the pseudocode, we will have an idea about the feasibility of the proposed solution and we can move on to the implementation. Because of the characteristics of this project, the tasks that are related to the design of the algorithm will be very time-consuming. This is why the estimated duration of this task will be of 70 hours.

Implementation

This task consists of the pseudocode implementation proposed in the previous task using the Python programming language. As mentioned before, this implementation must provide the user the possibility to solve the feature selection problem. This task will take 50 hours to complete.

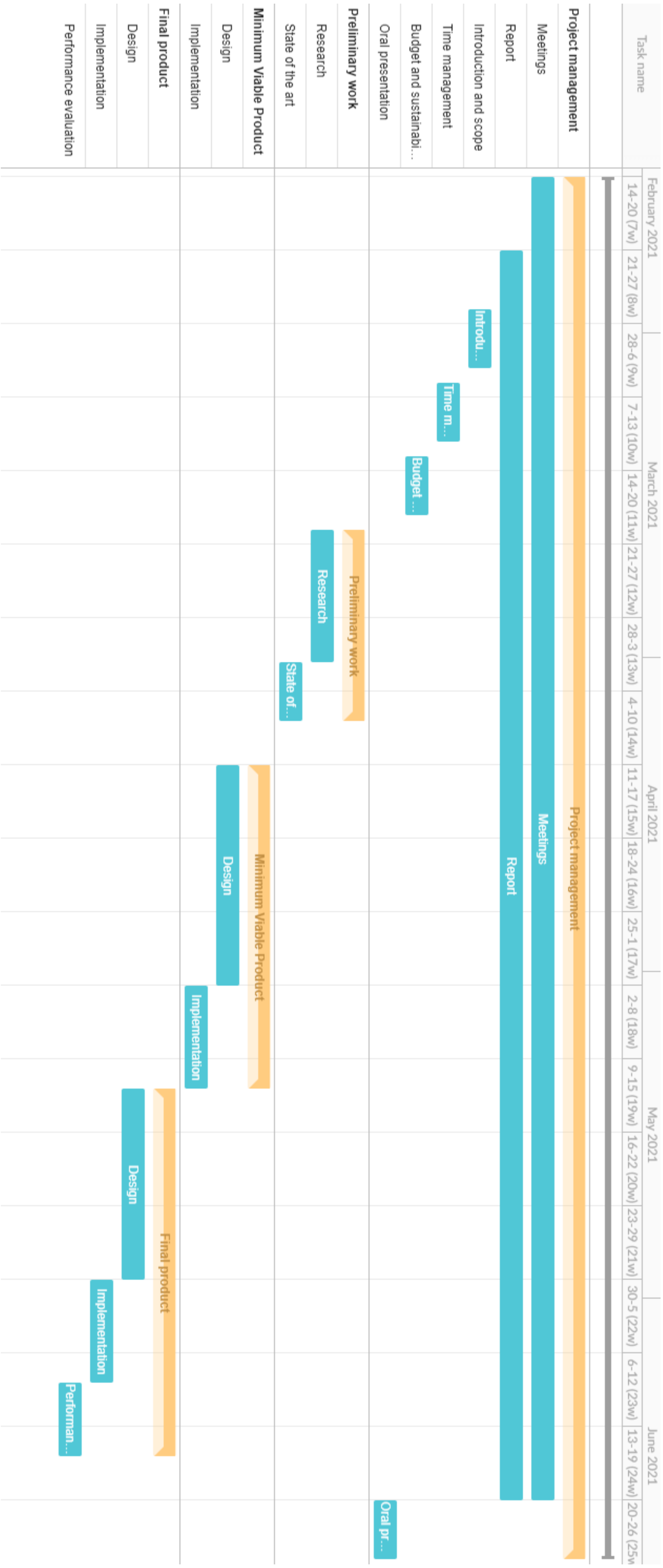
6.2.4 Final product

This is one of the last tasks of the project. At the end of this task, we must have a complete feature selection algorithm based on PageRank implemented in Python. This complete version will incorporate more feature graph representations than the minimum viable product. It will also incorporate validation methods to evaluate the feature selections made by the algorithm. This task has a duration of 160 hours.

TABLE 6.1: Project task decomposition.

Code	Name	Duration	Dependencies	Resources
T1	**Project management**	**110h**		
T1.1	Introduction and scope	22h		R1, R3
T1.2	Time management	8h	T1.1	R1, R3
T1.3	Budget and sustainability	6h	T1.1, T1.2	R1, R3
T1.4	Report	50h	T1.4	R1, R3
T1.5	Oral presentation	14h	T1.4	R1, R3
T1.6	Meetings	10h	T1.4	R1, R3
T2	**Preliminary work**	**70h**		
T2.1	Research	45h		R1, R3
T2.2	State of the art	8h	T2.1	R1, R3
T3	**Minimum Viable Product**	**120h**	T2	
T3.1	Design	70h		R1, R2, R3
T3.1.1	Feature graph	45h		
T3.1.2	PageRank	25h		
T3.2	Implementation	50h	T3.1	R1, R2, R3
T4	**Final product**	**160h**	T3	
T4.1	Design	70h		R1, R2, R3
T4.1.1	Additional feature graphs	45h		
T4.1.2	Evaluation algorithms	25h	T4.1.1	
T4.2	Implementation	50h	T4.1	R1, R2, R3
T4.3	Performance evaluation	40h	T4.2	R1, R2, R3
Total		**460h**		

FIGURE 6.1: Gantt diagram.



6.3 Risk management

In the previous chapter, we mentioned the possible obstacles that may occur during the realization of this project. The following points discuss what would happen if we, unfortunately, face any of the possible problems. For each problem we will propose a solution or an alternative task:

- **Disease or injury:** this obstacle will cause a delay in the following deadlines. Unfortunately, this obstacle is unpredictable and the only thing that we can do to mitigate its effects is to reduce the time dedication of the next tasks.
- **Errors during the design and implementation of the algorithm:** it is possible to face obstacles during the design and implementation of the feature selection algorithm. In the worst-case scenario, we will need to focus on the implementation of a less sophisticated working prototype.
- **Hardware failure:** in the case of an eventual failure in the computer used to develop the project, fortunately, the amount of work lost will not be severe because the team will use appropriate backup techniques. This obstacle has a nearly immediate solution because the author has more than one computer available.

Chapter 7

Budget

In this chapter, we will elaborate a detailed economic study to estimate the budget of this project. Once we have an idea of the resources needed, we will be able to determine its feasibility.

7.1 Staff costs

To start, we need to identify the different roles that are involved in the project: project manager, junior researcher, and junior developer. The project management role will be shared between the director and the author of the project. The roles of junior researcher and junior software developer will be individually assumed by the author. The following table (7.1) illustrates the cost per hour of work for each of the mentioned roles:

TABLE 7.1: Cost per hour of the different roles.

Role	Gross Salary [13]	SS	Total Cost
Project Manager	23 €/h	6.9 €/h	29.9 €/h
Junior Researcher	15 €/h	4.5 €/h	19.5 €/h
Junior Developer	22 €/h	6.6 €/h	28.6 €/h

Once we have defined the cost per hour of each role, we can calculate the CPA (Cost per activity). This can be done using the time management information that we described in the previous chapter. Table 7.2 shows the roles that are involved in each task and its total cost.

TABLE 7.2: Staff cost per activity.

Code	Name	Duration	PM	Researcher	Developer	Cost
T1	**Project management**	**110h**	**110h**	**-**	**-**	**3,289 €**
T1.1	Introduction and scope	22h	22h	-	-	657.8 €
T1.2	Time management	8h	8h	-	-	239.2€
T1.3	Budget and sustainability	6h	6h	-	-	179.4 €
T1.4	Report	50h	50h	-	-	1,495 €
T1.5	Oral presentation	14h	14h	-	-	418.6 €
T1.6	Meetings	10h	10h	-	-	299 €
T2	**Preliminary work**	**70h**	**-**	**70h**	**-**	**1,365 €**
T2.1	Research	45h	-	45h	-	877.5 €
T2.2	State of the art	25h	-	25h	-	487.5 €
T3	**Minimum Viable Product**	**120h**	**-**	**35h**	**85h**	**3,113.5 €**
T3.1	Design	70h	-	35h	35h	1,683.5 €
T3.2	Implementation	50h	-	-	50h	1,430 €
T4	**Final product**	**160h**	**-**	**55h**	**105h**	**4,075.5 €**
T4.1	Design	70h	-	35h	35h	1,683.5 €
T4.2	Implementation	50h	-	-	50h	1,430 €
T4.3	Performance evaluation	40h	-	20h	20h	962 €
Total		**460h**	**110h**	**160h**	**190h**	**11,843 €**

7.2 Generic costs

7.2.1 Amortitzation of the resources

- **Hardware:** The hardware used during this project will be a MacBook Pro (1448 €) and an external monitor (200 €). If we assume that the hardware used has a life expectancy of 5 years (60 months) the amortization of this resource will be $5/50 * (1448 + 200) =$ **154.8 €**
- **Software:** all software used in this project will be open source and free. The software amortization will be **0 €**

7.2.2 Indirect costs

To estimate the budget of the project we should also take into consideration indirect costs related to it. Since the thesis will be developed using a computer and an internet connection, we need to calculate the cost of the electricity and the internet service provider bill:

- **Internet:** the internet service provider bill is around 50 € per month. Hence, the total cost will be $50/(30 * 24) * 460 =$ **31.94 €**
- **Electricity:** At the moment, the kWh price is 0.07622 € [14]. Given that the wattage of the computer and the monitor used is 100W, the total cost of the electricity is $0.1 * 460 * 0.07622 =$ **3.51 €**

7.2.3 Total generic costs

The following table summaizes the total amount of generic costs of this thesis:

TABLE 7.3: Generic costs.

Name	Cost
Amortization	154.8 €
Internet	31.94 €
Electricity	3.51 €
Total	**190.25 €**

7.3 Contingency

We may experience delays and complications during the development of this project that could cause additional costs. Software development projects usually prepare an additional 15% of the general cost of the project as a contingency plan. The contingency cost of this project is $(11843 + 190.35) * 0.15 =$ **1,805 €**.

7.4 Incidental costs

The previous chapter has defined the possible risks that may appear durings this thesis. The following table shows the costs of applying the alternative plans related to each risk:

TABLE 7.4: Incidental costs.

Name	Estimated cost	Risk	Cost
New computer	1,449 €	5%	72.45€
Design time increase (20h)	572 €	20%	114.4€
Implementation time increase (15h)	429 €	20%	85.8€
Total			**272.65 €**

7.5 Final budget

This section summarizes all the information explained about the costs of this project:

TABLE 7.5: Final budget.

Name	Cost
Staff costs	11,843 €
General costs	190.25 €
Contingency	1,805 €
Incidental costs	272.65 €
Total	**14,110.9 €**

7.6 Management control

This section explains how are we going to identify alterations in the project cost.

We will use the following formulas:

- **Estimated cost:** the cost that have estimated for each task (see Table 7.2).
- **Real cost:** the actual cost that will be spent in the end. It will be used to recalculate the estimated cost of other tasks and identify tasks that have been miss-estimated.
- **Deviation:** the difference between the estimated cost of a task and its real one.

The indicator that we will focus on fundamentally is the deviation of each task. Notice that a positive deviation means that we have over-estimated the cost of a given task and we will destinate more resources to other tasks that need them. In contrast, a negative deviation means that we have under-estimated the costs of the task and we will have to reduce the resources that were destined for other tasks to accomplish the objectives of the underestimated task.

