UNIVERSITY OF PIRAEUS

DEPARTMENT OF DIGITAL SYSTEMS
POSTGRADUATE PROGRAMME IMFORMATION SYSTEMS AND SERVICES
Specialization: Big Data and Analytics

THESIS

DIABETES DIAGNOSIS USING MACHINE LEARNING

Eleni Mamandra

Supervisor: Andriana Prentza, Professor

PIRAEUS

FEBRUARY 2022

THESIS

DIABETES DIAGNOSIS USING MACHINE LEARNING

Eleni Mamandra

Student ID: ME1933

ΠΕΡΙΛΗΨΗ

Η ανάπτυξη της τεχνολογίας και της επιστήμης των δεδομένων έχει προσφέρει σημαντική βοήθεια στον τομέα της ιατρικής και ακόμα περισσότερο στην διάγνωση νοσημάτων για την έγκαιρη και αποτελεσματική αντιμετώπισή τους.

Η παρούσα διπλωματική εργασία έχει ως στόχο τη μελέτη της διάγνωση του διαβήτη με χρήση τεχνικών μηχανικής μάθησης. Πιο συγκεκριμένα, αποτελεί μια μελέτη σύγκρισης αλγορίθμων μηχανικής μάθησης, ώστε να βρεθεί αυτός που προσφέρει την πιο ακριβή πρόβλεψη.

Για τον σκοπό αυτό, χρησιμοποιήθηκε μια βάση δεδομένων ευρέως γνωστή στην επιστημονική βιβλιογραφία για την πρόβλεψη διαβήτη, η βάση Pima Indians Diabetes, όπου περιέχει δεδομένα και μετρήσεις που βοηθούν στην ανίχνευση του διαβήτη τύπου 2.

Χρησιμοποιώντας την παραπάνω βάση δεδομένων, πραγματοποιήθηκε μοντελοποίηση με τη χρήση πλήθος κατάλληλων αλγορίθμων καθώς και υπολογισμός ικανών μέτρων αξιολόγησης για την σύγκρισή τους, με τελικό στόχο την ανάδειξη του βέλτιστου αλγορίθμου.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Διάγνωση της νόσου του διαβήτη

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Διάγνωση διαβήτη, PIMA Indians, επιβλεπόμενη μηχανική μάθηση

ABSTRACT

The increase of technology and data science has provided significant development in the field of medicine and even more in the diagnosis of diseases for their timely and effective treatment.

The present thesis aims to study diabetes diagnosis using machine learning techniques. More specifically, it is a study focuses on comparing machine learning algorithms as to conclude to the one that provide to the most accurate prediction.

For this purpose, a database widely known in the scientific bibliography for the diagnosis of diabetes was used, the Pima Indians Diabetes database, which contains data and measurements that help detect type 2 diabetes.

Using the abovementioned database, modeling phase is performed with a number of appropriate algorithms and calculation of sufficient evaluation measures aiming to find out the optimal prediction algorithm.

SUBJECT AREA: Diabetes diagnosis

KEYWORDS: Diabetes diagnosis, PIMA Indians dataset, supervised machine learning category

ACNOWLEDGEMENTS

I would like to thank Ms. Andriana Prentza, for her cooperation and valuable contribution with her comments and guidance to the completion of the present thesis.

I would like also to thank my fellow student Eirini Angelopoulou for our cooperation as well as the support and knowledge that she offered to me throughout the duration of this master’s program.

TABLE OF CONTENTS

List of Tables

List of Diagrams

List of Images

1. INTRODUCTION

1.1. Problem definition

Machine Learning (ML) focuses on the creation of algorithms that can learn rules from data, adapt to changes and improve their performance over time. ML has become critical as computers are anticipated to handle increasingly complicated issues and become more interwoven into everyday life. [1]

In the medical field, ML plays a critical role in recognizing the patient's disease, monitoring his health, and suggesting preventative measures given the difficulty to diagnose diseases manually. It may vary from small illnesses to serious illnesses like cancer, which is difficult to detect in early stages. [2]

In diabetes research, ML and data mining technics are used to extract significant knowledge about the disease from a large amount of data. Diabetes has substantial social repercussions in addition to its effects on individual level. Diabetes is a key priority in medical research because of its substantial socioeconomic consequences, which unavoidably generates a large amount of data.

The specific thesis is a ML study, aiming to identify diabetic and non-diabetic type 2 persons from a given database consisting of characteristics easily measured. It can be attributed to classification problem category, aiming to classify diabetic and non-diabetic persons. The algorithms applied in the classification modelling, have been compared and the most efficient has been presented.

1.2. Structure of the thesis

The structure of the thesis consists of nine chapters, where each one contributes to a better understanding of the final results of the modelling phase.

In the first chapter, an introduction to the thesis is performed. It consists of the problem definition, the structure and the contribution of the thesis.

Subsequently, in the second chapter, a description of diabetes disease is presented. Additionally, in order to emphasize the significance of the specific disease, which is concerning a great percentage of adults and infants, some statistics in a worldwide level (EU and US) are listed, taken from reputable health organizations.

The third chapter is referring to ML theories based on main problem categories that can be implemented with ML methods, as well as ML categories based on the training method. Furthermore, ML models that fall under supervised learning training method will be presented.

In the fourth chapter, measures for the evaluation of classification problems, will be introduced.

In the fifth chapter, the most commonly used database with respect to diabetes prediction will be presented, in the context of variables included and categorized based on the type of the variable (numerical or categorical).

In the sixth chapter, some existing studies that tackle the diabetes problem are presented. In more details, four studies that use different approaches and models are described and presented along with their evaluation measure results.

In the seventh chapter, an extensive analysis of the prediction methodology followed in the specific thesis is introduced. More specifically, the models were trained with respect two techniques for performance evaluation and under the next three categories:

i. without using any oversampling method
ii. by using Synthetic Minority Over-Sampling Technique (SMOTE) oversampling method
iii. by using Adaptive Synthetic Sampling approach for imbalanced learning (ADASYN) oversampling method

for the following ML classification algorithms:

i. Decision Tree,
ii. Random Forest,
iii. Support Vector Machine (SVM),
iv. K-Nearest Neighbor (KNN),
v. Logistic Regression and

vi. Naive Bayes

In more details, the two evaluation techniques that are used for the diabetes prediction are the train-test split and the cross validation (CV). Additionally, python programming language, which is used for the prediction purposes is described in a nutshell. Moreover, it is given an extensively description of the database preparation, with respect to statistical analysis and preprocessing methods followed, such as missing values and outliers handling. Finally, the advantages that contributed to the selection of the algorithms are introduced, as well as any tuning parameter applied, in order to increase accuracy levels.

In the eighth chapter the diabetes prediction results are cited along with detailed comments for the modelling evaluation measures of each category and algorithms that were performed.

Finally, in the ninth chapter, all findings from the modelling phase are presented, as well as suggestions are made for further investigation of the problem of diabetes prediction.

1.3. Contribution of the thesis

The given thesis contributes to the prediction of diabetes disease, which concerns more and more lately, and can influence further engagement for better results' achievements. To understand the significance of this thesis, we can observe the worldwide statistics for diabetes, which is associated with a variety of complications. In few words, these complications include increased risk of cardiovascular disease, including ischemic heart disease, stroke and increased hospitalization rates.

Furthermore, based on the purpose of the thesis to identify the best possible learning model, it is aimed that this model may contribute positively to the prediction of diabetes suspicion before the actual diagnosis, with high predictive performance.

By combining the power of ML and the importance of medical prediction and treatment in plenty of diseases, it is undoubted that ML field of research may lead to a better living for a significant amount of people.

2. DIABETES

2.1. Introduction

According to the Center for Disease Control and Prevention of the United States of America [3] diabetes is a long-term illness that affects the way the body converts food into energy. The majority of the food someone eats is converted to sugar (also known as glucose) and absorbed into bloodstream. When the blood sugar levels rise, pancreas is prompted to release insulin. Insulin is a key that allows blood sugar to enter body's cells and be used as energy. If somebody have diabetes, the body either does not produce enough insulin or does not utilize it as effectively as it should. Great concentration of blood sugar persists in the bloodstream when there isn't enough insulin or when cells stop responding to insulin. Over time, this can lead to serious health problems like heart disease, vision loss and renal illness.

Although there is currently no cure for diabetes, losing weight, eating healthy meals, and exercising may seem helpful. Taking medication as needed, receiving diabetes self-management education and support, and keeping health-care appointments can all help to lessen the impact diabetes has on everyday life. [3]

2.2. Types of diabetes

Type 1, type 2, and gestational diabetes (diabetes while pregnant) are the three main kinds of diabetes.

Type 1 diabetes

Type 1 diabetes is thought to be caused by an autoimmune reaction (in which the body accidentally fights itself) that prevents the body from producing insulin. Type 1 diabetes affects approximately 5% to 10% of diabetics. Type 1 diabetes symptoms usually appear suddenly. Children, teens, and young adults are the most affected. Patients need to take insulin every day in case of type 1 diabetes. No one knows how type 1 diabetes can be avoided up to now. [3]

Type 2 diabetes

Type 2 diabetes specified when body can't use insulin properly and can't keep blood sugar at normal levels. The specific type of diabetes affects 90-95% of diabetics. It takes many years to develop and is usually diagnosed in adulthood (but more and more lately in children, teens, and young adults). Type 2 diabetes can be avoided or delayed by adopting a healthy lifestyle that includes decreasing weight, eating healthy foods and staying active. [3]

Gestational diabetes

Pregnant women who have never had diabetes referred to as gestational diabetes. If a pregnant woman has gestational diabetes, the baby may be more susceptible to health issues. Although gestational diabetes normally goes away after the baby is born, it raises the chance of developing type 2 diabetes later in life of the pregnant. Obesity is more prevalent in infant as a youngster or a teen, and type 2 diabetes is more common in later life. [3]

2.3. Statistics on diabetes for US and EU population

2.3.1. US population

The National Diabetes Statistics Report, a quarterly publication of the Centers for Disease Control and Prevention (CDC), contains data on diabetes and prediabetes prevalence and incidence, risk factors for complications, acute and long-term complications, deaths and costs. [4]

This report is an update of the 2017 National Diabetes Statistics Report and is aimed towards scientists and the results are presented below. [4]

With respect to diabetes, 34.2 million people in total have the disease, which relates to 10.5% of the total US population. In more details, 26.9 million (26.8 million adults) have been diagnosed with diabetes and 7.3 million people, or 21.4% of the population, fall under the undiagnosed category.

With respect to prediabetes, 88 million people aged 18 years or older have prediabetes which relates to 34.5% of the total adult US population. Additionally, 24.2 million aged 65 years or older have prediabetes.

2.3.2. EU population

In the European Union, around 32.3 million adults were diagnosed with diabetes in 2019, up to an estimated 16.8 million persons in 2000. In 2019, an estimated 24.2 million people in Europe would have diabetes yet would go untreated. Since 2000, the number of males diagnosed with diabetes has more than doubled, rising from roughly 7.3 million in 2000 to 16.7 million in 2019. Women with diabetes have also increased significantly, from 9.5 million in 2000 to 15.6 million in 2019, an increase of more than 50%. Men are more likely than women to get diabetes due to biological causes, and they must gain less weight to develop the disease. [5]

Diabetes affects a greater number of people as they get older: 19.3 million people aged 60-79 have diabetes in the EU, compared to 11.3 million people aged 40-59 and only 1.7 million people aged 20-39. While men are more likely than women to have diabetes in middle age (between 40 and 59 years old), women are more likely to develop diabetes after the age of 70, owing to their longer lives. [5]

In 2019, the average diabetes prevalence among adults (diagnosed and age-standardized) in EU countries was 6.2%. Rates ranged from more than 9% in Cyprus, Portugal, and Germany to less than 4% in Ireland and Lithuania. Diabetes prevalence appears to have leveled off in many European countries in recent years, particularly in the Nordic countries, however it has continued to rise modestly in Southern, Central and Eastern European countries. Part of the reason for these increased trends is the rise in obesity and physical inactivity, as well as its connections with population aging. [5]

3. MACHINE LEARNING METHODOLOGY AND MODELS

3.1. Introduction

The present chapter, is referring to ML theoretical background based on methodologies and category problems that can be handled with ML models. Firstly, the main problem categories that can be implemented with ML methods, as well as ML categories based on the training method will be presented. Lastly, ML models that fall under supervised learning category will be introduced.

