Republic of Iraq
Ministry of Higher Education
and Scientific Research
University of Babylon
College of Information Technology
Department of Software

A Modeling Approach to Handle the Prediction Problems in Small Datasets (HealthCare)

A Thesis
Submitted to the Council of the College of Information Technology for Postgraduate Studies of University of Babylon in Partial Fulfillment of the Requirements for the Degree of Master in Information Technology- Software

BY

Nuha Ahmed Salman Farag

SUPERVISED BY

Prof. Dr. Saad Talib Hasson Aljebori

2023 A.D. **1445 A.H.**

بِسۡمِ ٱللَّهِ ٱلرَّحۡمَٰنِ ٱلرَّحِيمِ

{يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ
دَرَجَاتٍ}

[المجادلة: 11]

صَدَقَ ٱللَّهُ ٱلْعَظِيمُ

Supervisor Certification

I certify that the thesis entitled “(A Modeling Approach to Handle the Prediction Problems in Small Datasets (Health Care)” was prepared under my supervision at the department of Software / College of Information Technology/ University of Babylon by (Nuha Ahmed Salman) as partial fulfilment of the requirements of the degree of master in information technology- Software.

Signature:

Supervisor Name: **Prof. Dr. Saad Talib Hasson**

Date: / /2023

The Head of the Department Certification

In view of the available recommendations, I forward the thesis entitled “A Modeling Approach to Handle the Prediction Problems in Small Datasets Health Care) for debate by the examination committee.

Signature:

Prof. Dr. Ahmad Saleem

Head of Software Department

Date: / /2023

Dedication

To My Parents

To My husband

To My husband's sister

And to my sons

With All My Respect and Love…

Nuha Ahmed Salman

Acknowledgement

First and foremost, I would like to thank Allah, God, without divine care I could not even have contemplated all the work involved in this study.

My thanks to my supervisor Dr. Saad Talib Hasson, for his help and support during the period of my study, he is the pillar of my work. Words are inadequate to express my heartfelt thanks for him tremendous support and help. I feel motivated and encouraged whenever I speak and share my thoughts with him. Without your encouragement and support, this thesis would not have materialized.

I would like to thank all the lecturers in my Information Technology college who have imparted immense knowledge to me.

Abstract

A small dataset is a dataset containing a little number of samples. Small datasets can pose a variety of challenges when it comes to building predictive models. Successfully building predictive models on small datasets requires a combination of domain knowledge, statistical methods, and machine learning techniques.

This thesis will focus on creating a prediction model for small data sets using conventional classification methods like logistic regression, decision trees, naive Bayes, and KNN. These methods have a very high degree of categorization prediction accuracy. Researchers may use different models with certain effort to increase classification prediction accuracy as machine learning methods grow in popularity and accessibility. To allow for adequate training and testing of models created using machine learning techniques, data are frequently constrained by smaller sets of observations than what is typically requested. One approach to improve the small dataset is to extend it by using a synthetic minority oversampling technique (SMOTE).

A statistical measure is performed for the original and extended datasets to indicate their similarity. Another approach is to check whether the dataset is balanced or not. Balancing the imbalanced dataset is essential before implementing any machine learning prediction tool. In this thesis three datasets are used. Four predictive machine learning algorithms are utilized (logistic regression, decision trees, naive Bayes, and KNN). Confusion matrices for these data are created before and after extensions. Accuracy, Precision, Recall and F-Score are calculated for each original small dataset and the extended datasets. As an example; the results for Higher disease small dataset shows the accuracy for Logistic regression is 85%, Naïve Baise is 83%, Decision tree is 69% and KNN is 64%. After extension, the Accuracy results are for Logistic regression is 91%, Naïve Baise is 91%, Decision tree is 88% and KNN is 94%.

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

DM	**Data Mining**
ML	**Machine Learning**
AI	**Artificial Intelligence**
FN	**False Negative**
FP	**False Positive**
TN	**True Negative**
TP	**True Positive**
FS	**Features Selection**
FR	**Feature Ranking**
NB	**Naive Bayes**
DT	**Decision Tree**
KNN	**K-Nearest Neighbor**
LR	**Logistic Regression**
SMOT	**Synthetic Minority Over-Sampling Technique**

CHAPTER ONE

Introduction

Chapter One
Introduction

1.1 Introduction

A dataset is a collection of related data, organized in a structured or semi-structured format, used for analysis, research, or to train machine learning models. It may be compared to a table or matrix where each row represents a single instance or observation and each column represents a particular property or feature of that occurrence (Harper, et.al, 2015). Datasets can be created from a variety of sources, such as surveys, observations, databases, scientific experiments, and even artificially created data. They may include several sorts of data, including text, multimedia, numerical, categorical, and categorical data (Mesaros, et.al, 2018).

The use of datasets is crucial in a variety of industries, including data science, machine learning, statistics, social sciences, and healthcare. They provide the framework for carrying out studies, formulating hypotheses, identifying patterns, and drawing conclusions. Additionally, datasets are frequently shared and made accessible to the public in order to promote cooperation, repeatability, and improvements in research. Understanding a dataset's characteristics, such as its size, data kinds, missing values, and inherent biases, is essential before using it. To further assure the data's quality and usefulness for analysis, preparation procedures including cleaning, converting, and normalizing the data could be required (Ramsey, C. B., 2017).