Chapter 8

Sustainability

8.1 Environmental impact

Have you thought about the environmental impact of your project? Have you considered minimizing this impact, by for example reusing resources?

Estimating the environmental impact of this thesis is not an easy thing to do because of its particular characteristics. The only remarkable impact that we could consider is the amount of energy that will be consumed during the project. We will use a very reduced quantity of energy thanks to the hardware that will be used.

How is it solved the problem you are trying to solve? Does your solution provide any improvement in the environmental impact?

The problem that we are trying to solve is feature selection. As we described in the first chapter, one of the use cases of feature selection is to reduce the dimensionality of a dataset that is going to be used to train a machine learning model. Usually, when the dimensionality of a dataset is reduced, the training time also decreases significantly. If the training time is reduced, the resources consumed will also be meaningfully smaller. Nowadays, there are solutions to the feature selection problem but, if the solution that we develop performs better than the existing ones, it will have a positive environmental impact.

8.2 Economy

Have you estimated the impact that your project will have, including both human and material costs?

In chapter 3, we have already estimated the impact of this thesis considering human and material costs.

How is it solved the problem you are trying to solve? Does your solution provide any improvement economically?

We have already mentioned that feature selection reduces the resources that will be needed when training a machine learning model. The amount of resources used has a strong direct correlation to the number of economical assets that will be spent.

8.3 Social

How do you think this project will enrich you personally?

To start, this thesis allows me to learn about a lot of topics that I am very interested in such as machine learning. Also, I will be surrounded by professionals with a lot of experience that will guide me through this journey and enrich me personally.

How is it solved the problem you are trying to solve? How do you think your solution will improve people's quality of life? Is there a real need of developing your solution?

The problem that we want to solve has a strong relationship with machine learning. Hence, society, in general, could be beneficiated because machine learning is used widely in today's society. Our solution is not strictly necessary because there are solutions to the feature selection algorithm already available but it will provide an alternative solution.

Appendix A

Results

This appendix recompiles some of the results obtained during the evaluation of the algorithm. It does not contain all the graphs that we have generated because there were a lot of them. We have divided the graphs and tables into three sections: rankings, accuracy results, feature type results and execution time results.

A.1 Rankings

Rank	**Feature name**
1	d1_d2_sum
2	dice3
3	dice4
4	dice1
5	dice2
6	i1
7	i4
8	i3
9	i2

TABLE A.1: Ranking of dice1 dataset using Feature graph, $\alpha = \mathit{correlation}$, $\beta = \mathit{uncorrelation}$ and $w = 0.2$

Rank	**Feature name**
1	d1_d2_sum
2	dice3
3	dice4
4	dice1
5	dice2
6	i1
7	i4
8	i3
9	i2

TABLE A.2: Ranking of dice1 dataset using Feature graph, $\alpha = \mathit{correlation}$, $\beta = \mathit{uncorrelation}$ and $w = 0.5$

Rank	**Feature name**
1	dice3
2	dice4
3	d1_d2_sum
4	dice1
5	dice2
6	i4
7	i1
8	i3
9	i2

TABLE A.3: Ranking of dice1 dataset using Feature graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.8$

Rank	**Feature name**
1	d1_d2_sum
2	dice4
3	dice3
4	i4
5	i3
6	i2
7	i1
8	dice1
9	dice2

TABLE A.4: Ranking of dice1 dataset using Feature graph, $\alpha = spearman\_correlation$, $\beta = spearman\_uncorrelation$ and $w = 0.2$

Rank	**Feature name**
1	d1_d2_sum
2	dice4
3	i4
4	dice3
5	i3
6	i2
7	i1
8	dice1
9	dice2

TABLE A.5: Ranking of dice1 dataset using Feature graph, $\alpha = spearman\_correlation$, $\beta = spearman\_uncorrelation$ and $w = 0.5$

Rank	Feature name
1	d1_d2_sum
2	dice4
3	i4
4	dice3
5	i3
6	i2
7	i1
8	dice1
9	dice2

TABLE A.6: Ranking of dice1 dataset using Feature graph, $\alpha = spearman\_correlation$, $\beta = spearman\_uncorrelation$ and $w = 0.8$

Rank	Feature name
1	d1_d2_sum
2	dice3
3	dice4
4	dice1
5	dice2
6	i1
7	i4
8	i3
9	i2

TABLE A.7: Ranking of dice1 dataset using Feature+Target graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.2$

Rank	Feature name
1	dice3
2	dice4
3	d1_d2_sum
4	dice1
5	dice2
6	i4
7	i1
8	i3
9	i2

TABLE A.8: Ranking of dice1 dataset using Feature+Target graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.5$

Rank	**Feature name**
1	dice3
2	dice4
3	dice1
4	d1_d2_sum
5	dice2
6	i4
7	i1
8	i3
9	i2

TABLE A.9: Ranking of dice1 dataset using Feature+Target graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.8$

Rank	**Feature name**
1	d1_d2_sum
2	dice4
3	dice3
4	dice1
5	dice2
6	i4
7	i1
8	i3
9	i2

TABLE A.10: Ranking of dice1 dataset using Feature+Target graph, $\alpha = accuracy$, $\beta = accuracy$ and $w = 0.2$

Rank	**Feature name**
1	d3_d4_sum
2	d1_d2_sum
3	dice3
4	dice4
5	dice1
6	dice2
7	i1
8	i3
9	i8
10	i4
11	i1
12	i7
13	i6
14	i2

TABLE A.11: Ranking of dice2 dataset using Feature graph, $\alpha = correlation$, $\beta = correlation$ and $w = 0.2$

Rank	Feature name
1	d3_d4_sum
2	d1_d2_sum
3	dice3
4	dice1
5	dice4
6	dice2
7	i1
8	i3
9	i8
10	i4
11	i1
12	i7
13	i6
14	i2

TABLE A.12: Ranking of dice2 dataset using Feature+Target graph, $\alpha = accuracy$, $\beta = accuracy$ and $w = 0.5$

Rank	Feature name
1	d3_d4_sum
2	d1_d2_sum
3	dice3
4	dice1
5	dice4
6	dice2
7	i7
8	i4
9	i2
10	i1
11	i3
12	i5
13	i8
14	i6

TABLE A.13: Ranking of dice2 dataset using Feature+Target graph, $\alpha = mutual\_information$, $\beta = uncorrelation$ and $w = 0.5$

Rank	Feature name
1	relevant4
2	redundant4
3	relevant1
4	relevant2
5	relevant5
6	relevant3
7	redundant1
8	redundant3
9	redundant5
10	redundant2
11	irrelevant3
12	irrelevant2
13	irrelevant1
14	irrelevant5
15	irrelevant4

TABLE A.14: Ranking of sklearn1 dataset using Feature graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.2$

Rank	Feature name
1	relevant4
2	redundant4
3	relevant1
4	relevant5
5	relevant2
6	relevant3
7	redundant1
8	redundant3
9	redundant5
10	irrelevant3
11	irrelevant2
12	irrelevant1
13	irrelevant5
14	irrelevant4
15	redundant2

TABLE A.15: Ranking of sklearn1 dataset using Feature graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.5$

Rank	Feature name
1	relevant4
2	redundant4
3	relevant1
4	relevant2
5	relevant5
6	relevant3
7	irrelevant1
8	irrelevant2
9	irrelevant5
10	irrelevant4
11	irrelevant3
12	redundant3
13	redundant2
14	redundant1
15	redundant5

TABLE A.16: Ranking of sklearn1 dataset using Feature graph, $\alpha = spearman\_correlation$, $\beta = spearman\_uncorrelation$ and $w = 0.2$

Rank	Feature name
1	relevant4
2	redundant4
3	relevant1
4	relevant3
5	relevant2
6	relevant5
7	redundant3
8	redundant1
9	redundant5
10	redundant2
11	irrelevant3
12	irrelevant1
13	irrelevant4
14	irrelevant2
15	irrelevant5

TABLE A.17: Ranking of sklearn1 dataset using Feature+Target graph, $\alpha = accuracy$, $\beta = accuracy$ and $w = 0.2$

Rank	Feature name
1	relevant4
2	redundant4
3	relevant1
4	relevant3
5	relevant5
6	relevant2
7	redundant3
8	irrelevant1
9	irrelevant2
10	irrelevant3
11	irrelevant5
12	irrelevant4
13	redundant1
14	redundant5
15	redundant2

TABLE A.18: Ranking of sklearn1 dataset using Feature+Target graph, $\alpha = mutual\_information$, $\beta = uncorrelation$ and $w = 0.2$

Rank	Feature name
1	cp
2	exang
3	oldpeak
4	thalach
5	ca
6	thal
7	slope
8	sex
9	age
10	restecg
11	trestbps
12	chol
13	fbs

TABLE A.19: Ranking of heart dataset using Feature graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.2$

Rank	Feature name
1	fbs
2	restecg
3	chol
4	sex
5	trestbps
6	cp
7	thal
8	slope
9	ca
10	exang
11	age
12	thalach
13	oldpeak

TABLE A.20: Ranking of heart dataset using Feature graph, $\alpha = spearman\_correlation$, $\beta = spearman\_uncorrelation$ and $w = 0.5$

Rank	Feature name
1	cp
2	thal
3	exang
4	ca
5	chol
6	slope
7	thalach
8	oldpeak
9	sex
10	fbs
11	restecg
12	trestbps
13	age

TABLE A.21: Ranking of heart dataset using Feature+Target graph, $\alpha = mutual\_information$, $\beta = uncorrelation$ and $w = 0.2$

Rank	**Feature name**
1	ca
2	cp
3	thal
4	chol
5	oldpeak
6	exang
7	sex
8	thalach
9	slope
10	fbs
11	restecg
12	trestbps
13	age

TABLE A.22: Ranking of heart dataset using Feature+Target graph, $\alpha = mutual\_information$, $\beta = uncorrelation$ and $w = 0.5$

Rank	**Feature name**
1	thal
2	cp
3	exang
4	slope
5	oldpeak
6	thalach
7	ca
8	age
9	sex
10	restecg
11	fbs
12	trestbps
13	chol

TABLE A.23: Ranking of heart dataset using Feature+Target graph, $\alpha = accuracy$, $\beta = accuracy$ and $w = 0.2$

Rank	**Feature name**
1	cp
2	thal
3	exang
4	oldpeak
5	slope
6	thalach
7	ca
8	age
9	sex
10	trestbps
11	restecg
12	chol
13	fbs

TABLE A.24: Ranking of heart dataset using Feature+Target graph, $\alpha = accuracy$, $\beta = accuracy$ and $w = 0.5$

Rank	**Feature name**
1	Sex
2	Pclass
3	Cabin
4	Fare
5	Parch
6	Age
7	SibSp

TABLE A.25: Ranking of titanic dataset using Feature graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.2$

Rank	**Feature name**
1	Sex
2	SibSp
3	Age
4	Fare
5	Cabin
6	Pclass
7	Parch

TABLE A.26: Ranking of titanic dataset using Feature graph, $\alpha = spearman\_correlation$, $\beta = spearman\_uncorrelation$ and $w = 0.2$

Rank	Feature name
1	Sex
2	SibSp
3	Cabin
4	Fare
5	Age
6	Pclass
7	Parch

TABLE A.27: Ranking of titanic dataset using Feature graph, $\alpha = spearman\_correlation$, $\beta = spearman\_uncorrelation$ and $w = 0.2$