3.2. Problem categories for machine learning implementation

There are several ML methods that can be applied, taking into consideration the type of the problem. In the next paragraphs different categories of problem are presented, which can be managed using ML approaches.

3.2.1. Classification problem

Classification refers to problems whose performance can only be one of a small number of predetermined groups. The classification problem group contains more complex problems whose results can be labeled as True/False or Yes/No. To be more specific, classification is a type of predictive modeling problem in which a class label is predicted for a given set of input data. [6]

In order to “train” the model, classification includes a training dataset with a large number of instances of inputs and outputs. The training dataset will be used to calculate how to major categorize examples of input data to particular class labels. As a result, the training dataset must be based on a particular problem and provide enough instances of each class label.

Class labels are commonly expressed as string values, such as "TRUE," "FALSE," and must be converted to numeric values before being used in modeling. Label encoding is the method of assigning a unique integer to each class label, such as "TRUE" = 0, "FALSE" = 1.

For classification predictive modeling problems, there are many different types of respective algorithms. When identifying algorithms into problem types, it is usually advised that the modeler conduct controlled experiments to determine which algorithm and algorithm configuration results is the best output for a given classification task. [7]

Taking into consideration the number of output classes, the problem can be fall in the next categories:

a. Binary Classification
b. Multi-Class Classification
c. Multi-Label Classification
d. Imbalanced Classification

a. Binary classification

Tasks of two class labels are referred to as binary classification models. Class label 0 is assigned to the normal state class, while class label 1 is assigned to the abnormal state class. A model that predicts a Bernoulli probability distribution for each example is widely used to model a binary classification task. [7]

The Bernoulli distribution is a discrete probability distribution that describes an event in which an occurrence has a binary outcome of 0 or 1. This means that the model estimates the likelihood of an example falling into class 1, or the abnormal state. [7]

Below are presented some ML algorithms that can be used for binary classification problems:

i. Logistic Regression
ii. KNN
iii. Decision Trees
iv. SVM
v. Naive Bayes

Some algorithms, such as Logistic Regression and SVM, are designed specifically for binary classification and do not support more than two classes by default.

b. Multi-class classification

Classification tasks with more than two class labels, are referred to as multi-class classification. Multi-class classification, unlike binary classification, does not differentiate between normal and abnormal outcomes. Instead, instances are allocated to one of a few pre-defined classes. [7]

On some problems, the number of class labels can be very large. In a face recognition system, for example, a model might predict that a picture belongs to one of thousands or tens of thousands of faces. Text translation model, for instance, is a type of multi-class classification problem that involves predicting a series of terms. Each word in the sequence of words to be predicted is divided into several classes, with the size of the vocabulary deciding the number of classes that can be predicted, which may be tens or hundreds of thousands of words. [7]

A model that predicts a Multinoulli probability distribution is commonly used to model a multi-class classification task. The Multinoulli distribution is a discrete probability distribution that describes an occurrence with a categorical outcome, such as K in 1, 2, 3,..., K. This means that the model estimates the likelihood of an example belonging to each class labels when it comes to classification. [7]

Many algorithms used for binary classification can be also used for multi-class classification such as:

i. KNN
ii. Decision Trees
iii. Naive Bayes
iv. Random Forest
v. Gradient Boosting

Multi-class problems can be solved using binary classification algorithms that have been adapted. This entails fitting multiple binary classification models for each class versus all other classes (known as one-vs-rest) or a single model for each pair of classes (called one-vs-one). A basic example of one of the above-mentioned techniques is shown below. [7]

<u>One-vs-Rest</u>: One binary classification model for each class is fitted versus all other classes.

<u>One-vs-One:</u> One binary classification model is fitted for each pair of classes.

Binary classification algorithms which can be performed including the above strategies for multi-class classification include:

1. Logistic Regression
2. SVM

c. Multi-label classification

Classification tasks with two or more class labels, where one or more class labels can be predicted for each case, are referred to as multi-label classification.

Multi-label classification tasks are often modelled with an algorithm that predicts several outcomes, each of which is predicted as a Bernoulli probability distribution. For each case, this is essentially a model that makes several binary classification predictions. [7]

Multi-label classification algorithms cannot be used explicitly with binary or multi-class classification algorithms. Specialized variants of regular classification algorithms, known as multi-label versions of the algorithms, can be used, for example [7]:

- Multi-label Decision Trees
- Multi-label Random Forests
- Multi-label Gradient Boosting

An alternative approach includes a separate classification algorithm for the prediction of the labels for each class.

d. Imbalanced classification

Imbalanced classification refers to dataset cases in which the number of examples in each class is allocated unequally.

In most of them, imbalanced classification tasks are binary classification tasks in which the majority of examples in the training dataset belongs to the normal class and the abnormal class has a minority of examples. [7]

Some of the common examples include:

- Fraud detection

- Outlier detection
- Medical diagnostic tests

While specialized techniques may be required, these problems are modeled as binary classification tasks. By undersampling the majority class or oversampling the minority class, specialized techniques may be used to adjust the composition of samples in the training dataset. In the below image 1, the two methods for imbalanced classification handling have been visualized, for a better overview of the modification of the preprocessed dataset.

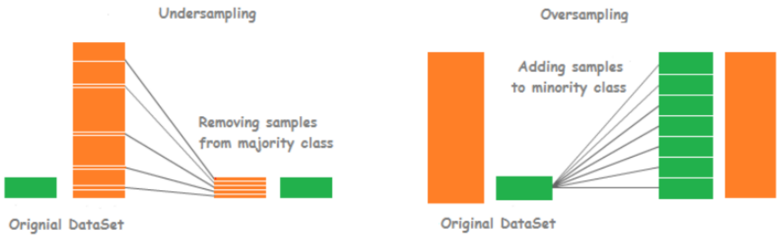
Undersampling
Removing samples from majority class
Orignial DataSet
Oversampling
Adding samples to minority class
Original DataSet

Image 1: Differences between undersampling and oversampling [8]

Some of the common examples include:

- *Random resampling*: Resampling is a commonly used technique for dealing with extremely unbalanced data sets. A simple solution is to delete samples from the majority class (undersampling) and/or add more examples from the minority class (oversampling).
- *Random undersampling*: This undersampling method entails eliminating events from the majority class at random. Instead of discarding events at random, data scientists can apply reasoning to determine which data should be kept and which should be discarded.
- *SMOTE oversampling*: The basic application of SMOTE is the generation of data for the minority class using existing data. A KNN classifier for this point is determined by randomly selecting a point from the minority class. Between the selected point and its neighbors, synthetic points are added. As a result, the decision function used by the algorithms during training may differs.
- *ADASYN Oversampling*: ADASYN focuses on creating samples next to the original samples, which have been incorrectly sorted, using a KNN classifier.

Specialized modeling algorithms, such as cost-sensitive ML algorithms, can be used to “pay extra attention” to the minority class when fitting the model on the training dataset.

3.2.2. Anomaly detection problem

Anomaly detection, which is also known as outlier detection, is the process of identifying unusual observations that raise suspicion by deviating greatly from the rest of the data. Typically, the anomalous objects will point to a problem such as bank fraud, a structural flaw, medical issues, or textual flaws. Outliers, novelties, noise, deviations, and exceptions are all different ways to define anomalies.

Anomaly detection techniques can be divided into three types. In an unlabeled test dataset, Unsupervised anomaly detection methods look for instances that seem to fit at least to the rest of the dataset, assuming that the vast majority of the dataset's cases are normal. A data set that has been categorized as "normal" and "abnormal" is necessary for supervised anomaly detection approaches, which requires to train a classifier. Semi-supervised anomaly detection strategies build a model depicting normal behavior from a given normal training data set and after assess the likelihood of the model being used to generate a test instance.

3.2.3. Regression problem

When dealing with problems that have both continuous and numeric performance, regression algorithms are used. These are typically used for problems involving questions such as “how much” or “how many”. [6]

Regression is a type of process used in statistical modeling to estimate the relationships between a dependent variable and one or more independent variables. Linear regression is the most common type of regression analysis.

3.2.4. Clustering problem

Clustering is defined as an unsupervised learning algorithm. Often known as cluster analysis, is the problem of categorizing a set of items so that objects in the same cluster are more related than those in other clusters. These algorithms attempt to learn structures within the data and create clusters based on data structure similarity. Clusters are groups with tiny distances between members, dense portions of the data space, intervals, or statistical distributions. As a result, clustering can be thought of as a multi-objective optimization problem. The best clustering algorithm and parameter settings (such as the distance function to employ, a density threshold, and the number of expected clusters) are determined by the data collection and the intended application of the results. [6]

The specific method is an iterative process of information discovery or interactive multi-objective optimization that involves trial and error, rather than an automatic activity. Usually, it depends to alter the data preparation and model parameters until the outcome has the desired attributes.

It is a common technique for statistical data analysis used in many domains, including pattern recognition, image analysis, information retrieval, bioinformatics, data compression, computer graphics, and ML, and is a main goal of exploratory data analysis.

3.3. Machine learning categories based on the training method

ML problems can be grouped into ten categories, as presented below, based on the algorithm's training method and the availability of the output while training. The sub-sections that follow describe each one of these concepts.

1. Supervised learning
2. Unsupervised learning
3. Semi-supervised learning
4. Reinforcement learning
5. Evolutionary learning
6. Ensemble learning
7. Artificial neural network
8. Instance-based learning
9. Dimensionality reduction algorithms and
10. Hybrid learning.

3.3.1. Supervised learning

Supervised learning is a ML category aimong to categorize data using a training dataset, which is a collection of instances used to train models. Each instance of supervised learning has an input set (usually a vector of attributes) and a desired output value. Supervised learning algorithms analyze training data to produce a model that can be used to generate new examples. The best-case scenario allows the algorithm to correctly tag unknown examples with the appropriate category tag. To accomplish this, the learning algorithm must generalize from unknown training data in a "logical" manner. [6]

3.3.2. Unsupervised learning

Unsupervised learning is a type of ML aiming to uncover a potential structure hidden behind unlabeled data. There is no error or reward signal to test potential solutions as long as the instances used are not numbered. Non-supervised learning differs from supervised and supportive (semi-supervised) learning in this aspect. [6]

3.3.3. Semi-supervised learning

This method of learning combines supervised and unsupervised learning techniques, in which supervised are used in a training set of unlabeled data. In other words, a small amount of classified data is combined with a huge amount of unlabeled data. [9]

When labeled data is paired with a small, labeled dataset, research has shown that the learning accuracy improves dramatically. These methods are most often found in web site text or images. [9]

3.3.4. Reinforcement learning

Reinforcement learning (RL) is a ML technique that considers how intelligent agents should behave in a given environment in order to maximize the concept of cumulative reward. RL along with supervised and unsupervised learning, is one of the three basic ML concepts. [6]

Reinforcement learning varies from supervised learning since it does not require the presentation of labelled input/output pairings or the explicit correction of sub-optimal behaviors. However, achieving a balance between exploration and exploitation is the main goal. The advantages of supervised and reinforcement algorithms can be combined with partially supervised reinforcement algorithms.

3.3.5. Evolutionary learning

Biological species that have the ability to adapt to their surroundings are the inspiration for evolutionary learning. The basic algorithm seems to understand the behavior of the inputs, allowing it to adjust and rule out unlikely solutions. In a broader sense, is aiming to match and suggest the most important solution to the problem. [6]

3.3.6. Ensemble learning

Ensemble methods are meta-algorithms that incorporate many ML techniques into one predictive model in order to reduce variance (bagging), bias (boosting), or enhance predictions (stacking). [6]

The approaches used in ensembles can be classified into two categories:

- Sequential ensemble approaches, where the base learners are produced in a sequential order (e.g. AdaBoost). The primary motivation for sequential approaches is to take advantage of the base learners' interdependence. By giving previously mislabeled examples more weight, the overall performance can be improved. [6]
- Parallel ensemble methods, where the base learners are generated in parallel (e.g. Random Forest). The main reason for parallel approaches is to take control of the independence between the base learners because averaging will drastically reduce error. [6]

Most ensemble methods employ a single base learning algorithm to generate homogeneous base learners, or learners of the same type, resulting in homogeneous ensembles.