Rank	Feature name
1	Sex
2	Fare
3	Pclass
4	Cabin
5	SibSp
6	Age
7	Parch

TABLE A.28: Ranking of titanic dataset using Feature+Target graph, $\alpha = mutual\_information$, $\beta = uncorrelation$ and $w = 0.2$

Rank	Feature name
1	Sex
2	Fare
3	Pclass
4	Parch
5	Cabin
6	SibSp
7	Age

TABLE A.29: Ranking of titanic dataset using Feature+Target graph, $\alpha = mutual\_information$, $\beta = uncorrelation$ and $w = 0.5$

Rank	Feature name
1	Sex
2	Cabin
3	Pclass
4	Fare
5	Age
6	SibSp
7	Parch

TABLE A.30: Ranking of titanic dataset using Feature+Target graph, $\alpha = accuracy$, $\beta = accuracy$ and $w = 0.2$

Rank	**Feature name**
1	Sex
2	Cabin
3	Pclass
4	Fare
5	Age
6	Parch
7	SibSp

TABLE A.31: Ranking of titanic dataset using Feature+Target graph, $\alpha = accuracy$, $\beta = accuracy$ and $w = 0.5$

A.2 Accuracy results

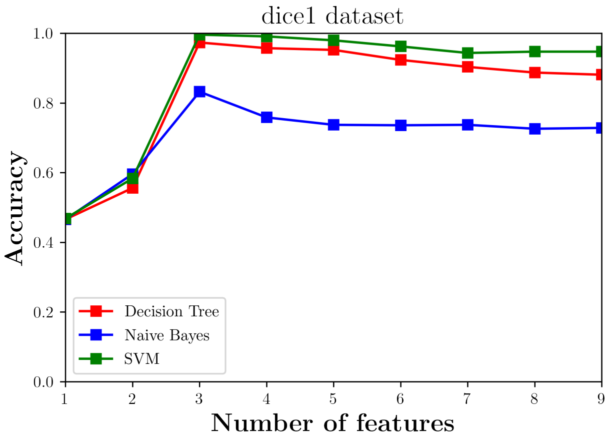


FIGURE A.1: Accuracy of dice1 dataset using Feature graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.2$

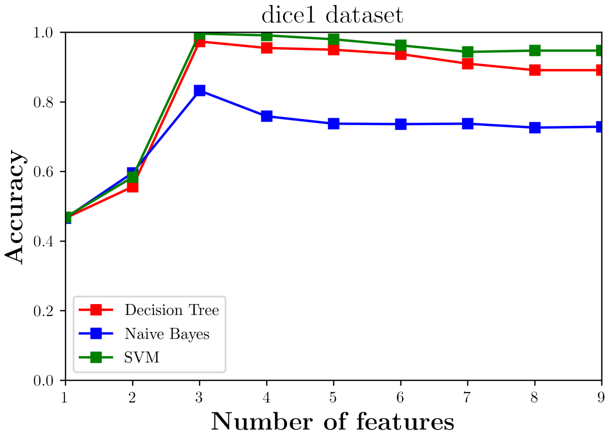


FIGURE A.2: Accuracy of dice1 dataset using Feature graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.5$

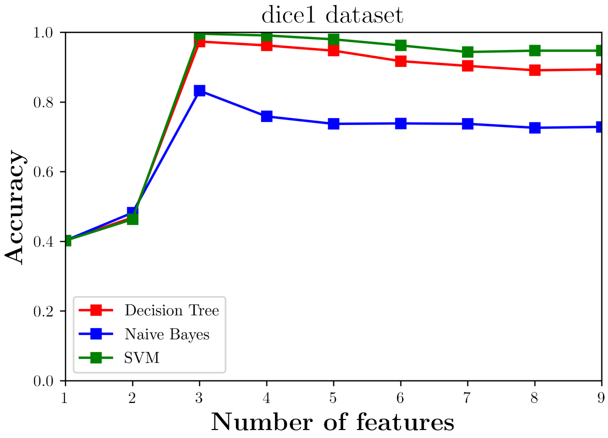


FIGURE A.3: Accuracy of dice1 dataset using Feature graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.8$

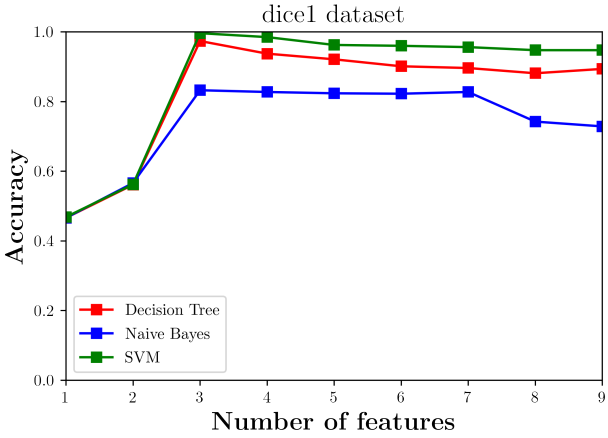


FIGURE A.4: Accuracy of dice1 dataset using Feature graph, $\alpha = spearman\_correlation$, $\beta = spearman\_uncorrelation$ and $w = 0.2$

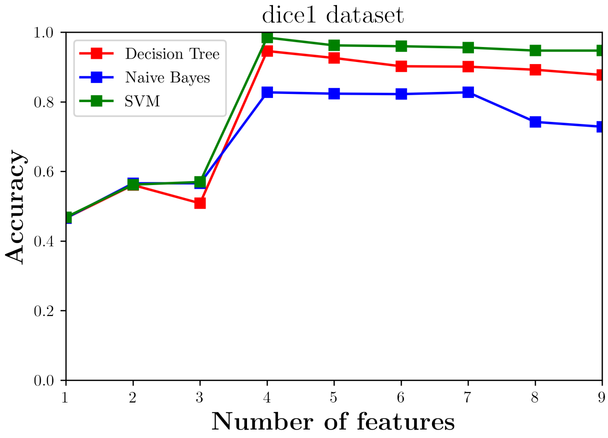


FIGURE A.5: Accuracy of dice1 dataset using Feature graph, $\alpha = spearman\_correlation$, $\beta = spearman\_uncorrelation$ and $w = 0.5$

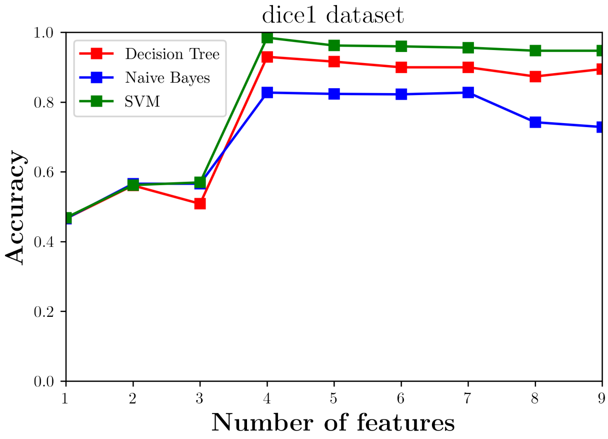


FIGURE A.6: Accuracy of dice1 dataset using Feature graph, $\alpha = spearman\_correlation$, $\beta = spearman\_uncorrelation$ and $w = 0.8$

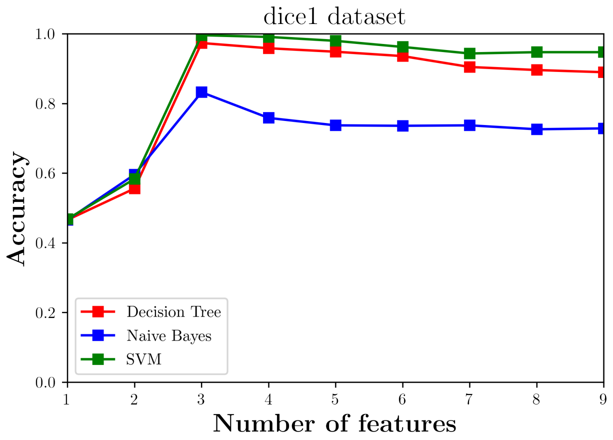


FIGURE A.7: Accuracy of dice1 dataset using Feature+Target graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.2$

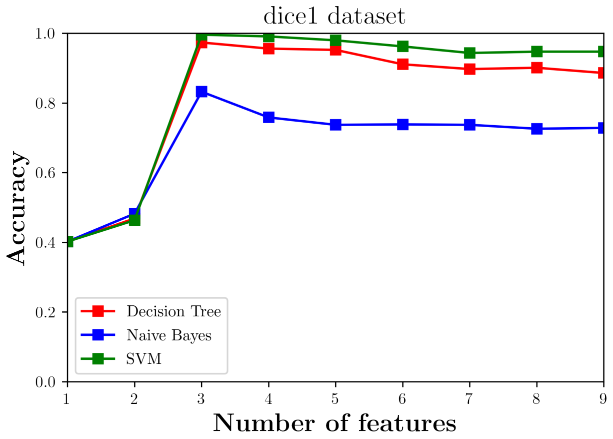


FIGURE A.8: Accuracy of dice1 dataset using Feature+Target graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.5$

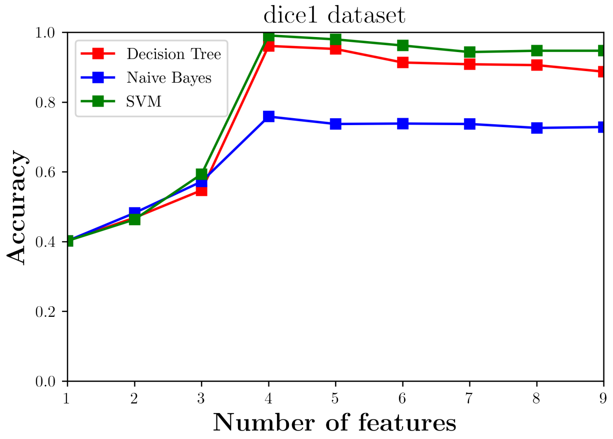


FIGURE A.9: Accuracy of dice1 dataset using Feature+Target graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.8$

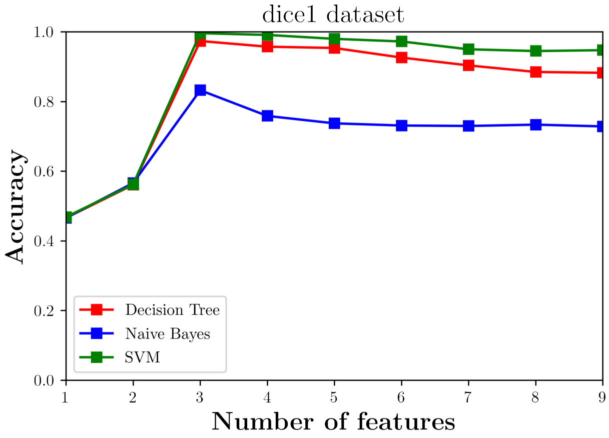


FIGURE A.10: Accuracy of dice1 dataset using Feature+Target graph, $\alpha = accuracy$, $\beta = accuracy$ and $w = 0.2$