Some approaches employ heterogeneous learners, or learners of various kinds, resulting in heterogeneous ensembles. Ensemble approaches must have as reliable and diverse base learners as possible in order to be more accurate than all of their individual participants.

3.3.7. Artificial neural network

Artificial neural networks (ANN) are encouraged by the biological neural network. A neural network is an interconnection of neuron cells that help the electric impulses to propagate through the brain. The basic unit of learning in a neural network is a neuron, which is a nerve cell. A neuron consists of four parts, namely dendrites (receptor), soma (processor of electric signal), nucleus (core of the neuron) and axon (the transmitting end of the neuron). Analogical to a biological neural network, an ANN works on three layers: input layer, hidden layer and output layer. This type of network has weighted interconnections and learns by adjusting the weights of interconnections in order to perform parallel distributed processing. The perceptron learning algorithm, back-propagation algorithm, hopfield networks, radial basis function network (RBFN) are some popular algorithms. [6]

3.3.8. Instance-based learning

Unlike other ML approaches that begin with a simple description of the target function derived from the training data, this process starts with no target function. Instead, it saves the training instance and awaits until a new instance is classified before generalizing. As a result, it is also known as the "lazy learner." When new data is introduced as input, these methods compare it to other instances in the database using a similarity measure to find the nearest match and make the prediction. [6]

The lazy learner estimates differently the target function and locally for every new instance to be classified instead of estimating it globally for the whole instance space. Hence it is faster to train but, takes time in making prediction. [6]

Some popular instance-based algorithms are k-means, k-medians, hierarchical clustering and expectation maximization.

3.3.9. Dimensionality reduction algorithms

Intelligent ML models have been used in a variety of sophisticated and data-intensive applications during the last few decades, including climatology, biology, astronomy, medicine, economics, and finance. Existing ML systems, on the other hand, are insufficiently efficient and extensible to deal with huge and voluminous data. Data with a high dimensionality has proven to be a curse when it comes to processing. The scarcity of data is another issue. Finding a global optimum for such data is time consuming. By lowering the number of dimensions in the data, a dimensionality reduction algorithm helps with the reduction of processing costs. It accomplishes this by eliminating redundant and irrelevant data and cleaning the data to increase the accuracy of the results. [6]

Dimensionality reduction searches exploits the latent structure in data in an unsupervised manner [10]-[11]. Principal component analysis (PCA), multidimensional scaling (MDS), principal component regression (PCR), and linear discriminant analysis (LDA) are some of the dimensionality reduction technologies that can be combined with classification and regression procedures. [6]

PCA is the most well-known linear dimensionality reduction approach.

PCA's key goal is to reduce the size of a data set while maintaining as much original data uncertainty as possible. A linear combination of data makes up the principal components (PC) of PCA. Since they are unrelated, they are perpendicular to the vector space. The first component represents the largest percentage of the original data's variance, and its geometric interpretation is related to the projection of the original data into a new space that best reflects the data's variance. The components that follow explain the data's initial variation to a lesser degree, with the latter providing no additional detail. It's worth mentioning that the PCA algorithm returns the key components in descending order of the percentage of variance they represent in the original results.

3.3.10. Hybrid learning

Though ensemble learning appeared as a relief to researchers dealing with the common problems of computational complexity, over fitting and sticking to local minima in classification algorithms, they have found problems with the specific method. It's hard to enforce and interpret the results because of the complicated ensemble of multiple classifiers. Ensembles may increase error at the level of individual base learners rather than improving model accuracy. Because of the combination of weak classifiers used in ensembles, accuracy can suffer. Hybridization, or the development of an ensemble of heterogeneous models, is a recent approach dealing with such issues. In this case, several methods are combined, such as clustering and decision tree or clustering and association mining. [6]

3.4. Supervised machine learning algorithms

3.4.1. Support vector machine

SVM is a ML algorithm that fall into supervised learning category. It is used to recognize subtle patterns through complex datasets. It can be used for both regression and classification tasks, but it is widely used in classification problems. More specifically, it performs discriminative classification, learning by example with respect to predict classifications of previously unseen data. It is commonly performed in fields such as text categorization, image recognition and more recently has been applied in numerous bioinformatics domains. [12]

There are four primary factors that corroborate to the popularity of the SVM algorithm. Firstly, it has a strong theoretical basis, based upon the dual ideas of VC dimension and structural risk minimization [13]. Secondly, SVM is performing well in relatively large datasets. Thirdly, it is quite flexible algorithm as it can be observed from different fields and cases applied, part of which are mentioned above. The algorithm's robustness, as well as the parameterization via a vast class of functions known as kernel functions, allow for this flexibility. In order to incorporate prior knowledge of a classification task, the behavior of the SVM can be modified through the modifications of the kernel function. Finally, the most important key of the popularity of the SVM is that produces significant accuracy with less computation power. [12]

The SVM algorithm's goal is to find a hyperplane in an N-dimensional space (N — the number of characteristics) that categorizes the data points clearly. There are various hyperplanes from which to choose to divide the two classes of data points. It is aimed to be determined a plane with the maximum margin, or the maximum distance among data points from both classes. Maximizing the margin distance gives some reinforcement, making it much easier to classify subsequent data points. [14]

Support vectors are data points that are closest to the hyperplane and have an impact on the hyperplane's position and orientation. The classifier's margin is maximized by using these support vectors. [14]

In SVM, if the output of the linear function is greater than 1, it is identified with one class and if the output is -1, it is identified with another class. Since the threshold values are changed to 1 and -1 in SVM, this reinforcement range of values ([-1,1]) is obtained and acts as margin. [14]

SVM algorithm, is aiming to maximize the margin between the data points and the hyperplane. The loss function helps maximize the margin is hinges loss. [14]

$$c\big(x, y, f(x)\big) = \begin{cases} 0, & if\ y * f(x) \geq 1 \\ 1 - y * f(x), & else \end{cases} \quad (1)$$

The cost equals to 0 if the expected and actual variables have the same sign. If not, the loss value is computed and a regularization parameter is added to the cost function. The regularization parameter's goal is to strike a compromise between margin maximization and loss. After adding the regularization parameter, the cost functions look as below. [14]

$$min_w \lambda \left|\left|w\right|\right|^2 + \sum (1 - y_i < x_i, w >)_+ \ (2)$$

After the loss function is obtained, partial derivatives are taken with respect to the weights, to find the gradients. Using the gradients, the weights can be updated. [14]

$$\frac{\delta}{\delta w_k}\lambda\left|\left|w\right|\right|^2 = 2\lambda w_k \ (3)$$

$$\frac{\delta}{\delta w_k}((1 - y_i < x_i, w >)_+ = \begin{cases} 0, & if\ y_i < x_i, w > \geq 1 \\ -y_i x_{ik}, & else \end{cases} \quad (4)$$

When there isn’t a misclassification, that is the model correctly predicts the class of the data point and it only needs to update the gradient from the regularization parameter. [14]

$$w = w - a * (2\lambda w) \quad (5)$$

When there is a misclassification, that is the model makes a mistake on the prediction of the class of the data point, it needs to be included the loss along with the regularization parameter to perform gradient update. [14]

$$w = w + a * (y_i * x_i - 2\lambda w) \quad (6)$$

3.4.2. K-nearest-neighbor

KNN algorithm is a non-parametric classification method which is used both for regression and classification problems. In more details, is a method used for classifying objects based on closest training examples in the feature space. Additionally, is a type of instance-based learning, or lazy learning. [15]

When there is little or no prior knowledge about the distribution of the data, the KNN is the most basic and simplest classification algorithm. As a result, if a sample's classification is unknown, it can be predicted by looking at the classification of its closest neighbors. All the distances between an unknown sample and all the samples in a training set can be computed given an unknown sample and a training dataset. With respect to input, it compromises of K closest training examples in a dataset. As regards output, it depends on the problem’s category. [15]

More specifically:

- In classification, the output is a class membership. Neighbors of each object vote and the classification is based on the popularity of the votes. The object is been assigned to the most common class between its k nearest neighbors. Parameter k is a positive and typically small integer. The nearest neighbor rule (NN) is the simplest form of KNN when K = 1, where K=1 indicates that object is classified similarly to its surrounding samples. [15]

- In regression, the output is the property value of the object, where this value is the average values of the k nearest neighbors. [15]

KNN algorithm is a classification type that the function is approached locally, and all calculations postponed until the evaluation of the function. Taking into consideration that the algorithm is based on distance for the classification part, if the features are being part of different physical units or come in vastly different scales then normalizing the training data can improve dramatically its accuracy. [15]

Both for classification and regression, a useful technique can be considered the assignment of the weights to the neighbor's contribution, in order the closest neighbors to contribute more than the distant ones in the average. For example, a common scheme consists in giving each neighbor a weight of 1/d, where d is the distance to the neighbor. [15]

The neighbors are taken from a set of objects for which the class (for KNN classification) or the object property value (for KNN regression) is known. Although no explicit training phase is necessary, this can be considered as of the algorithm's training set. [15]

The KNN algorithm has a peculiarity that it is sensitive to the data's local structure. [15]

3.4.3. Artificial neural network

ANNs are huge parallel computing models that mimic the human brain's function. An ANN is made up of a large number of basic processors connected by weighted connections. The processing nodes could be referred to as "neurons" by analogy. Each node's output is solely based on data that is locally available at the node, either stored internally or received via weighted connections. Each unit takes input from a large number of other nodes and sends its output to yet another set of nodes.

A single processing element isn't particularly powerful in and of itself; it produces a scalar output with a single numerical value that is a basic non-linear function of its inputs. [16]

The difference between the desired response and the system output is the error. This erroneous data is supplied back to the system, which uses it to alter the system settings in a systematic manner (the learning rule). The method is repeated until the results are satisfactory. This description makes it obvious that the performance is strongly reliant on the data. Neural network technology is probably not the best solution if data that covers a major amount of the operating circumstances is insufficient or if data is noisy. On the other hand, neural network technology is an excellent choice if there is plenty of data and the problem is too complicated as to develop an approximation model. [16]

The ANN does not “solve” the problem in a strictly mathematical sense, but it does display information processing qualities that provide a rough solution to a specific problem. ANNs have been widely employed in a variety of applications, including difficult non-linear function mapping, image processing, pattern identification and classification, and so on. Feed-forward networks are types of neural network that is commonly used. A feed forward network consists of an input layer that receives the problem's inputs, hidden layers that determine and reflect the relationship between the inputs and outputs using synaptic weights, and an output layer that emits the problem's outputs. Three basic pieces are used to model the neural feed forward network: a) a collection of synapses with synaptic weights, b) a summation adder or linear combiner for the input signals and c) an activation function that limits the amplitude of a neuron's output to a fixed value. Using a bias term, the activation function's input can be enhanced. [16]

3.4.4. Multifactor dimensionality reduction

Ritchie et al. [17] were the first to describe the Multifactor dimensionality reduction (MDR) approach. The key idea of original MDR method was to reduce the dimensionality of multi-locus data by pooling genotypes from different loci into high-risk and low-risk categories, resulting in a one-dimensional variable.

To evaluate its ability to identify and forecast disease status, CV and permutation testing are used. The data is divided into k approximately equal parts for CV. The MDR models are created for each of the k-1/k possible individuals (training sets) and then applied to the remaining 1/k individuals (testing sets) to make disease status predictions. [18]

Three steps can describe the core algorithm [18]:

i. Select d factors, either genetic or discrete environmental, using l_i, i= 1,..., d, levels from a total of N factors.

ii. Reflect the selected factors in d-dimensional space within the current training set and calculate the case (n_1) to control (n_0) ratio $r_j = \frac{n_{1_j}}{n_{0_j}}$ in each cell c_j, j=1,…,$\prod_{i=1}^{d} l_i$

iii. If r_j exceeds any threshold T (e.g., T=1 for balanced datasets), label c_j as high risk (H), otherwise as low risk.

For each of the possible d-factor combinations, these three steps are repeated in all CV training sets. CV consistency (CVC), classification error (CE), and prediction error (PE) are used to test the models produced by the core algorithm. A single model is chosen for each d=1,…,N that minimizes the average classification error ($\overline{CE}$) across the CEs in the CV training sets on this stage. CE is characterized as the percentage of individuals in the training set who are misclassified. The CVC is determined by the number of training sets in which a given model has the lowest CE. This generates a list of the best models, one for each d value. The final model is chosen from among these best classification models by minimizing the average prediction error ($\overline{PE}$) across the PEs in the CV testing sets. The PE is characterized as the proportion of misclassified individuals in the testing collection, similar to how the CE is defined. A Monte Carlo permutation approach is utilized to determine statistical significance using the CVC. [17]

A balanced data collection is required for the Ritchie et al. [17] technique, with the same number of cases and controls and no missing values in either factor. To address the above problem, Hahn et al. [19] hat each factor be given an additional level for missing data. Velez et al. [20]-[20] discuss the issue of unbalanced data sets, regardless of their scale. They tested three methods to prevent MDR from emphasizing patterns that are important for the larger set:

1. over-sampling, which involves resampling the smaller set with replacement;
2. under-sampling, which involves randomly eliminating samples from the larger set; and
3. balanced accuracy (BA), which involves using a modified threshold.

The accuracy of a factor combination is calculated by the BA as (sensitivity and specifity)/2, rather than by (1-CE), so that errors in both groups are given equal weight regardless of their scale. The ratio of cases and controls in the entire data set is the modified threshold T_{adj}. Using the BA in conjunction with the modified threshold is recommended based on their findings. [17]

3.4.5. Decision tree

A decision tree is a type of ML algorithm that divides data into groups. Decision tree is a technique for approximating discrete valued target function which represents the learning function in the form of a decision tree [21].

On the basis of feature values, a decision tree classifies instances by sorting them from root to leaf nodes. Every branch indicates a possible value for that feature, whereas each node represents a choice (test condition) on an attribute of the instance. The decision node, which is the root node, is where an instance's classification begins. The tree traverses down along the edge that corresponds to the value of the result of feature test based on the value of node. In the sub-tree headed by the new node at the end of the previous edge, this process continues. Finally, the classification categories or the final decision are represented by the leaf node. When employing a decision tree, the focus is on determining which attribute at each node level is the best classifier. [6] For each node, statistical measures like information gain, Gini index, Chi-square, and entropy are calculated to determine its performance. [20] Decision trees are implemented using a variety of algorithms. Classification and Regression Tree (CART), Iterative Dichotomiser 3 (ID3), Automatic Interaction Detection (CHAID), Chi-Squared C4.5 and C5.0, and M5 are among the most popular. [6]

3.4.6. J48 classification algorithm

Decision tree algorithm aims to find out the way the attributes-vector behaves for a number of instances.

The J48 algorithm is an implementation of the C4.5 decision tree algorithm. The basic steps of the algorithm are presented below [22]:

i. Based on the tree representation as a leaf when instances are of the same class, the leaf is returned by labeling with the same class.
ii. For each attribute, the potential information is determined using a test on the attribute. The gain in knowledge that would arise from a test on the attribute is then determined.
iii. With respect to the current selection criterion, the best property is identified, and that characteristic is chosen for branching.

Next, the features of the algorithm are presented. Firstly, this algorithm handles both discrete and continuous attributes. The threshold values for handling continuous attributes are decided by C4.5. The specific values divide the data into two categories, those with attribute values below the threshold and those having more than or equal to it. As a note, the algorithm handles the missing values in the training data. Finally, after the full construction of the tree, the pruning of the tree is performed. After C4.5 construction, it drives back through the tree and challenges to remove branches that are not helping in reaching the leaf nodes.

3.4.7. Artificial metaplasticity on multilayer perceptron

Synaptic plasticity is a term used in biology, neuroscience, physiology, neurology, and other fields to describe memory and learning processes. It refers to a synapse's ability to modify its efficiency between two neurons. It is caused by a variety of cellular mechanisms that alter synaptic efficiency and can result in an increase in synaptic strength (Long-term Potentiation) or a decrease in synaptic strength (Long-term Depression). By metaplasticity property, reinforcement is lower when time-averaged level of postsynaptic firing is high than when time-averaged level of postsynaptic firing is low. [23]

Andina et al. [24] were the first to coin the term artificial metaplasticity (AMP). In an ANN, the method focuses on the degree of variation in artificial synaptic strength or weights. High post synaptic activity, according to Andina, must be linked to frequent excitations (frequent input classes in an artificial model). The statistical distribution of patterns is vital in defining the amount of variance in the artificial synaptic weight, just as the value of postsynaptic activation is crucial in determining the amount of variation in the biological synaptic strength. Then, patterns with a higher probability of occurring will have more frequent activations over time and, as a result, less reinforcement of artificial weights. As a result, during the "training phase" of an ANN's learning, the AMP assigns greater values for updating the artificial synaptic weights in the less likely patterns than in the more probable ones. More specifically, the AM as a probabilistic learning procedure produces greater modifications in the synaptic weights with less-frequent patterns than frequent patterns, leading to extract more information from the former than from the latter. [25]

The metaplasticity, indicates a higher level of plasticity, expressed as a change or transformation in the way synaptic efficacy is modified. Metaplasticity is defined as the induction of synaptic changes, that depends on prior synaptic activity. Although implementation has only been tried for the multilayer perceptron type (MLP), referred to as Artificial metaplasticity on multilayer perceptron (AMMLP), the AMP Concept is applicable to any ANN. [25]

AMP is analytically introduced in an arbitrary MLP training, by the application of a weighting function $w_X^*(x)$ [26]:

$$f_X^*(x) = \frac{A}{\sqrt{(2\pi)^N * e^{B\sum_{i=1}^{N} x_i^2}}} = \frac{1}{w_X^*(x)} \qquad (7)$$

where $w_X^*(x)$ is defined as $\frac{1}{f_X^*(x)}$, being $f_X^*(x)$ an approximation of the input patterns probability density function (pdf), N is the number of neurons in the MLP input layer, and parameters A and $B \in R^{+}$ are algorithm optimization values which depend on the specific application of the AMLP algorithm. [26]

As the pdf weighting function proposed is the distribution of the input patterns that does not depend on the network parameters, the AMMLP algorithm can then be summarized as a weighting operation for updating each weight in each MLP learning iteration as [26]:

$$\Delta^{*} w = w^{*}(x) \Delta w \quad (8)$$

3.4.8. Bayesian classifier

Naive Bayes classifiers can be described as members of "probabilistic classifiers" which are based on Bayes' theorem with strong (naive) independence assumptions between the features. In fact, there are between the simplest Bayesian networks models, which combined with kernel density estimation can achieve higher accuracy levels. [26]

According to Bayesian theorem, the probability of a set of data x_{t} belonging to c is [26]:

$$P\left(\frac{C}{X_t}\right) = \frac{p(C)p\left(\frac{X_t}{C}\right)}{p(X_t)} \quad (9)$$

where

- $p(C)$ is the prior probability of C: the probability that C is correct before the dataX_t are seen.
- $p(X_t/C)$ is the conditional probability of seeing the data X_t given that the hypothesis C is true. This conditional probability is called the likelihood.
- $p(X_t)$ is the marginal probability of X_t.
- $p(C/X_t)$ is the posterior probability: the probability that the hypothesis is true, given the data and the previous state of belief about the hypothesis.

Using the above formula, Bayes classifier calculates the conditional probability of an instance belonging to each class. Based on this conditional probability the instance is classified as the class with the highest conditional probability. It has the same interpretability as decision tree and can use previous data knowledge in order to build analysis model for future classification or prediction. [27]-[28]

If the eigenvalues of data are continuous, there are two ways to process [27]-[28]:

- Assume the distribution is normal, and find the means, variances of the eigenvalues as likelihood.
- Use splitting method to transfer continuous data into discrete data.

3.4.9. Logistic regression

Logistic regression is a modeling technique that can be used to represent the connection between multiple independent factors and a dichotomous dependent variable. As a result, it lends itself to categorizing a binary outcome such as the presence and occurrence of an illness or event, or the absence and nonoccurrence of a disease or event. [29]

The logistic function is a squashing function that converts an input with a value between 0 and 1 into an output with a value between 0 and 1. The function f(z) reflects the likelihood of a disease or event occurring.

$$f(z) = \frac{1}{1 \pm e^{-z}} \qquad (10)$$

where $z = b + w1p1 + w2p2 + \ldots wkpk$ is an index of combined risk factors. Since in the domain for z of + infinity, $f(z)$ ranges from 0 to 1, it can clearly be used to describe the probability or risk of an event or an occurring disease. [29]

4. EVALUATION MEASURES

When it comes to classification problems, the confusion matrix is a widely utilized tool. It can be used to solve problems involving binary and multiclass categorization. Table 10 illustrates an example of a binary classification confusion matrix.

Table 1: Confusion matrix for binary classification

Confusion Matrix		**Predicted**	
		Negative	**Positive**
Actual	**Negative**	*TN*	*FP*
	Positive	*FN*	*TP*

Confusion matrices are used to describe counts based on predicted and actual values. The output "TN" stands for true negative, and it displays the number of correctly identified negative cases. Similarly, "TP" stands for true positive, which denotes the number of correctly identified positive examples. "FP" stands for false positive value, which is the number of actual negative examples classified as positive, and "FN" stands for false negative value, which is the number of actual positive examples classified as negative. Accuracy is one of the most widely utilized criteria while performing categorization. Some of the evaluation indicators generated from a confusion matrix are presented below.

True positive rate (TPR) or recall or sensitivity:

The percentage of positive examples that are categorized correctly. The probability of a sample being classified as positive when it is actually positive (in this study, a diabetes patient) (Equation 9). The complementary sensitivity index is the false negative rate (FNR).

$$TPR = \frac{TP}{TP + FN} \quad (11)$$

True negative rate (TNR) or specificity:

The percentage of negative examples that are categorized correctly. It is the probability that a really negative sample will be classified as negative (in this study, a non-diabetic person) (Equation 10). The complementary indicator of specificity is the false positive rate (FPR).

$$TNR = \frac{TN}{TN + FP} \quad (12)$$

Precision (positive predictive value)

The percentage of the total number of positive predictions divided by the number of correct positive predictions. In other words, precision is the ratio of correct positive results to positive results predicted by the classifier. Positive predictive value (PPV) is an alternative name for precision.

$$PREC = \frac{TP}{TP + FP} \quad (13)$$

Accuracy

It is the percentage of control samples that were correctly categorized by the model (Equation 12).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (14)$$

F1 score

F1-score is the harmonic mean of precision and recall computed from the number of mispronunciations detected by both the computer and human evaluator. It is defined as [30]

$$F1 = \frac{2\ \times Precision \times Recall}{Precision + Recall} \quad (15)$$

Area under curve (AUC)

AUC as described from Bowers and Zhou [31] is a step further over ROC analysis, AUC is calculated by summing the area under the ROC curve, and the bigger the area, the more accurate the predictor. Formally, the formula for calculating AUC is

$$AUC = \int_0^1 f(x)dx \qquad (16)$$

5. DATABASE FOR DIABETES DIAGNOSIS

5.1. Introduction

By observing the bibliography with respect to diabetes diagnosis using machine learning methods, the majority converge in the use of a specific database, the Pima Indians database. Additionally, given that it is the most common database for diabetes prediction, it was chosen for the purposes of the present thesis. The specific database can be found in the next address: https://www.kaggle.com/uciml/pima-indians-diabetes-database, and is described below.

5.2. Pima diabetes database

The National Institute of Diabetes and Digestive and Kidney Diseases contributed this dataset, which was released publicly through the UCI ML repository.

The objective of the dataset is to use diagnostic measures to determine whether or not a patient has diabetes type 2. All of the patients at this center are Pima Indians women who are at least 21 years old. More specifically the data set contains 768 patients and includes 9 attributes:

- Pregnancies
- Glucose
- BloodPressure
- SkinThickness
- Insulin
- BMI
- DiabetesPedigreeFunction
- Age
- Outcome

The 9 attributes consisted of are described in more details below and categorized based on the type of the variable (numerical or categorical).