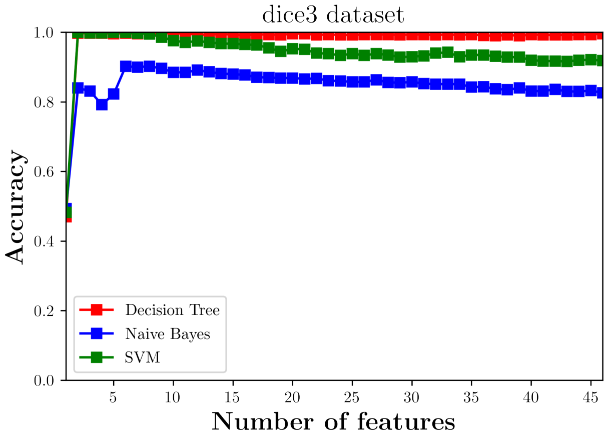


FIGURE A.11: Accuracy of dice3 dataset using Feature+Target graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.5$

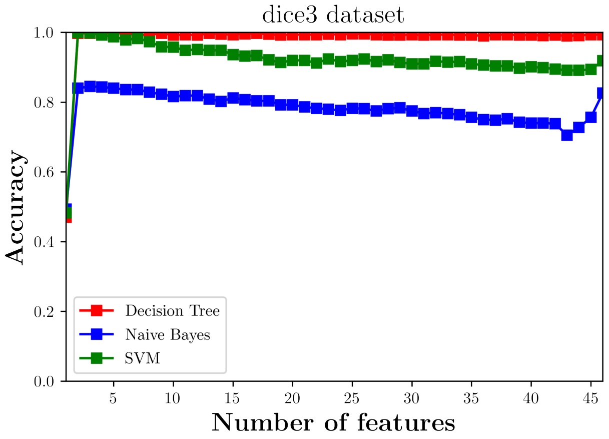


FIGURE A.12: Accuracy of dice3 dataset using Feature graph, $\alpha = spearman\_correlation$, $\beta = spearman\_uncorrelation$ and $w = 0.5$

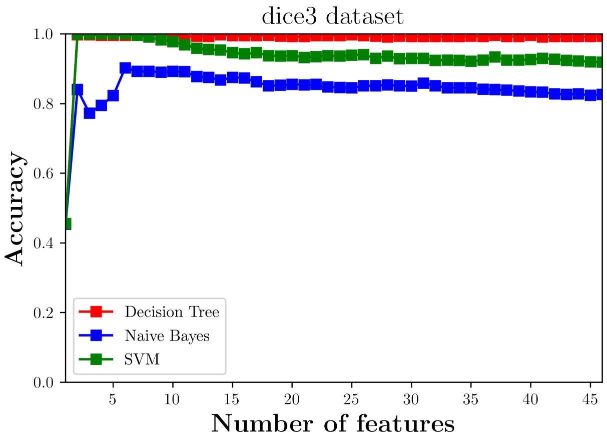


FIGURE A.13: Accuracy of dice3 dataset using Feature+Target graph, $\alpha = mutual\_information$, $\beta = uncorrelation$ and $w = 0.5$

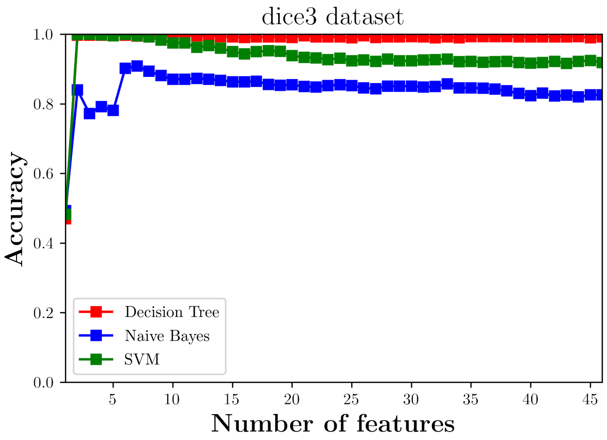


FIGURE A.14: Accuracy of dice3 dataset using Feature+Target graph, $\alpha = accuracy$, $\beta = accuracy$ and $w = 0.5$

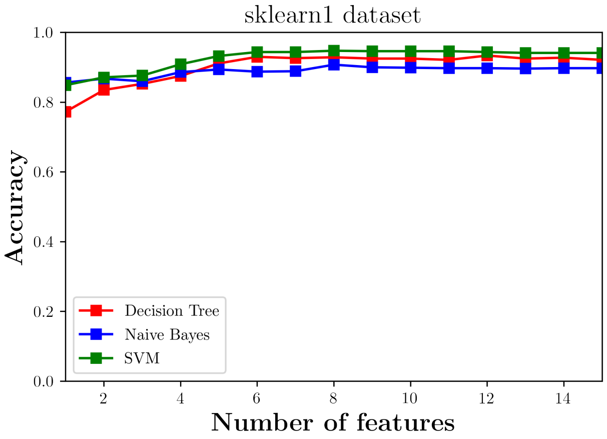


FIGURE A.15: Accuracy of sklearn dataset using Feature graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.2$

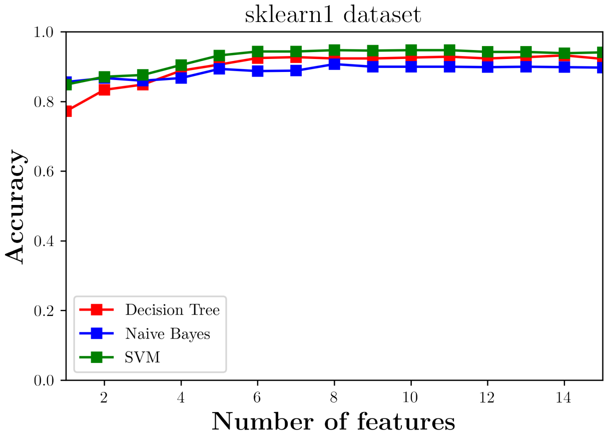


FIGURE A.16: Accuracy of sklearn1 dataset using Feature graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.5$

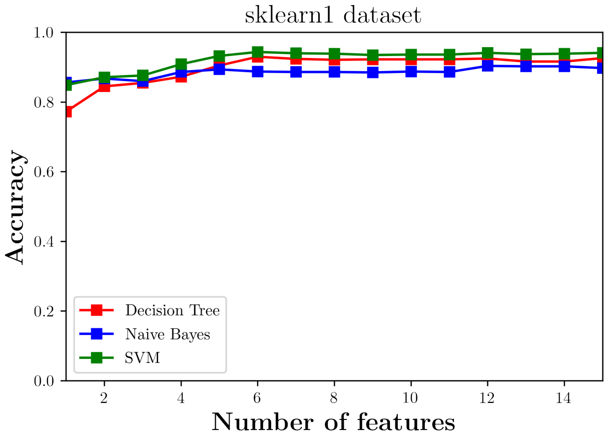


FIGURE A.17: Accuracy of sklearn1 dataset using Feature graph, $\alpha = spearman\_correlation$, $\beta = spearman\_uncorrelation$ and $w = 0.2$

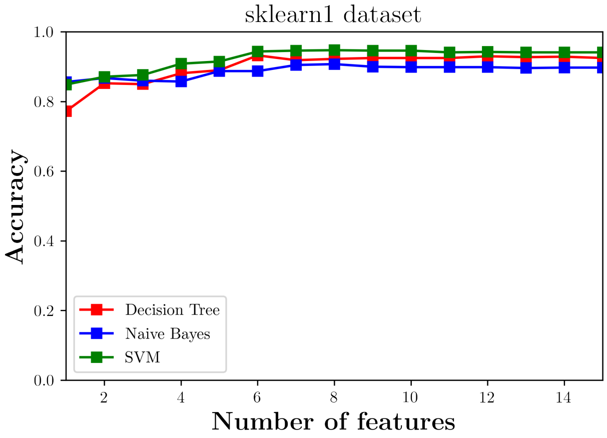


FIGURE A.18: Accuracy of sklearn1 dataset using Feature+Target graph, $\alpha = accuracy$, $\beta = accuracy$ and $w = 0.2$

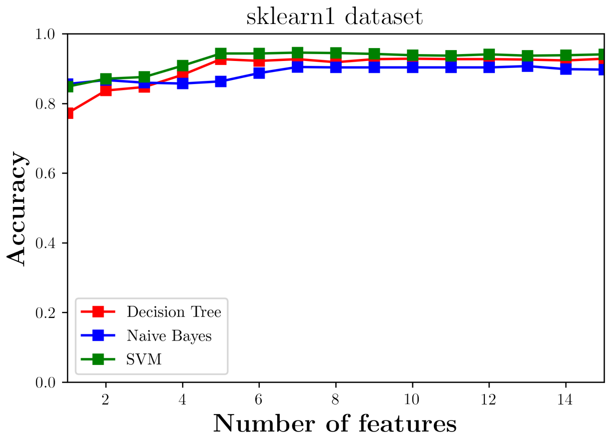


FIGURE A.19: Accuracy of sklearn1 dataset using Feature+Target graph, $\alpha = mutual\_information$, $\beta = uncorrelation$ and $w = 0.2$

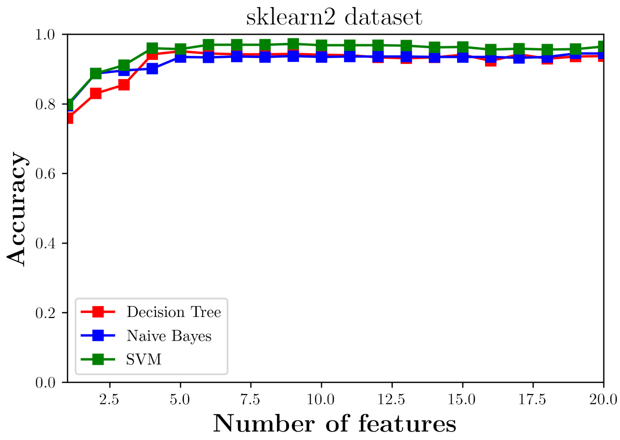


FIGURE A.20: Accuracy of sklearn2 dataset using Feature graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.2$

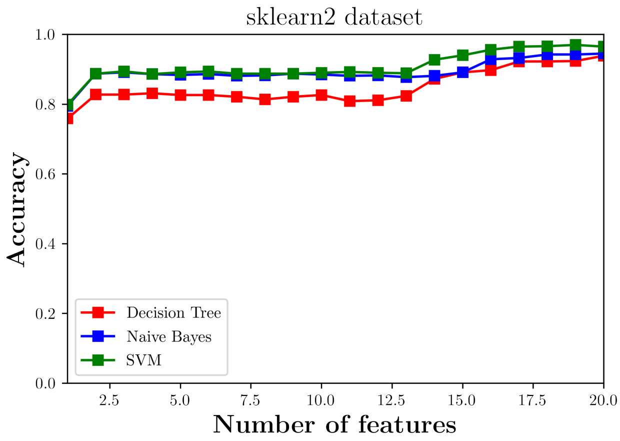


FIGURE A.21: Accuracy of sklearn2 dataset using Feature graph, $\alpha = spearman\_correlation$, $\beta = spearman\_uncorrelation$ and $w = 0.2$