Table 2: Pima Indians database description

Variable	Description	Category	Type
Pregnancies	Number of times pregnant	Numerical	Integer
Glucose	Plasma glucose concentration a 2 hours in an oral glucose tolerance test	Numerical	Integer
BloodPressure	Diastolic blood pressure (mm Hg)	Numerical	Integer
SkinThickness	Triceps skin fold thickness (mm)	Numerical	Integer
Insulin	2-Hour serum insulin (mu U/ml)	Numerical	Integer
BMI	Body mass index (weight in kg/(height in m)^2)	Numerical	Float
DiabetesPedigreeFunction	A function that assesses the risk of diabetes according to a person's family history	Numerical	Float
Age	Age in years	Numerical	Integer
Outcome	Class variable, where 0 indicates non-diabetic person and 1 a diabetic patient	Categorical	Integer

6. PRESENTATION OF PREVIOUS STUDIES FOCUSED ON DIABETES DIAGNOSIS

6.1. Introduction

Several studies have been implemented in the field of diabetes diagnosis using ML algorithms. In the specific paragraph, some of the studies for diabetes diagnosis will be presented in the context of algorithms and parameters applied, as well as their performance evaluation. In general, all the studies presented have been use Pima Indians database, which was described previously.

6.2. First study

In the study, proposed by Aishwarya. R, Gayathri. P and N. Jaisankar [32], a system for the diabetes detection is developed as well as discloses the diabetes' compilations. The system that has been proposed is based on PCA with respect to the preprocessing techniques. The diagnosis of a patient having or not diabetes is using SVM algorithm.

The specific analysis, using SVM with PCA as preprocessing method shows important increase in accuracy metric. More detailed accuracy with this system achieved output of 95%.

6.3. Second study

Another approach has been followed in "Predictive modelling and analytics for diabetes using a ML approach" study [33]. In this study, five different models have been performed in order to detect diabetes in female patients, such as:

- linear kernel SVM (SVM-linear),
- radial basis kernel SVM (SVM-RBF),
- KNN,
- ANN and
- multifactor dimensionality reduction (MDR) algorithms.

In their proposed study, pre-processing method of raw data and different feature engineering techniques have been used, in order to obtain better results. More specifically, the pre-processing method refers to the removal of outliers and KNN imputation as to predict the missing values. For better outcomes, highly correlated variables have been used.

The next table shows the different ML methods trained on Pima Indians diabetes dataset with optimized tuning parameters.

Table 3: ML methods with optimized tuning parameters

S.No.	Model Name	Tuning Parameters
1	SVM-linear	C=1
2	SVM-RBF	C=1 and Sigma=0.107
3	KNN	K=13
4	ANN	Size=10
5	MDR	recode function to converts the value into 0, 1, and 2

Performance of all the five different models is evaluated based on parameters such as precision, recall, AUC and F1 score (Table 4). In order to avoid problem of over fitting and under fitting, 10-fold CV was applied. In the next table, the performance of all five models is presented.

Table 4: Performance scores of five ML models

S.No	Predictive Models	Accuracy	Recall	Precision	F1 score	AUC
1	SVM-linear	0.89	0.87	0.88	0.87	0.90
2	SVM-RBF	0.84	0.83	0.85	0.83	0.85
3	KNN	0.88	0.90	0.87	0.88	0.92
4	ANN	0.86	0.88	0.85	0.86	0.88
5	MDR	0.83	0.87	0.82	0.84	0.89

6.4. Third study

In the proposed study “Improved J48 Classification Algorithm for the Prediction of Diabetes” by Gaganjot Kaur and Amit Chhabra [34], classification algorithm modified J48 decision tree is used, in order to analyze the prediction of diabetes and increase the accuracy of the J48 decision tree method.

The evaluation of the model is based on accuracy, which for the proposed model achieved 99.87%.

6.5. Fourth study

In the study proposed by Marcano-Cedeno and Andina [35], a10-fold CV in classification model construction and efficiency evaluation is employed, and the classification algorithms adopted AMMLP, decision tree and Bayesian classifier.

As a preprocessing method of the database, all the observations with zero entries are removed. Only 763 instances remain in the study's data after deleting all of the aforementioned values and variables. Each dataset was randomly divided into training and testing datasets, with 60-40% training and testing sets per dataset.

To develop a classification model and evaluate classification efficiency, the 10-fold CV approach was used to input training and testing datasets into AMMLP, decision tree and Bayesian classifier, respectively. The same training and testing processes were repeated ten times in each attribute input mode. Finally, the average classification accuracy of each classification model was calculated.

The parameters used and results of the three models are summarized in the next tables.

Table 5: The network structure, metaplasticity parameters, epochs and MSE used in the training and testing phases of the classifiers AMMLP

Classifier	Network Structure			Metaplasticity Parameters		Mean Squared Error	Epoch	Accuracy
	I	HL	O	A	B			
AMMLP1	8	4	1	37	0.2	0.01	2000	89.93%
AMMLP2	8	7	1	38	0.3	0.01	2000	88.50%
AMMLP3	8	8	1	39	0.5	0.01	2000	87.85%

Table 6: Best results obtained by decision tree (J48), with different confidence factors

Classifier	Confidence factor	Minimum number objects	Accuracy
Decision Tree, DT1	0.25	2	77.60%
Decision Tree, DT2	0.1	2	72.08%
Decision Tree, DT3	0.6	2	74.68%

Table 7: Results obtained by Bayesian classifier with and without using kernel estimation

Classifier	Using kernel estimation	Accuracy
Bayesian classifier, BC1	False	78.57%
Bayesian classifier, BC2	True	78.51%

From the results presented above, it seems that the best accuracy for the estimation of diabetes obtained by AMMLP model, with 89.93%.

7. DIABETES DIAGNOSIS METHODOLOGY

7.1. Introduction

In this chapter, an extensive analysis of the diagnosis methodology of the present thesis will be presented. In general, taking into consideration the nature of the study, as a supervised binary classification problem, it was chosen to apply a sufficient number of algorithms that fall under the classification category. More specifically, the following ML classifiers will be used:

- Decision Tree
- Random Forest
- SVM
- KNN
- Logistic Regression
- Gaussian Naïve Bayes

To evaluate the classification results, the following indicators have been calculated:

- True Positive Rate
- False Negative Rate
- True Negative Rate
- False Positive Rate
- Accuracy

Given that the study is a medical detection focused on diabetes, it was considered that the optimal evaluation indicators is the TPR and the supplementary FNR. This decision is based on the idea that for a person under examination, it is preferable to properly categorize him as a patient, truly sick, for the best possible treatment of the disease. Additionally, accuracy can be misleading when employed with unbalanced datasets, such as PIMA dataset that is used in the current study and will be described in a following paragraph.

7.2. Prediction using Python programming language

As described from Diego de Sousa Miranda in [36], “Python is an interpreted, interactive, object-oriented programming language. It provides high-level data structures such as list and associative arrays (called dictionaries), dynamic typing and dynamic binding, modules, classes, exceptions, automatic memory management, etc. It has a remarkably simple and elegant syntax and yet is a powerful and general-purpose programming language. It was designed in 1990 by Guido van Rossum. Like many other scripting languages, it is free, even for commercial purposes, and it can be run on practically any modern computer. A python program is compiled automatically by the interpreter into platform independent byte code which is then interpreted.

Python is modular by nature. The kernel is very small and can be extended by importing extension modules. The Python distribution includes a diverse library of standard extensions (some written in Python, others in C or C++) for operations ranging from string manipulations and Perl-like regular expressions, to Graphical User Interface (GUI) generators and including web-related utilities, operating system services, debugging and profiling tools, etc. New extension modules can be created to extend the language with new or legacy code. We describe these extension capabilities below. There are a substantial number of extension modules that have been developed and are distributed by members of the Python user community”.

For the purposes of the he specific thesis, some of the most common libraries have been used for the database preparation and modelling phase such as “numpy”, “pandas”, “matplotlib”, “sklearn”.

7.3. Database preparation

In this paragraph, some basic descriptive statistics for the database used for diabetes diagnosis are presented. Furthermore, based on the descriptive analysis an extensive database preparation was performed, in order to tackle any issue observed.

The most significant factor is the data quality, as it has an impact on the quality of the analysis outcomes. Data should be collected, combined, characterized, and prepared for analysis carefully.

In the present thesis data preprocessing techniques were used, as to increase the quality of the result and the efficiency of the process.

With respect to the class variable, the distribution is presented in the below diagrams.

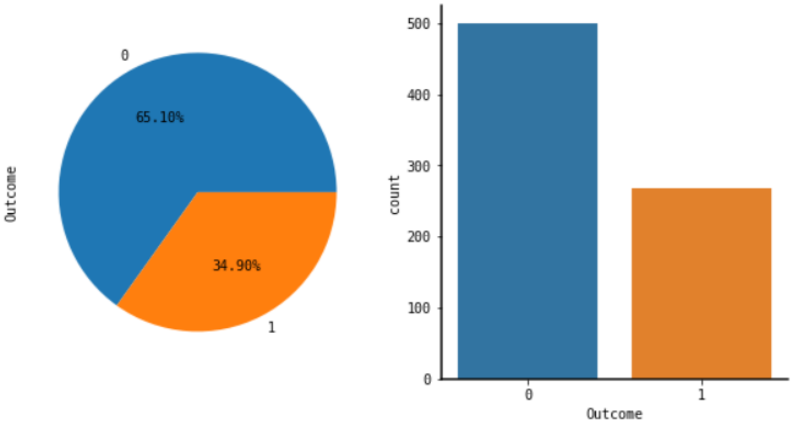

Diagram 1: Percentage and amount of diabetes cases in dataset

After observing the distribution of the “Outcome” value, an unequal distribution was found which may lead to the export of an invalid diagnosis in the remaining category of non-diabetes cases.

In table 8 basic descriptive statistic measures for dataset variables have been calculated and presented.

Table 8: Basic descriptive statistic measures for dataset variables

	count	mean	std	min	0.25	0.5	0.75	max
Pregnancies	768	3.85	3.37	0	1	3	6	17
Glucose	768	120.9	31.98	0	99	117	140.25	199
BloodPressure	768	69.11	19.36	0	62	72	80	122
SkinThickness	768	20.54	15.95	0	0	23	32	99
Insulin	768	79.8	115.24	0	0	30.5	127.25	846
BMI	768	31.99	7.89	0	27.3	32	36.6	67.1
DiabetesPedigree Function	768	0.47	0.33	0.078	0.24	0.3725	0.62625	2.42
Age	768	33.24	11.76	21	24	29	41	81

Variables do not appear to have any missing values. However, data must be examined thoroughly to avoid any problems in the training phase of ML algorithms, as inaccuracies in attribute values are widespread. This phenomenon is known as noise, and it occurs when data is collected via experimental measurements or when the human factor is involved in the development of training data in general. The presence of extensive noise is likely to drive the learning process off. If the assessment data contains noise of the same type, it must be evaluated.

To observe if there is any noise in the dataset, more checks were performed. By looking at all variables' values, it is apparent that the dataset contains some inconsistencies, as all of the characteristics have values of zero, except of the "DiabetesPedigreeFunction" and "Age". This cannot be true, save from the "DiabetesPedigreeFunction" and "Pregnancies" variables, which may be zero.

All observations with zero entries could be treated with three ways. The first method is to exclude those values form the dataset. The second approach is to replace each zero value with the mean value of the variable. The third approach of dealing with zero values includes their replacement with the average of each class identified (0 or 1), which is similar to the second method but more successful in the present situation. Additionally, as can be observed from diagram 1, in the specific dataset 65% of the patients are diabetic cases (around 500), that can lead to inaccurate replacement, by performing the second method.

Thus, in the present thesis, the third approach has been followed. In order to attribute, in a proper manner, means to zero values, is required to calculate means of each variable based on whether a patient has diabetes or not, excluding the zero values. The specific is considered important because it may result to not-representative mean values. Therefore, the means attributed to zero values of the variables are presented in the next table:

Table 9: Mean values of variables for each class

Outcome	**Glucose**	**BloodPressure**	**SkinThickness**	**Insulin**	**BMI**
0	110.64	70.88	27.24	130.29	30.86
1	142.32	75.32	33	206.85	35.41

Next step is to examine the existence of outliers. In the below diagram some outliers can be detected by creating box-plot diagrams for the dataset’s variables.