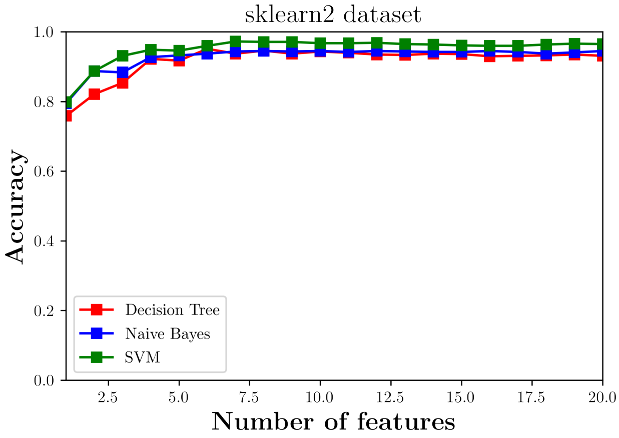


FIGURE A.22: Accuracy of sklearn2 dataset using Feature+Target graph, $\alpha = mutual\_information$, $\beta = uncorrelation$ and $w = 0.2$

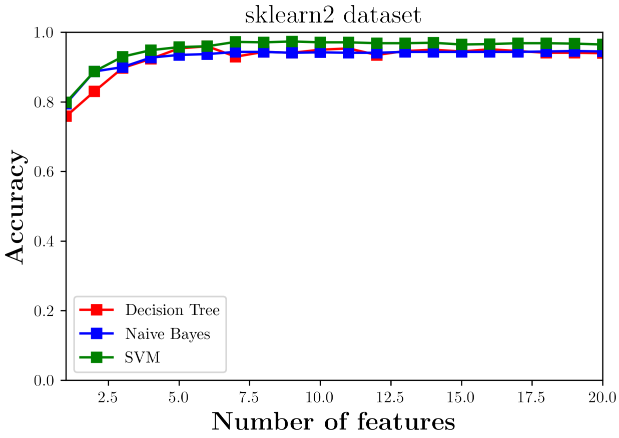


FIGURE A.23: Accuracy of sklearn2 dataset using Feature+Target graph, $\alpha = accuracy$, $\beta = accuracy$ and $w = 0.2$

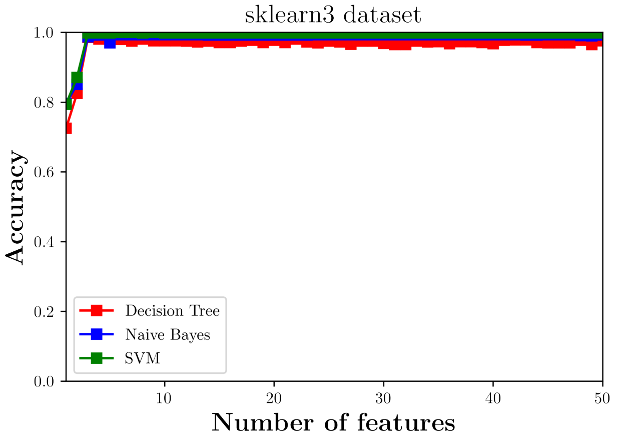


FIGURE A.24: Accuracy of sklearn3 dataset using Feature graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.5$

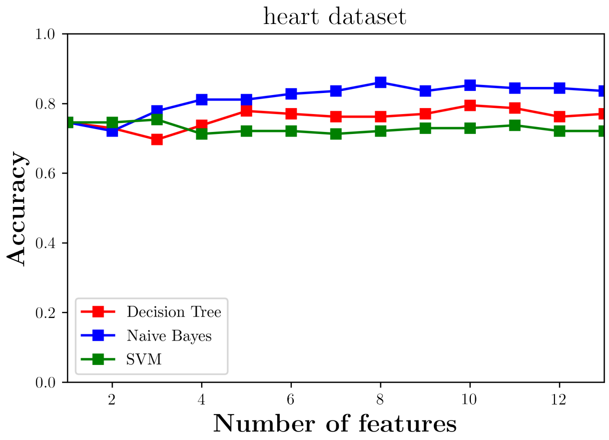


FIGURE A.25: Accuracy of heart dataset using Feature graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.2$

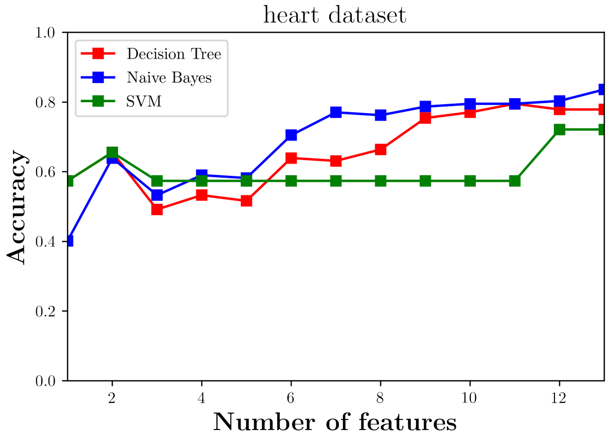


FIGURE A.26: Accuracy of heart dataset using Feature graph, $\alpha = spearman\_correlation$, $\beta = spearman\_uncorrelation$ and $w = 0.5$

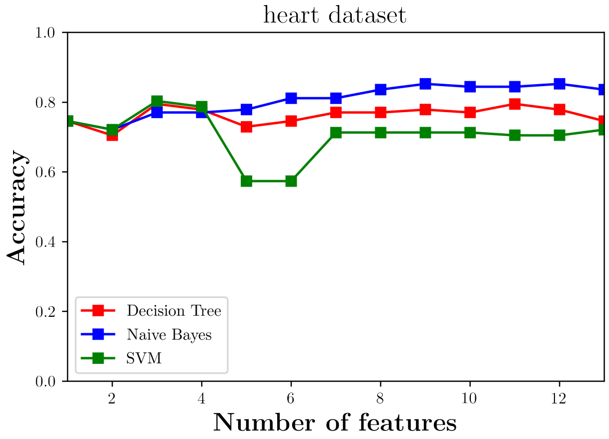


FIGURE A.27: Accuracy of heart dataset using Feature+Target graph, $\alpha = mutual\_information$, $\beta = uncorrelation$ and $w = 0.2$

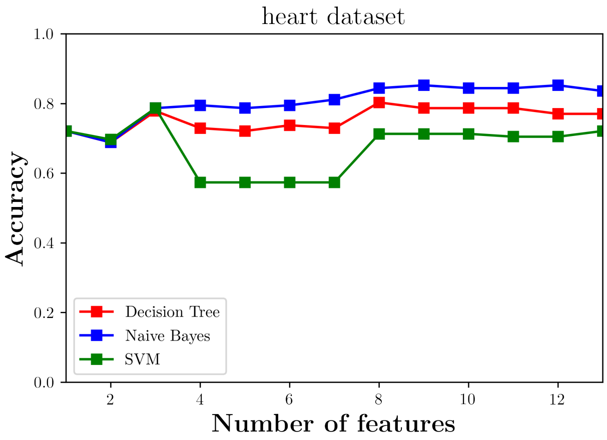


FIGURE A.28: Accuracy of heart dataset using Feature+Target graph, $\alpha = mutual\_information$, $\beta = uncorrelation$ and $w = 0.5$

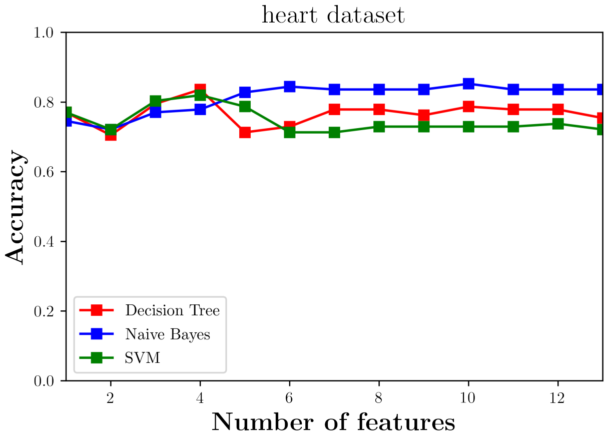


FIGURE A.29: Accuracy of heart dataset using Feature+Target graph, $\alpha = accuracy$, $\beta = accuracy$ and $w = 0.2$

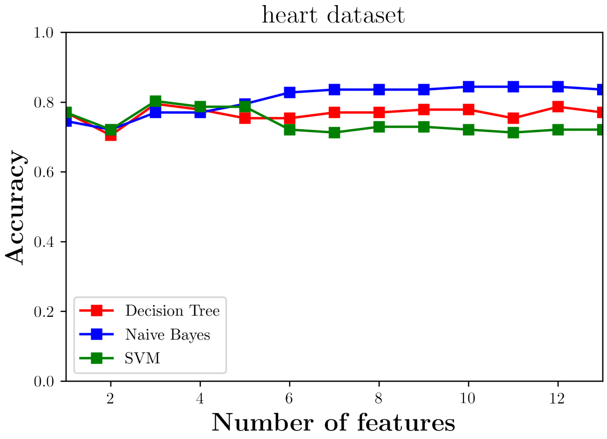


FIGURE A.30: Accuracy of heart dataset using Feature+Target graph, $\alpha = accuracy$, $\beta = accuracy$ and $w = 0.5$

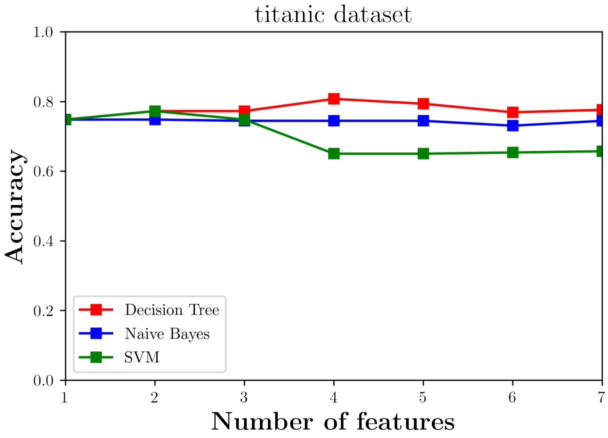


FIGURE A.31: Accuracy of titanic dataset using Feature graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.2$

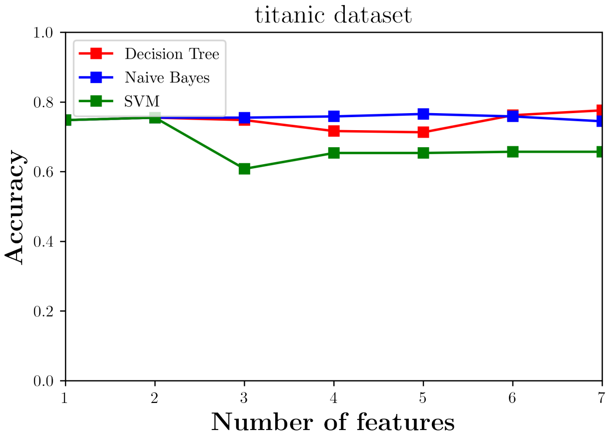


FIGURE A.32: Accuracy of titanic dataset using Feature graph, $\alpha = spearman\_correlation$, $\beta = spearman\_uncorrelation$ and $w = 0.2$

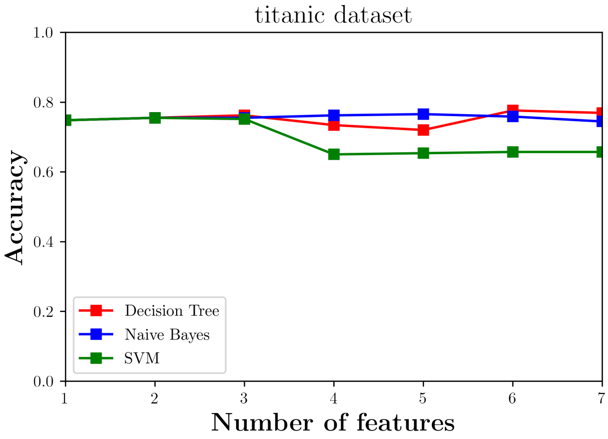


FIGURE A.33: Accuracy of titanic dataset using Feature graph, $\alpha = spearman\_correlation$, $\beta = spearman\_uncorrelation$ and $w = 0.5$

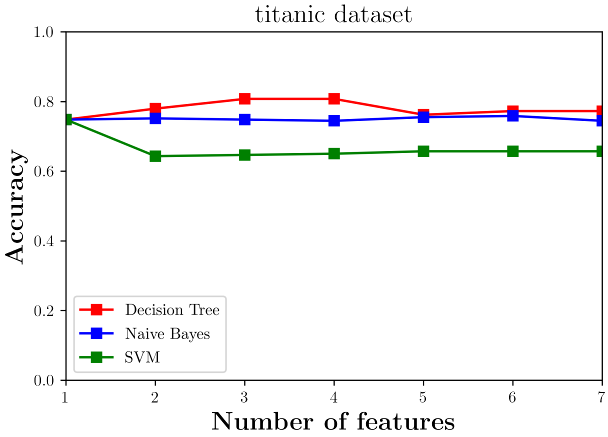


FIGURE A.34: Accuracy of titanic dataset using Feature+Target graph, $\alpha = mutual\_information$, $\beta = uncorrelation$ and $w = 0.2$

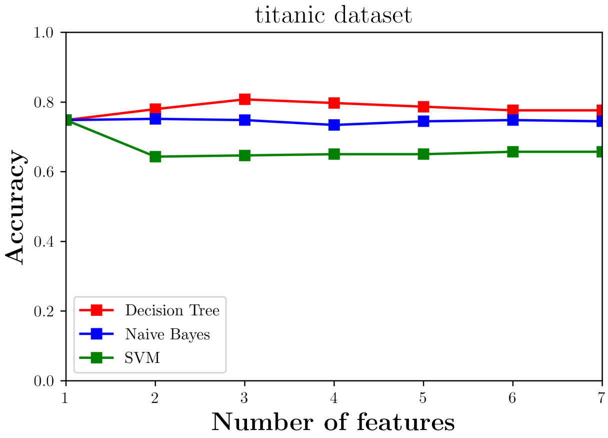


FIGURE A.35: Accuracy of titanic dataset using Feature+Target graph, $\alpha = mutual\_information$, $\beta = uncorrelation$ and $w = 0.5$

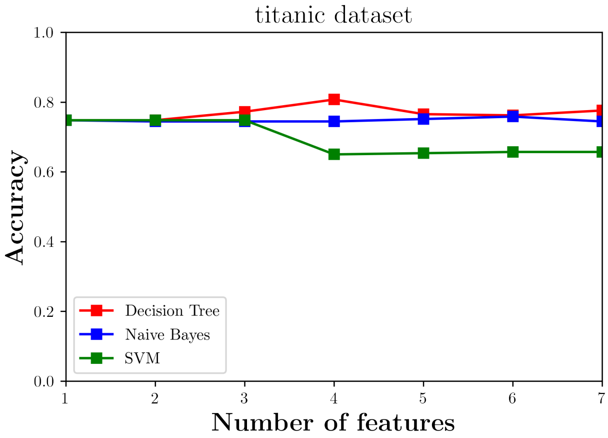


FIGURE A.36: Accuracy of titanic dataset using Feature+Target graph, $\alpha = accuracy$, $\beta = accuracy$ and $w = 0.2$

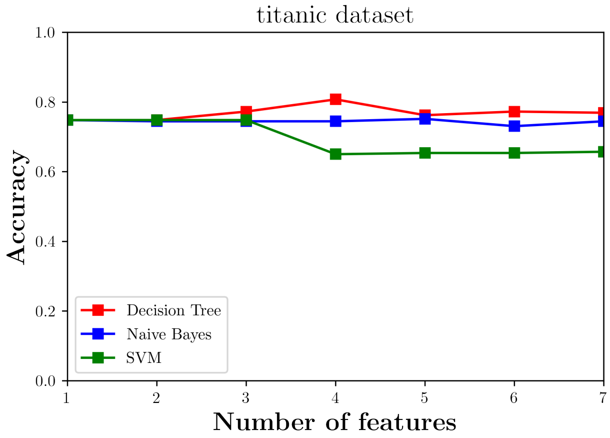


FIGURE A.37: Accuracy of titanic dataset using Feature+Target graph, $\alpha = accuracy$, $\beta = accuracy$ and $w = 0.5$

A.3 Feature type results

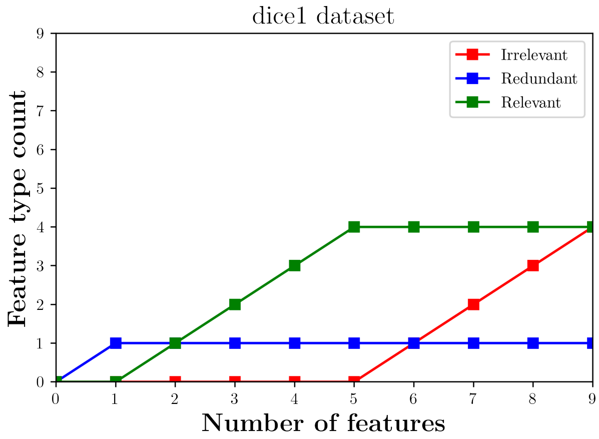


FIGURE A.38: Feature type count of dice1 dataset using Feature graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.2$

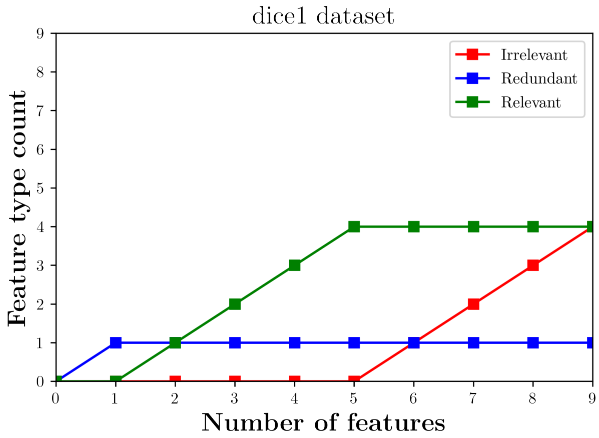


FIGURE A.39: Feature type count of dice1 dataset using Feature graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.5$

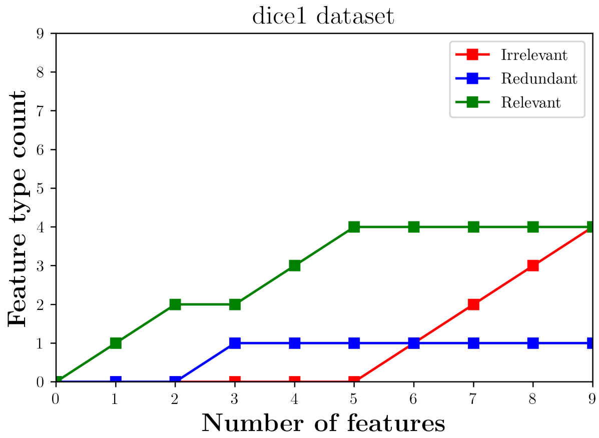


FIGURE A.40: Feature type count of dice1 dataset using Feature graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.8$

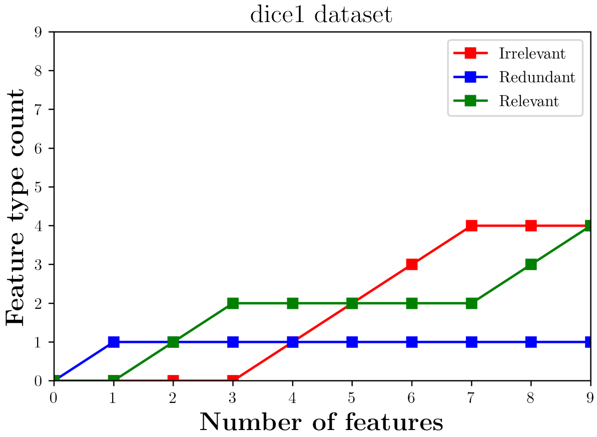


FIGURE A.41: Feature type count of dice1 dataset using Feature graph, α = *spearman_correlation*, β = *spearman_uncorrelation* and $w = 0.2$

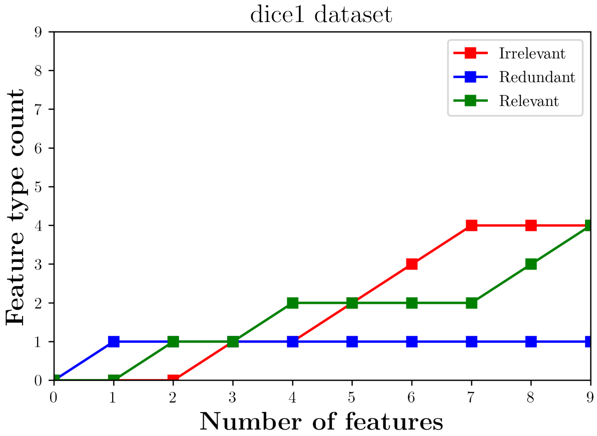


FIGURE A.42: Feature type count of dice1 dataset using Feature graph, α = *spearman_correlation*, β = *spearman_uncorrelation* and $w = 0.5$

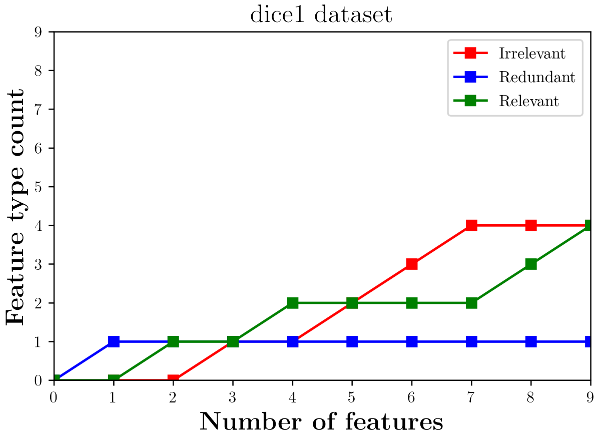


FIGURE A.43: Feature type count of dice1 dataset using Feature graph, α = *spearman_correlation*, β = *spearman_uncorrelation* and $w = 0.8$

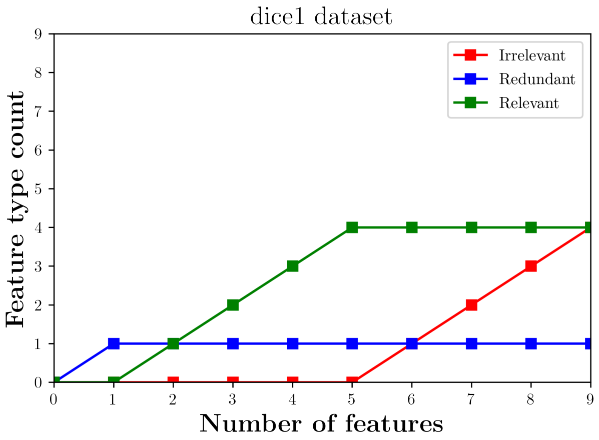


FIGURE A.44: Feature type count of dice1 dataset using Feature+Target graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.2$