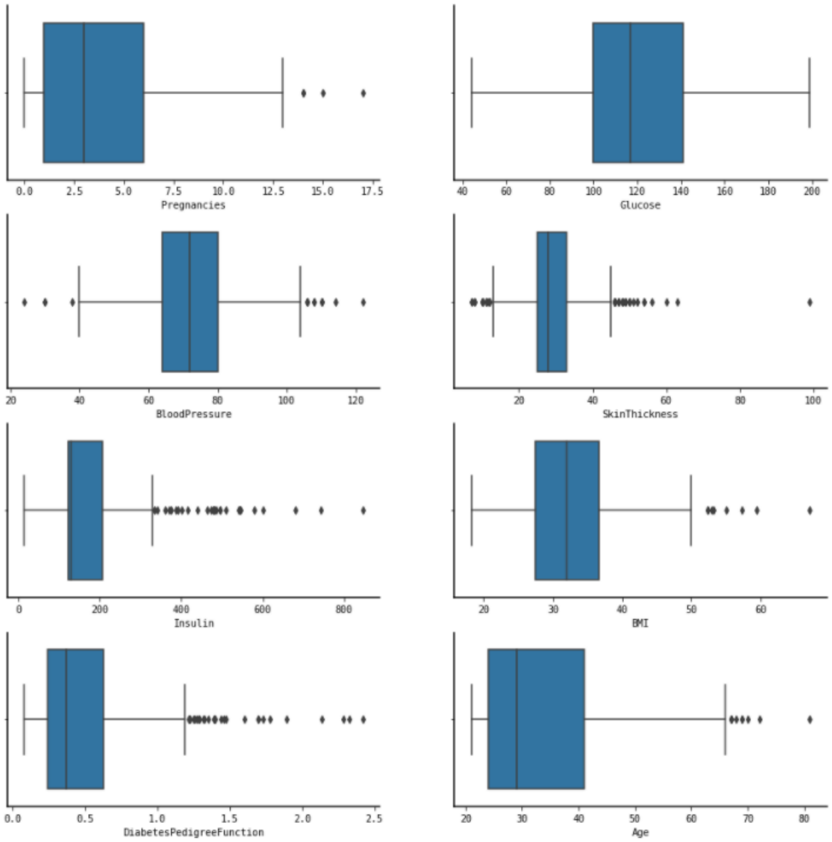

Diagram 2: Box-Plot diagrams of variables

In order to face the outliers' issue, that seems to be existed in almost all variables, it was chosen to be performed the interquartile range (IQR) measure.

The IQR in descriptive statistics is a measure of variability. It is defined as the difference between the data's 75th and 25th percentiles. By using linear interpolation, the data set is partitioned into quartiles, or four rank-ordered even sections, to calculate the IQR. These quartiles are denoted as follows:

- Q1 which is known as the lower quartile,
- Q2 which is known as the median and
- Q3 which is known as the upper quartile.

Thus,

$$IQR = Q3 - Q1 \quad (8)$$

because the lower quartile corresponds to the 25th percentile and the upper quartile corresponds to the 75th percentile.

The IQR is an example of a trimmed estimator, which improves the accuracy of dataset statistics by deleting lesser contribution, outlying values. It's also utilized as a reliable scale indicator. The box on a Box-plot can plainly depict this.

When looking for outliers in data, the IQR is frequently performed. On a box-plot, a fence is used to identify and categorize types of outliers:

- Lower Fence: Q1 - 1.5 * IQR
- Upper Fence: Q3 + 1.5 * IQR

Based on the abovementioned formulas, the threshold values for outlier observations were calculated, as Lower Limit and Upper Limit.

Table 10: Lower and upper limit for outlier observation

Variable	Lower limit	Upper limit
Pregnancies	0*	13.50
Glucose	37.88	202.88
BloodPressure	40.00	104.00
SkinThickness	13.00	45.00
Insulin	0*	334.87
BMI	13.85	50.25
DiabetesPedigreeFunction	0*	1.20
Age	0*	66.50

Given the fact that the values of the specific dataset's variables cannot be negative, lower limit is set to zero for those that a negative value was calculated (see * cases in table 9)

After the calculation of upper and lower limit, the replacement of the outliers with the respective value for each variable was followed. More specifically, seven variables (see diagram 2) appear to have outliers which have been replaced based on the figures above and the updated variables are presented below.

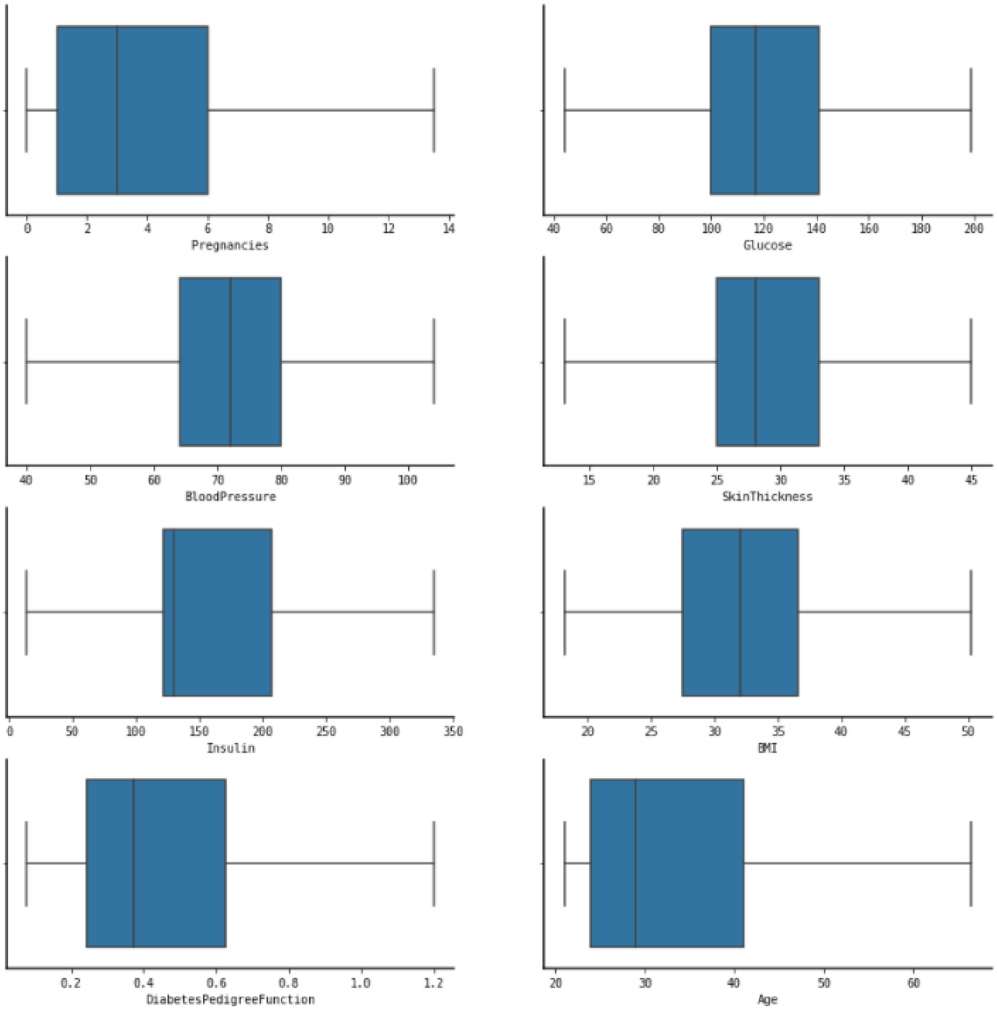

Diagram 3: Box-Plot Diagrams of variables, after outliers' detection

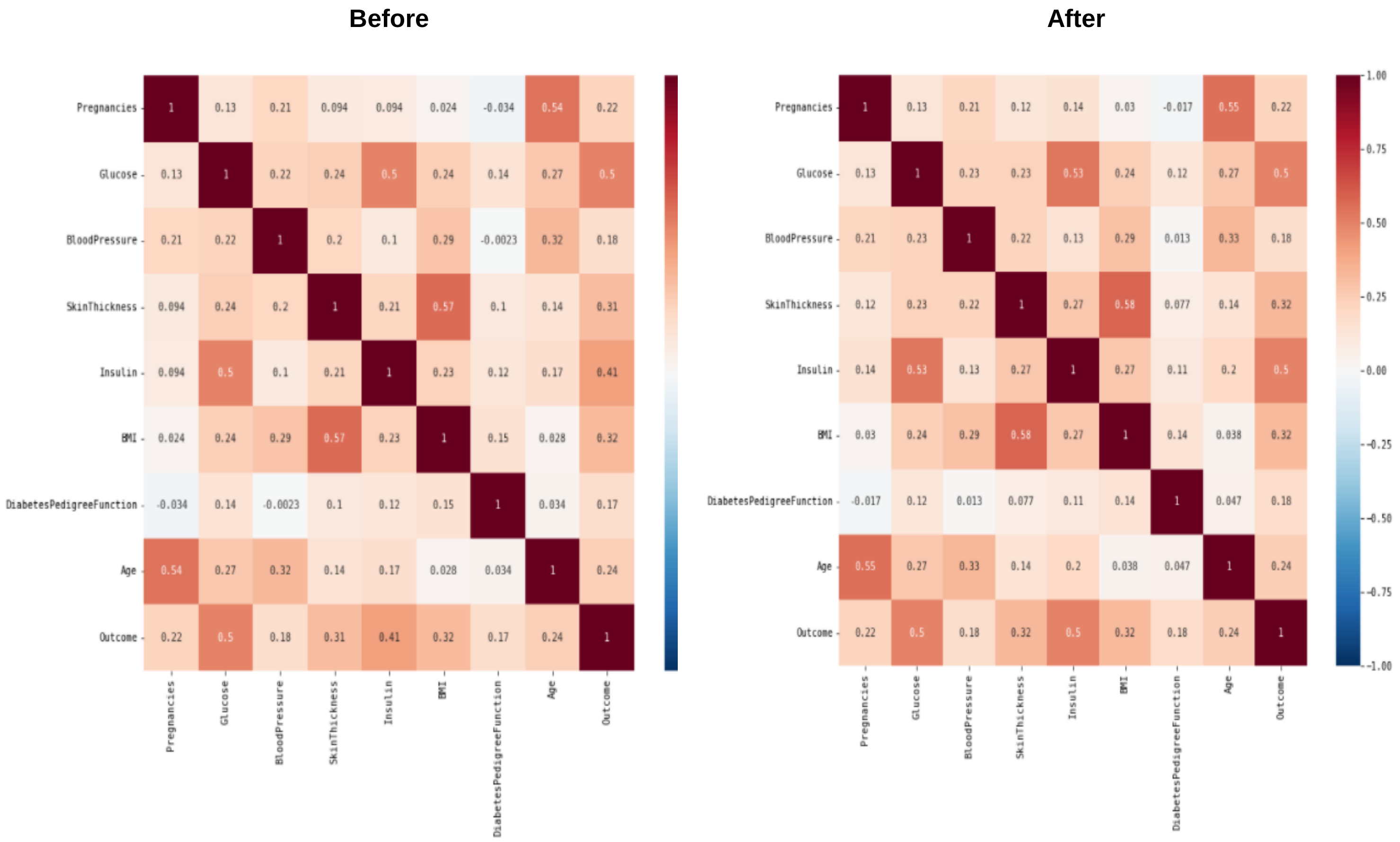

Diagram 4: Correlation matrix of variables before and after data processing

By comparing correlation matrices (diagram 4) before and after data processing, it can be observed that all variables have not significant correlation (>75%) in between them. The variables with the greatest correlation figure are “BMI” with “SkinThickness”, but there is no reason to exclude one of them. Thus, was decided to proceed in the modelling phase with all variables included in the dataset. Moreover, correlation with class variable is increased in all cases after handling zero values and outliers.

In the next diagram 5 the distribution of each variable is presented, after all modifications described above.

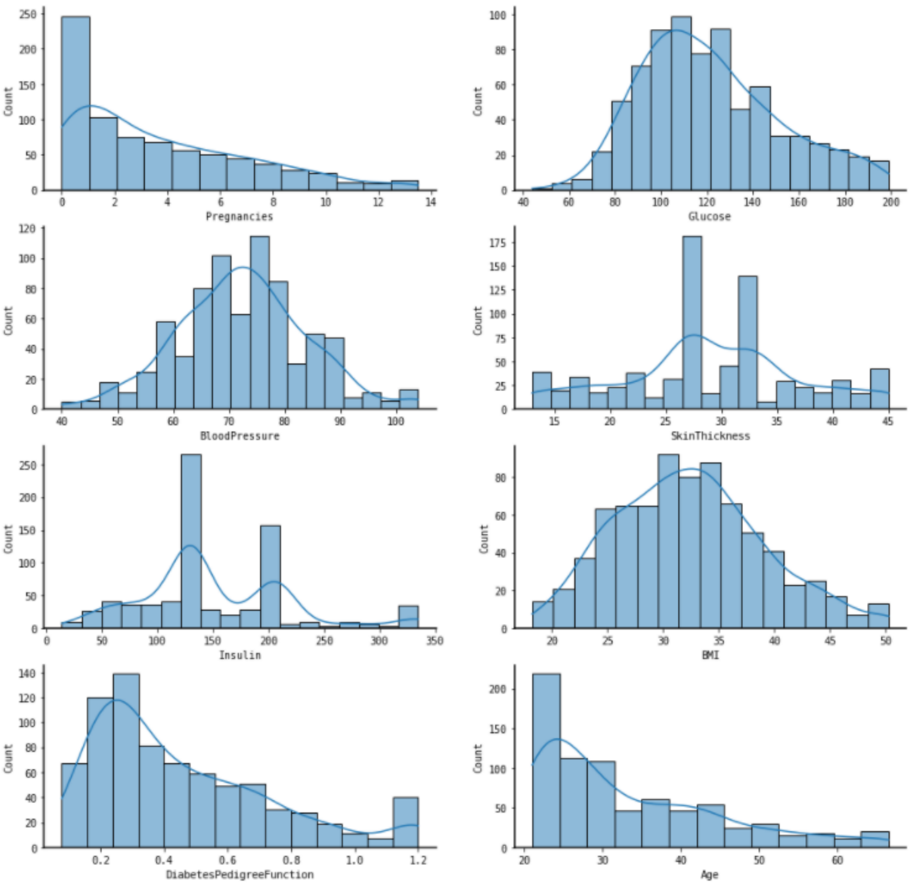

Diagram 5: Histograms of the variables

7.4. Training of classifiers

The structure of this phase includes three different levels of modelling and two methods for the calculation of the evaluation measures. More specifically, given that in the class variable the number of cases in each class is allocated unequally, as described above, the six algorithms were trained with respect to two techniques for performance evaluation:

i. Train-test split method
ii. CV method

under the next three categories:

i. without using any oversampling method
ii. by using SMOTE oversampling method

iii. by using ADASYN oversampling method

7.4.1. Train-test split method

With respect to model training and evaluation, the train-test split method was followed. The specific is a method for assessing a ML algorithm's performance. It can be used for any supervised learning technique and can be utilized for classification or regression tasks. For the present thesis, it was chosen to perform the split method with the percentages of 70%-30% for training and test datasets respectively.

The technique includes the separation of the dataset into two subgroups, train and test datasets. The training dataset is the first subset, which is used to fit the model. From the other hand, the test dataset is not used in the training process of the model; instead, the predicted values through train dataset are compared to the knowledge provided by test dataset for evaluation reasons. In few words:

Train Dataset: It is used to fit the ML model.

Test Dataset: It is used for the fitted ML model evaluation.

7.4.2. Cross validation method

The train-test split method described above may presuppose two key characteristics on the dataset. The first, is related to the sample size and the second to the balance of the class value. To be more specific, the former characteristic can be explained based on the fact that in case of splitting the existing dataset into lesser size, it may cause insufficient data for the training process. The latter, refers to the fact that in case of imbalanced class variable the randomly split train dataset, may not contain sufficient classified samples in every class and as a consequence prediction may not be reliable.

Based on the above two conditions, it was considered that the dataset has imbalanced class variable as well as is relatively small, given the medical nature and importance of the study. Therefore, a k-fold CV has also performed and the respective mean accuracy of each model is calculated. This aims to observe whether CV method would increase models’ performance.

For each algorithm that CV method was used, the dataset is divided into smaller sub datasets. In this method the dataset is separated in such a way as to provide abundant data not only for the training phase, but also for its validation. In this way, over-training of the model is avoided. In more details, the dataset is divided into k subsets, with k ranges from five (5) to ten (10) usually, depending on the data set.

Thus, it was chosen a k=10-fold CV additional calculations, for each category and model.

7.5. Parameter tuning

In the context of the algorithms chosen to be performed for the diagnosis of diabetes, the advantages that contributed in the selection of the algorithms are introduced below, as well as any tuning parameter applied, in order to increase accuracy levels. The tuning was performed in the train-test split method and for the first training category, that is without using any oversampling method.

7.5.1. Decision tree

Decision tree is a prediction method used in statistics, data mining and ML. Decision tree is simple to understand and interpret, since it can be visualized. Additionally, it requires very minimum data preparation, whereas other techniques frequently necessitate data normalization, dummy variable creation, and blank value elimination. The cost of using the decision tree (for data prediction) is proportional to the number of data points utilized to train it. Decision tree can handle both categorical and numerical data whereas other techniques are specialized for only one type of variable. [37]

With respect to tuning parameters, and in order to obtain the best results and avoid any overfitting by performing decision tree model, the parameter that specifies the maximum depth of each tree, which is called max_depth parameter has been modified. When max_depth is set to default (none), each tree will grow till every leaf is pure. A pure leaf is one in which all of the data on it is from the same class. By setting max_depth to default, to 1 and to 2 the accuracy alters from 0.84 to 0.82 and to 0.84, as presented in table 11.

Table 11: Accuracy of test dataset for max_depth tuning

max_depth	Accuracy of test dataset
default	0.84
1	0.82
2	0.84

So, it was decided to set max_depth in default parameter.

7.5.2. Random forest

The random forest method is a ML method for sorting, regression and other processes. It is a type of decision tree and specially is a collection of simple tree predictors, such that each tree produces a response when a set of predictor values is given as input. It recognizes outliers and anomalies in knowledgeable data. It is one of the most accurate learning algorithms available. For many datasets, it produces highly accurate classifiers. Additionally, given that random forest sometimes overfits with datasets with noisy classification/regression tasks [38], this issue was tackled by performing preprocessing of the dataset for the noise, presented in previous paragraph.

In order to improve the training and the results of the Random Forest algorithm, a modification on the model’s parameters has been performed. More specifically, the parameter “number of estimators” which specifies the number of trees in the forest of the model, has been modified. It was observed that by changing the number of estimators from 100 to 110 and 120, accuracy alters from 0.86 to 0.87 and to 0.87, as presented in table 12.

Table 12: Accuracy of test dataset for number of estimators tuning

number of estimators	Accuracy of test dataset
100	0.86
110	0.87
120	0.87

Thus, for the specific model the parameter number of estimators was set equal to 110.

7.5.3. Support vector machine

In ML, SVMs are support networks and supervised learning models with related learning algorithms for classification and regression processes. It can be applied to data that is not evenly distributed and whose distribution is unknown. Additionally, data cannot always be divided by a straight line, and must be evaluated over a greater region using cores. There are several classification algorithms used in ML, however SVM is better than most of them since it produces more accurate results.

Following the previous described algorithms parameter's tuning, a parameter's modification is also applied in SVM. More specifically, the C parameter was tested as equal to 1, 2 and 3 with the according results. For C=1,2 and 3 accuracy achieved was 0.81, 0.82 and 0.82 respectively, as presented in table 13.

Table 13: Accuracy of test dataset for C parameter tuning

C parameter	Accuracy of test dataset
1	0.81
2	0.82
3	0.82

Thus, it was chosen C=2 for the diabetes prediction with SVM.

7.5.4. K-nearest-neighbor

Taking into consideration that it takes less time to train, the KNN technique is known as a lazy-learning algorithm. It uses eager-based learning techniques (such as decision trees, random forest, naive Bayes, and others) and requires less time to classify. The KNN makes predictions based on a training dataset that is stored in memory. To categorize unknown data, for example, KNN does a distance calculation to discover the set of k objects from the training data that are nearest to the input data instance and assigns the most voted classes from these surrounding classes. [38]

For the specific algorithm, the number of neighbor's parameter has been tested. For k=10, 11, 12 and 13 the following accuracy numbers have been achieved: 0.82, 0.82, 0.81 and 0.83 respectively, as presented in table 14. Based on these figures, it was chosen to use k=13.

Table 14: Accuracy of test dataset for k parameter tuning

k parameter	Accuracy of test dataset
10	0.82
11	0.82
12	0.81
13	0.83

7.5.5. Logistic regression

For the logistic regression model, the max_iter parameter was modified. The specific parameter has a default value of 100 and refers to the maximum iterations deeded for the convergence of the solvers. Usually, this parameter provides greater accuracy numbers and in order to check if greater accuracy levels can be produced, max_iter=100 (default), 150 and 200 were used. For default value accuracy achieved was 0.79 followed by 0.77 and 0.78, as presented in table 15. It was decided to use the default max_iter parameter.

Table 15: Accuracy of test dataset for max_depth parameter tuning

max_depth	Accuracy of test dataset
default	0.79
150	0.77
200	0.78

7.5.6. Naive Bayes

The naive Bayes method predicts the class of new data instances based on the probabilities of each attribute belonging to each class in the training set. With the assumption that attributes belong to a class that is independent of each other, naive Bayes predicts datasets. The Gaussian Naive Bayes algorithm is used in this study, and it performs well with both continuous and discrete datasets. [38]

The advantage of naive Bayes is that it just requires a little amount of training data to estimate the classification parameters.

8. DIABETES DIAGNOSIS RESUTLS

8.1. Introduction

In the current diabetes diagnosis thesis, six ML models were developed, as described in previous paragraph. These models were chosen as to produce enough evaluation measures to compare and observe how the PIMA Indians dataset performs in each of them.

8.2. Training method summary

Below is presented a summary of the parameters that have been chosen after the parameter tuning described in previous chapter for each algorithm, as well as the two methods for the calculation of the evaluation measures.

Table 16: Parameters used in each classification model

Models	Parameters	Train-Test Split Method	CV Method
Decision Tree	Default	70% - 30%	k=10-fold
RandomForest	n_estimators=110	70% - 30%	k=10-fold
SVM	kernel = 'linear', C=2	70% - 30%	k=10-fold
KNN	n_neighbors=13, metric='euclidean'	70% - 30%	k=10-fold
Logistic Regression	Default	70% - 30%	k=10-fold
Naive Bayes	Default	70% - 30%	k=10-fold

8.3. Results presentation

In the next three tables, the results of the modelling phase for the three methodologies and the two training methods are presented. The TPR, FNR, TNR, FPR and Accuracy have been calculated for the train-test split method, as well as the accuracy for the 10-fold CV method for all six ML classification algorithms.

Table 17: Evaluation measures without using oversampling method

Model	Test dataset of the train-test split method					10-fold CV
	TPR	FNR	TNR	FPR	Accuracy	Mean Accuracy
Decision Tree	0.85	0.15	0.88	0.12	0.85	0.85
Random Forest	0.81	0.19	0.89	0.11	0.87	0.88
SVM	0.80	0.20	0.83	0.17	0.82	0.83
KNN	0.74	0.26	0.87	0.13	0.83	0.85
Logistic Regression	0.66	0.34	0.85	0.15	0.79	0.81
Naive Bayes	0.76	0.24	0.79	0.21	0.78	0.81

Table 18: Evaluation measures with SMOTE oversampling method

Model	Test dataset of the train-test split method					10-fold CV
	TPR	FNR	TNR	FPR	Accuracy	Mean Accuracy
Decision Tree	0.81	0.19	0.82	0.18	0.81	0.87
Random Forest	0.81	0.19	0.84	0.16	0.83	0.91
SVM	0.86	0.14	0.80	0.20	0.82	0.83
KNN	0.79	0.21	0.82	0.18	0.81	0.88
Logistic Regression	0.84	0.16	0.76	0.24	0.78	0.81
Naive Bayes	0.83	0.17	0.74	0.26	0.77	0.81

Table 19: Evaluation measures with ADASYN oversampling method

Model	Test dataset of the train-test split method					10-fold CV
	TPR	FNR	TNR	FPR	Accuracy	Mean Accuracy
Decision Tree	0.79	0.21	0.79	0.21	0.79	0.84
Random Forest	0.84	0.16	0.82	0.18	0.84	0.91
SVM	0.88	0.12	0.73	0.27	0.78	0.79
KNN	0.89	0.11	0.78	0.22	0.82	0.83
Logistic Regression	0.85	0.15	0.73	0.27	0.77	0.78
Naive Bayes	0.85	0.15	0.72	0.28	0.77	0.77

As can be observed from the results presented in the previous paragraph, in the first two methods, random forest, decision tree and SVM seem to obtain better results, with respect to TPR and FNR as well as in accuracy measure. In the third method, SVM and KNN algorithms reach their major results but accuracy is not following this trend. On the other hand, random forest and decision tree give better accuracy results in this method too. For all models and methods, the 10-fold CV seems to increase models' mean accuracy with the best result reaching around 91%, in random forest model.