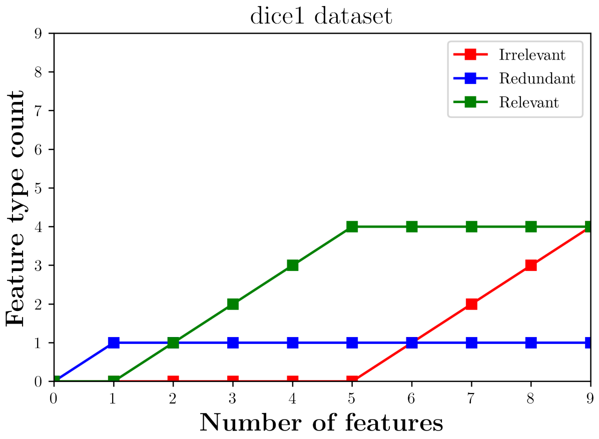


FIGURE A.45: Feature type count of dice1 dataset using Feature+Target graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.5$

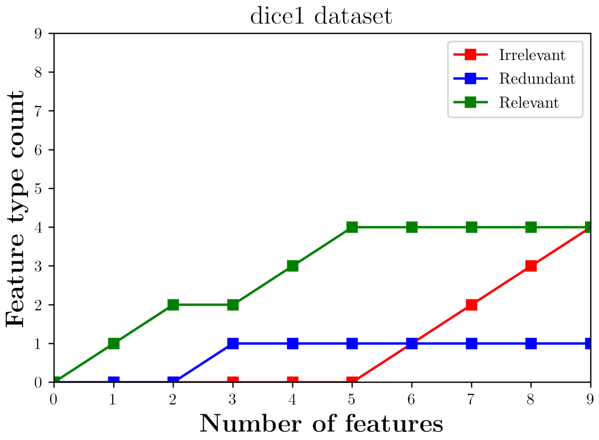


FIGURE A.46: Feature type count of dice1 dataset using Feature+Target graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.8$

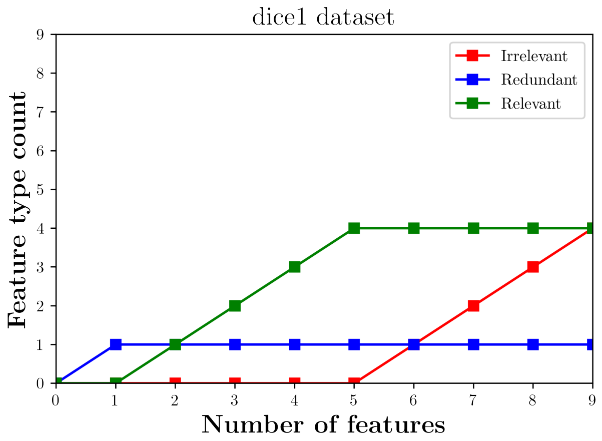


FIGURE A.47: Feature type count of dice1 dataset using Feature+Target graph, $\alpha = accuracy$, $\beta = accuracy$ and $w = 0.2$

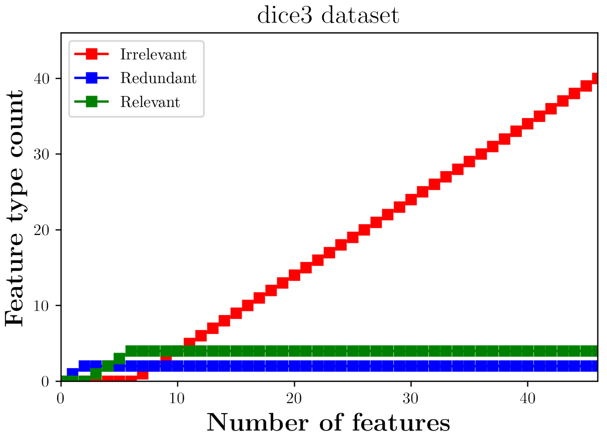


FIGURE A.48: Feature type count of dice3 dataset using Feature graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.5$

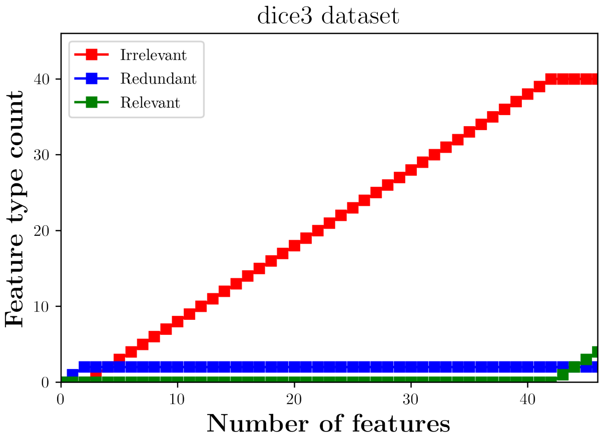


FIGURE A.49: Feature type count of dice3 dataset using Feature graph, $\alpha = spearman\_correlation$, $\beta = spearman\_uncorrelation$ and $w = 0.5$

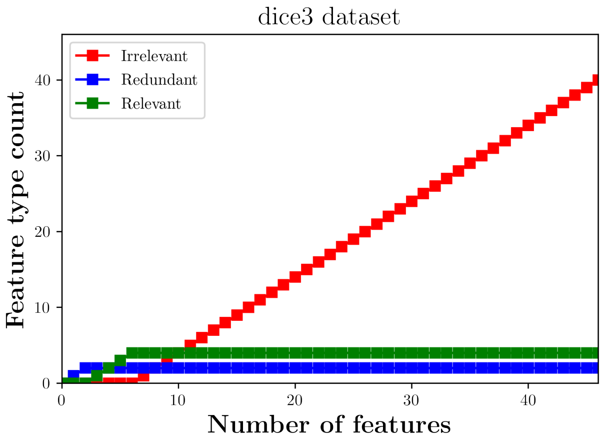


FIGURE A.50: Feature type count of dice3 dataset using Feature+Target graph, $\alpha = mutual\_information$, $\beta = uncorrelation$ and $w = 0.5$

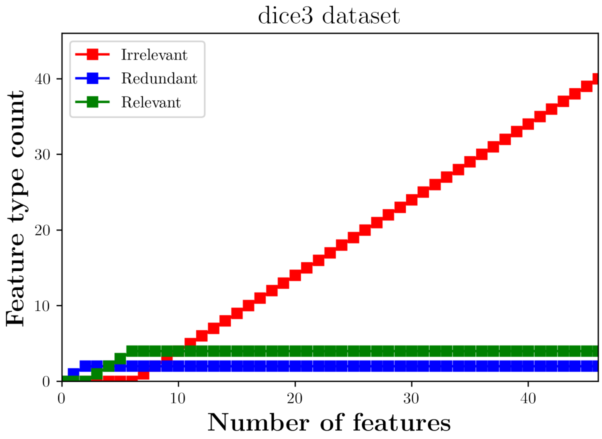


FIGURE A.51: Feature type count of dice3 dataset using Feature+Target graph, $\alpha = accuracy$, $\beta = accuracy$ and $w = 0.5$

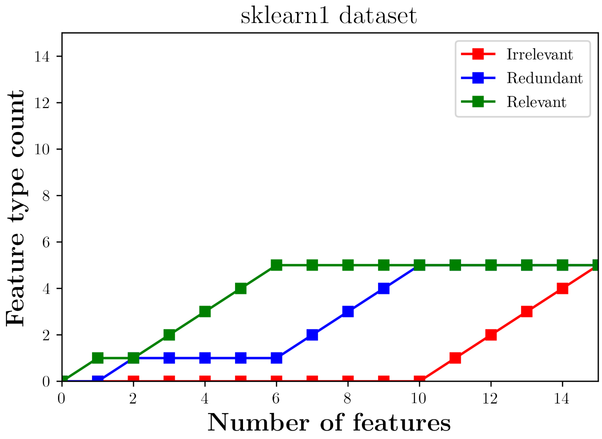


FIGURE A.52: Feature type count of sklearn1 dataset using Feature graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.2$

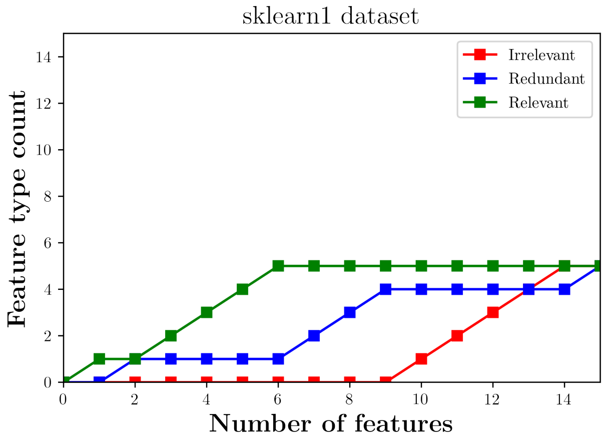


FIGURE A.53: Feature type count of sklearn1 dataset using Feature graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.5$

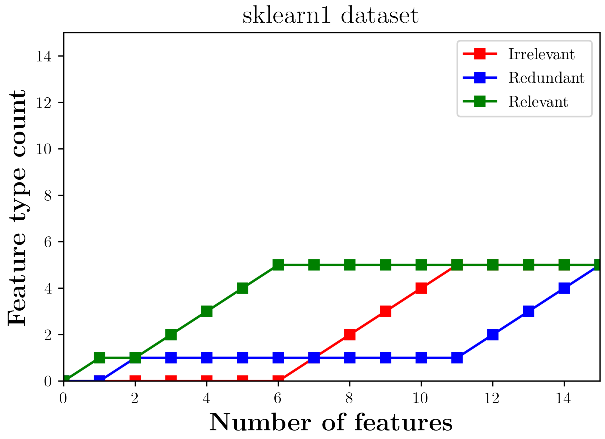


FIGURE A.54: Feature type count of sklearn1 dataset using Feature graph, $\alpha = spearman\_correlation$, $\beta = spearman\_uncorrelation$ and $w = 0.2$

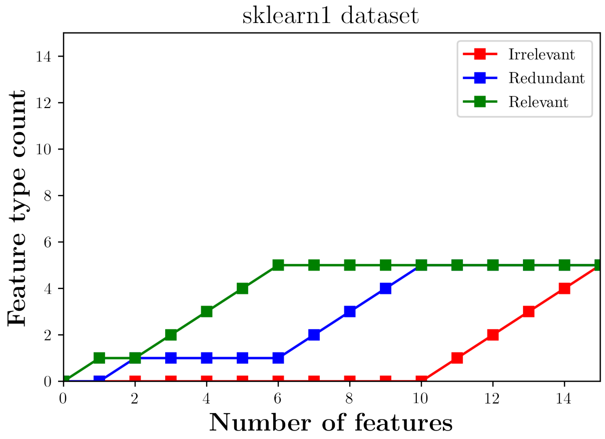


FIGURE A.55: Feature type count of sklearn1 dataset using Feature+Target graph, $\alpha = accuracy$, $\beta = accuracy$ and $w = 0.2$