In general, the third, ADASYN oversampling method, produces much better results in TPR and less accuracy and mean accuracy measures. Additionally, all three methods seem to achieve in all measurements and models' results greater than 70% (for TPR, TNR and accuracy/mean accuracy) supposing that all the preprocessing method and handling of outliers and zero values resulted to satisfactory measurements' results.

By comparing the methods among them, one can see that those that face imbalanced classification issue, produce quite greater TPR measurement, which is the purpose of the specific medical study. Another important figure is that KNN, which can be classified as a lazy learner algorithm achieved adequate greater results in the oversampling methods models. The most impressing increase is related to logistic regression, fact that can prove that the specific algorithm is susceptible to overfitting, if the dataset used appears unbalanced class variable. More detailed, the TPR increased from 66% to around 84% in the oversampling methods.

To conclude, from all the above presented methods, models and categories, it seems that ADASYN oversampling method for random forest produces the best combination of all measurements, by giving greater importance in TPR, followed by the other measures.

9. CONCLUSIONS AND FUTURE PROPOSALS

The specific thesis dealt with the diabetes diagnosis using ML algorithms. It falls under the binary classification problem.

In general, ML has already played a revolutionary role in many aspects of research fields of prediction. Medicine field is one of the most positively affected, as ML can contribute to the diagnosis of a variety of diseases, including diabetes.

By performing all the pre-processing method for missing values handling and outlier's detection, the final dataset has been used for training 6 ML algorithms, in three categories. Given the nature of the dataset, with imbalanced class value, the algorithms have been performed without any oversampling method, with SMOTE oversampling and ADASYN oversampling methods. Additionally, in order to train the model a train-test split method has been performed as well as o k=10 CV.

The results showed that the random forest model in ADASYN category produced overall greater measures, with respect to TPR at first and subsequently in accuracy.

By observing the key role of pre-processing methods of the dataset as well as the tuning parameters of the algorithms, some extra processing can be applied in future researches. More specifically, in order to tackle with the outliers observed in the dataset, a quantile method in addition to the IQR method performed in the specific study, could also be performed. Furthermore, missing values can be handled using a prediction model, such as a regression. Finally, with respect to tuning parameters, all models could be tested with modified parameters as to achieve better evaluation measures.

ABBREVIATIONS

ML	Machine Learning
SMOTE	Synthetic Minority Over-Sampling Technique
ADASYN	Adaptive Synthetic Sampling for imbalanced learning
SVM	Support Vector Machine
KNN	K-Nearest Neighbor
NN	Nearest Neighbor
CDC	Centers for Disease Control and Prevention
RL	Reinforcement learning
RBFN	Radial basis function network
SVM-linear	Linear kernel support vector machine
SVM-RBF	Radial basis kernel support vector machine
AMMLP	Artificial metaplasticity on multilayer perceptron
AMP	Artificial Metaplasticity
pdf	Probability density function
MLP	Multilayer Perceptron
CV	Cross-validation
AUC	Area under curve
PCA	Principal component analysis
PC	Principal components
MDR	Multifactor dimensionality reduction
MDS	Multidimensional scaling
PCR	Principal component regression
LDA	Linear discriminant analysis
CVC	CV consistency

CE	Classification error
$\overline{\mathrm{CE}}$	Average classification error
PE	Prediction error
$\overline{\mathrm{PE}}$	Average prediction error
BA	Balanced accuracy
CART	Classification and Regression Tree
ID3	Iterative Dichotomiser 3
CHAID	Automatic Interaction Detection
TPR	True positive rate
FNR	False negative rate
TNR	True negative rate
FPR	False positive rate
PPV	Positive predictive value
IQR	Interquartile range
TN	True negative
TP	True positive
FP	False positive
FN	False negative

REFERENCES

[1] A. Blum: “Machine Learning Theory”, Department of Computer Science, Carnegie Mellon University

[2] Data Flair: “Machine Learning in Healthcare – Unlocking the Full Potential!”, Available in https://data-flair.training/blogs/machine-learning-in-healthcare/

[3] Center of Disease Control and Prevention: Diabetes Basics.

Available in: https://www.cdc.gov/diabetes

[4] National Diabetes Statistics Report 2020. Estimates of diabetes and its burden in the United States.

Available in: https://www.cdc.gov/diabetes/pdfs/data/statistics/national-diabetes-statistics-report.pdf

[5] Diabetes prevalence | Health at a Glance: Europe 2020 : State of Health in the EU Cycle | OECD iLibrary. Available in: https://www.oecd-ilibrary.org/sites/83231356-en/index.html?itemId=/content/component/83231356-en

[6] J. Alzubi, A. Nayyar, A. Kumar, “Machine Learning from Theory to Algorithms: An Overview”, Journal of Physics: Conference Series, Volume 1142, Second National Conference on Computational Intelligence (NCCI 2018) 5 December 2018, Bangalore, India

[7] J. Brownlee: “4 Types of Classification Tasks in Machine Learning”, available in https://machinelearningmastery.com/types-of-classification-in-machine-learning/

[8] R. Mohammed, J. Rawashdeh, M Abdullah: “Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results”, 2020

[9] O. Chapelle, B. Scholkopf, A. Zien, Eds: “Semi-Supervised Learning”, IEEE Transactions on Neural Networks, 2009

[10] Sandhya N. dhage, Charanjeet Kaur Raina: “A review on Machine Learning Techniques”, March 16 Volume 4 Issue 3 , International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC), ISSN: 2321-8169, PP: 395 – 399

[11] AyonDey , “Machine Learning Algorithms: A Review”, (IJCSIT) International Journal of Computer Science

[12] Paul Pavlidis, Ilan Wapinski2 and William Stafford Noble: “Support vector machine classification on the web”, 2003

[13] Vapnik,V.N. (1998) Statistical Learning Theory. Adaptive and Learning Systems for Signal Processing, Communications, and Control. Wiley, New York

[14] R. Gandhi: “Support Vector Machine — Introduction to Machine Learning Algorithms”, available in https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

[15] M. Bolandraftar: “Application of K-nearest neighbor (KNN) approach for predicting economic events theoretical background”, 2013

[16] A.D.Dongare, R.R.Kharde, Amit D.Kachare: Introduction to Artificial Neural Network, International Journal of Engineering and Innovative Technology (IJEIT), 2012, ISSN: 2277-3754

[17] Ritchie MD, Hahn LW, Roodi N et al. Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer. Am J Hum Genet 2001;69:138–47

[18] D. Gola, J.M. Mahachie John, Kristel van Steen, I.R. König: “A roadmap to multifactor dimensionality reduction methods”, Briefings in Bioinformatics, Volume 17, Issue 2, March 2016, Pages 293–308

[19] Hahn LW, Ritchie MD, Moore JH. Multifactor dimensionality reduction software for detecting gene-gene and gene-environment interactions. Bioinformatics 2003;19:376–82

[20] Velez DR, White BC, Motsinger AA, et al. A balanced accuracy function for epistasis modeling in imbalanced datasets using multifactor dimensionality reduction. Genet Epidemiol 2007; 31:306–15

[21] Shokri R, Stronati M, Song C, Shmatikov V. Membership inference attacks against machine learning models. InSecurity and Privacy (SP), 2017 IEEE Symposium on 2017 May 22 (pp. 3-18). IEEE.

[22] Korting, Thales Sehn. "C4. 5 algorithm and Multivariate Decision Trees." Image Processing Division, National Institute for Space Research—INPE

[23] Diego Andina, Santiago Torres-Alegre, Martín J Alarcón, Juan I Seijas andMarta de-Pablos-Álvaro: “Robustness of Artificial Metaplasticity Learning Algorithm”. Group for Automation in Signals and Communications. Technical University of Madrid. Madrid, Spain

[24] D. Andina, A. Alvarez-Vellisco, A. Jevtic and J. Fombellida, "Artificial metaplasticity can improve artificial neural network learning", Intelligent Automation and Soft Computing, SI on Signal Processing and Soft Computing .15(4) (2009) 683-696.DOI: 10.1080/10798587.2009.106430578. F J Ropero-Pelaez, D Andina, "Do biological synapses perform probabilistic com-putations?", Neurocomputing

[25] A. M. Cedeño, J. Quintanilla-Dominguez, D. Andina: “Breast cancer classification applying artificial metaplasticity algorithm”, Neurocomputing. Vol 74(8), pp. 1243-1250. 2011. doi: 10.1016/j.neucom.2010.07.019

[26] A. M. Cedeño, D Andina: “Data mining for the diagnosis of type 2 diabetes”, June 2012, Conference: World Automation Congress (WAC 2012)

[27] Loether H.1. and McTavish D.G., Descriptive and inferential statistics: An introduction (4th ed.). Needham Heights, MA: Allyn and Bacon. (1993)

[28] Yan S., Yiyuan T., Shuxue D., Shipin L., Yifen c., Diagnose the mild cognitive impairment by constructing Bayesian network with missing data. Expert Syst. Appl., Vol 38(1), pp. 442-449, 2011. doi:1O.1016/j.eswa.201O.06.084

[29] W. Wong, P. J. Fos and F. E. Petry, “Combining the Performance Strengths of the Logistic Regression and Neural Network Models: A Medical Outcomes Approach”, The Scientific World JOURNAL (2003)

[30] H. Huang, H. Xu, X. Wang and W. Silamu: “Maximum F1-Score Discriminative Training Criterion for Automatic Mispronunciation Detection”, IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, VOL. 23, NO. 4, APRIL 2015

[31] A. J. Bowers, X. Zhou: “Receiver Operating Characteristic (ROC) Area Under the Curve (AUC): A Diagnostic Measure for Evaluating the Accuracy of Predictors of Education Outcomes”, Journal of Education for Students Placed At Risk, 2019

[32] R. Aishwarya, P. Gayathri, N. Jaisankar: “A Method for Classification Using Machine Learning Technique for Diabetes”, International Journal of Engineering and Technology (IJET)

[33] H. Kaur, V. Kumari: “Predictive Modelling and Analytics for Diabetes using a Machine Learning Approach”, December 2018, DOI:10.1016/j.aci.2018.12.004

[34] G. Kaur, A. Chhabra: “Improved J48 Classification Algorithm for the Prediction of Diabetes”, July 2014, International Journal of Computer Applications 98(22):13-17, DOI:10.5120/17314-7433

[35] A. M. Cedeño, D Andina: “Data mining for the diagnosis of type 2 diabetes”, June 2012, Conference: World Automation Congress (WAC 2012)

[36] M.F. Sanner: “PYTHON: A PROGRAMMING LANGUAGE FOR SOFTWARE INTEGRATION AND DEVELOPMENT”, The Scripps Research Institute

[37] G. Bhumika, R. Aditya, J. Akshay, A. Arpit, D. Naresh: “Analysis of Various Decision Tree Algorithms for Classification in Data Mining”, International Journal of Computer Applications (0975 – 8887) Volume 163 – No 8, April 2017

[38] A. Singh, M. N. Halgamuge, R. Lakshmiganthan “Impact of Different Data Types on Classifier Performance of Random Forest, Naïve Bayes, and K-Nearest Neighbors Algorithms”, International Journal of Advanced Computer Science and Applications, Vol. 8, No. 12, 2017