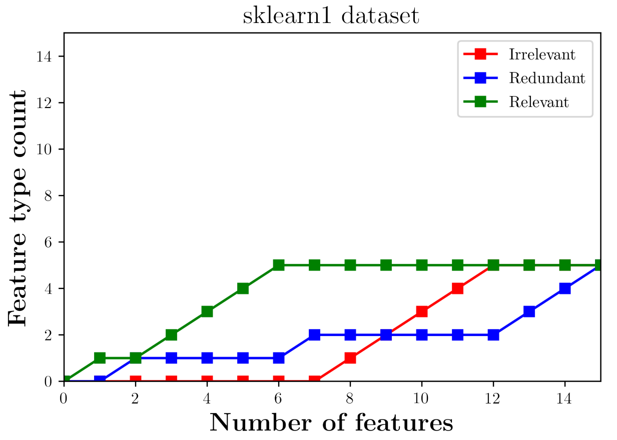


FIGURE A.56: Feature type count of sklearn1 dataset using Feature+Target graph, $\alpha = mutual\_information$, $\beta = uncorrelation$ and $w = 0.2$

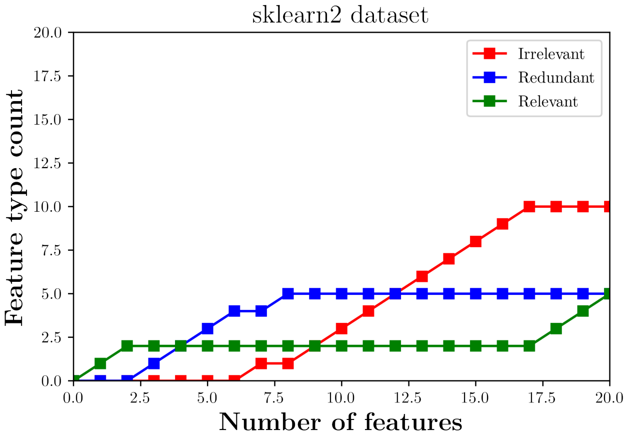


FIGURE A.57: Feature type count of sklearn2 dataset using Feature graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.2$

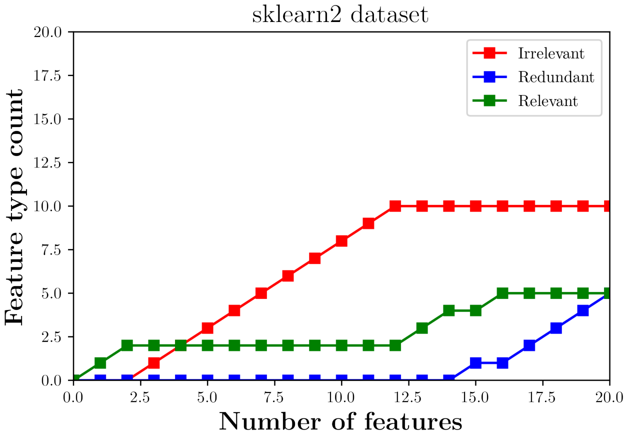


FIGURE A.58: Feature type count of sklearn2 dataset using Feature graph, $\alpha = spearman\_correlation$, $\beta = spearman\_uncorrelation$ and $w = 0.2$

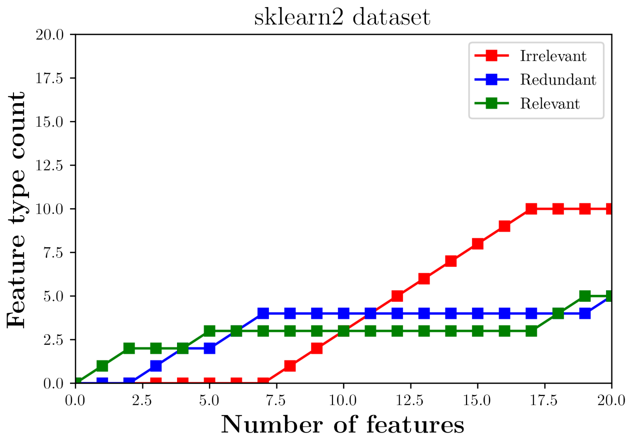


FIGURE A.59: Feature type count of sklearn2 dataset using Feature+Target graph, $\alpha = mutual\_information$, $\beta = uncorrelation$ and $w = 0.2$

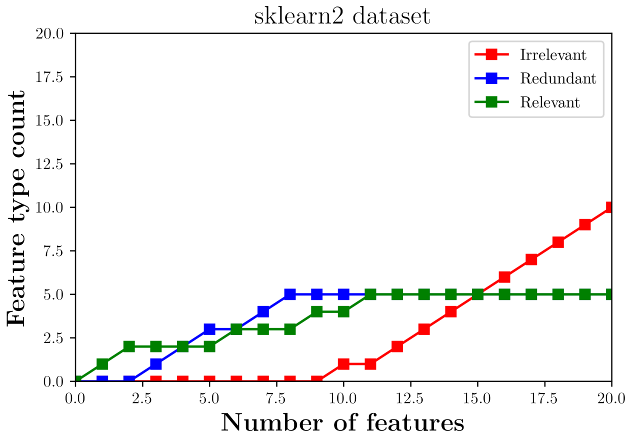


FIGURE A.60: Feature type count of sklearn2 dataset using Feature+Target graph, $\alpha = accuracy$, $\beta = accuracy$ and $w = 0.2$

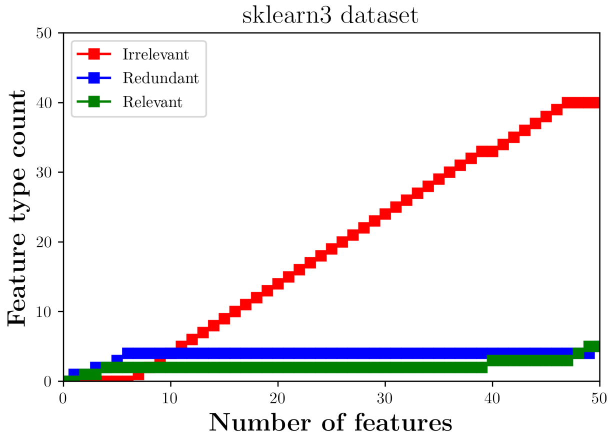


FIGURE A.61: Feature type count of sklearn3 dataset using Feature graph, $\alpha = correlation$, $\beta = uncorrelation$ and $w = 0.5$

A.4 Execution time results

Dataset	Graph type	Alpha function	Beta function	Weight	Execution time (s)
dice1	Feature	Correlation	Uncorrelation	0.2	0.0342
dice1	Feature	Correlation	Uncorrelation	0.5	0.0352
dice1	Feature	Correlation	Uncorrelation	0.8	0.0349
dice1	Feature	Spearman	Spearman	0.2	0.0375
dice1	Feature	Spearman	Spearman	0.5	0.0379
dice1	Feature	Spearman	Spearman	0.8	0.0328
dice1	Feature+Target	Correlation	Uncorrelation	0.2	0.0236
dice1	Feature+Target	Correlation	Uncorrelation	0.5	0.0195
dice1	Feature+Target	Correlation	Uncorrelation	0.8	0.0186
dice1	Feature+Target	Accuracy	Accuracy	0.2	0.1667
dice1	Feature+Target	Accuracy	Accuracy	0.8	0.1760
dice2	Feature	Correlation	Uncorrelation	0.2	0.0758
dice2	Feature	Spearman	Spearman	0.5	0.0659
dice2	Feature+Target	MI	Uncorrelation	0.5	0.1775
dice2	Feature+Target	Accuracy	Accuracy	0.5	0.4712
dice3	Feature	Correlation	Uncorrelation	0.5	0.5963
dice3	Feature	Spearman	Spearman	0.5	0.5472
dice3	Feature+Target	MI	Uncorrelation	0.5	0.6889
dice3	Feature+Target	Accuracy	Accuracy	0.5	4.2872
sklearn1	Feature	Correlation	Uncorrelation	0.2	0.0759
sklearn1	Feature	Correlation	Uncorrelation	0.5	0.0839
sklearn1	Feature	Spearman	Spearman	0.2	0.0973
sklearn1	Feature+Target	Accuracy	Accuracy	0.2	0.4919
sklearn1	Feature+Target	MI	Uncorrelation	0.2	0.1748
sklearn2	Feature	Correlation	Uncorrelation	0.2	0.1266
sklearn2	Feature	Spearman	Spearman	0.2	0.1273
sklearn2	Feature+Target	MI	Uncorrelation	0.2	0.2312
sklearn2	Feature+Target	Accuracy	Accuracy	0.2	0.8761
sklearn3	Feature	Correlation	Uncorrelation	0.2	0.6190
heart	Feature	Correlation	Uncorrelation	0.2	0.0558
heart	Feature	Spearman	Spearman	0.5	0.0514
heart	Feature+Target	MI	Uncorrelation	0.2	0.0716
heart	Feature+Target	MI	Uncorrelation	0.5	0.0775
heart	Feature+Target	Accuracy	Accuracy	0.2	0.3057
heart	Feature+Target	Accuracy	Accuracy	0.5	0.2999
titanic	Feature	Correlation	Uncorrelation	0.2	0.0172
titanic	Feature	Spearman	Spearman	0.2	0.0207
titanic	Feature	Spearman	Spearman	0.5	0.0197
titanic	Feature+Target	MI	Uncorrelation	0.2	0.0486
titanic	Feature+Target	MI	Uncorrelation	0.5	0.0511
titanic	Feature+Target	Accuracy	Accuracy	0.2	0.1201
titanic	Feature+Target	Accuracy	Accuracy	0.5	0.1194

TABLE A.32: Execution time results table

Bibliography

[1] Expert.ai Team. “What is Machine Learning? A Definition”. In: (2020). URL: https://www.expert.ai/blog/machine-learning-definition/.

[2] Sidath Asiri. “Machine Learning Classifiers”. In: (2018). URL: https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623.

[3] Wikipedia. “Feature selection”. In: (). URL: https://en.wikipedia.org/wiki/Feature_selection.

[4] S. Brin and L. Page. “The Anatomy of a Large-Scale Hypertextual Web Search Engine”. In: (1998). URL: http://infolab.stanford.edu/~backrub/google.html.

[5] Wenpu Xing and Ali Ghorbani. “Weighted PageRank Algorithm”. In: (2004). URL: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.92.6452&rep=rep1&type=pdf.

[6] David F. Gleich. “PageRank beyond the Web”. In: (2014). URL: https://arxiv.org/abs/1407.5107.

[7] Ken Schwaber and Jeff Sutherland. “The 2020 Scrum Guide”. In: (2020). URL: https://scrumguides.org/scrum-guide.html.

[8] Cornellius Yudha Wijaya. “What it takes to be correlated”. In: (2020). URL: https://towardsdatascience.com/what-it-takes-to-be-correlated-ce41ad0d8d7f.

[9] Juhi Ramzai. “Clearly explained: Pearson V/S Spearman Correlation Coefficient”. In: (2020). URL: https://towardsdatascience.com/clearly-explained-pearson-v-s-spearman-correlation-coefficient-ada2f473b8.

[10] Aerd statistics. “Spearman’s Rank-Order Correlation”. In: (2020). URL: https://statistics.laerd.com/statistical-guides/spearmans-rank-order-correlation-statistical-guide.php.

[11] Pablo Aznar. “What is Mutual Information?” In: (2021). URL: https://quantdare.com/what-is-mutual-information/.

[12] Scikit Learn. “Make Classification function”. In: (2015). URL: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_classification.html#sklearn.datasets.make_classification.

[13] Glassdor. “Glassdor”. In: (2021). URL: https://www.glassdoor.es/.

[14] Selectra. “Tarifaluzhora”. In: (2021). URL: https://tarifaluzhora.es/.