A small dataset refers to a collection of a limited number of observations, which may pose challenges in statistical analysis and generalization. However, these datasets still have utility and can contribute to various research fields, as long as the limitations and characteristics of small samples are taken into account (Mesaros, et.al, 2018).

The practice of analyzing previous data to predict or forecast future results is known as predictive modeling for datasets. It entails figuring out how the data relate to one another and trends in order to build a model that can be applied to forecast unknown or upcoming data points. Datasets are often split into two subsets, the training set and the testing set, in order to do predictive modeling. The prediction model is developed and trained utilizing a variety of methods and techniques, including artificial neural networks, decision trees, and linear regression. The model gains the capacity to predict outcomes by learning from the patterns and information provided in the training data (Gebru, et.al, 2021).

The testing set is used to assess the model once it has been trained. This makes it possible to evaluate the model's effectiveness and accuracy by contrasting its predictions with the actual values in the testing data. This assessment aids in determining the model's efficacy and dependability in producing precise predictions for unobserved data. Numerous businesses and disciplines, including banking, healthcare, marketing, and weather forecasting, can benefit from the use of predictive modeling. Predictive models can offer insightful information, support decision-making, help identify potential dangers, and enhance various business processes by using past data (Reddy, et.al, 2020).

Prediction problems in applications of small data sets refer to challenges that arise when using machine learning or statistical models to make predictions based on small amounts of data. These problems can include (Harper, et.al, 2015) (Gebru, et.al, 2021) (Mesaros, 2018):

- Overfitting: When a model learns the data's noise rather than its underlying patterns, it performs poorly when applied to new data.
- Insufficient data: small data sets may not provide enough information to capture the complexity of the underlying relationships, resulting in weak or biased models.
- High variance: small data sets can lead to high variance in model performance, making it difficult to evaluate the true effectiveness of a model.
- Lack of diversity: small data sets may not represent the full range of possible scenarios or input values, leading to biased predictions.

Practitioners may utilize approaches like cross-validation, regularization, feature engineering, or ensemble methods to overcome these issues and enhance the performance of models created from tiny data sets (Reddy, et.al, 2020).

1.2 Problem statement

Implementing predictive modeling approaches on small datasets is a challenging process. A small dataset is usually poor in training phases due to limited number of available data. Different approaches can be used to improve prediction accuracy. One approach is to utilize the Synthetic Minority Over-Sampling Technique (SMOTE).

Most of the special healthcare datasets are from the small dataset types due to privacy reasons. It is important to work on developing accurate, reliable, and generalizable predictive models that can effectively work with limited data points while minimizing the risk of overfitting and maintaining interpretability.

The wanted solution is to create a reliable prediction model that can effectively handle the challenges associated with small datasets while providing accurate and efficient predictions for practical applications.

1.3 Thesis aim

The main aim of this thesis is to utilize an acceptable predictive modeling approach to handle small datasets.

This aim can be achieved by the following objectives:

1. Analyze the small dataset by identifying and creating the relevant features from the available data to improve the model's predictive power and better capture the underlying relationships in the small dataset.
2. Implement the SMOTE algorithm to extend and balance the small datasets.
3. Implement a correlation algorithm to Rank the features due to their importance and effectiveness with respect to the target.
4. Implement another Correlation process to indicate the correlation between any feature pair. The highly correlated coefficient value can be eliminated.
5. Implement four prediction models on the original small datasets and on the extended datasets.

1.4 Related works

In recent years, the discipline of machine learning has made major advances in allowing the interpretation of massive volumes of datasets. This involves studying small data problems, which are becoming increasingly crucial for understanding to predict that data. Table 1.1 shows a previous related work summary containing the used problem, used methods, goal, Dataset and their results:

Table 1.1 Related Works

Ref.	Problem	The used algorithm	Goal	The used Dataset	Results
(Alotaib, 2019)	How to predict accurate about heart disease diagnosis	Decision Tree, Logistic Regression, Logistic Regression SVM, Naïve Bayes and Random Forest	Heart Disease Prediction	UCI heart disease dataset.	The accuracy of Decision Tree is 82.22%, Logistic Regression 82.56%, Random Forest 84.17%, and Naïve Bayes 84.24% and Logistic Regression SVM is 84.85
(Imran, et.al, 2019)	understanding student and their learning environment	J48, and MLP	Automatic Student performance prediction	UCI Student Performance	J48 achieved highest accuracy 95.78%
(Fornand & Kearney, 2020)	Unexpected financial industry advances and advancements	Classification and Regression Trees (CART)	anticipate the banks' credit ratings in the GCC	dataset of the macro and bank 2010 to 2018	Accuracy for Classification and Regression Trees (CART) is 0.86.
(Tate, et.al, 2020)	Predicting mental health problems in Small dataset of adolescence	random forest, support vector machines, neural network,	develop a model that can predict mental health problems in mid-adolescence.	adolescence with 474 variables	random forest model (AUC = 0.739, 95% CI 0.708–0.769), followed closely by support vector

		and XGBoost			machines (AUC = 0.735, 95% CI = 0.707–0.764).
(Izonin, et.al, 2021)	Processing and effective data mining of small datasets	General Regression Neural Network (GRNN)	Prediction from Small Datasets in Medical Application	small set of clinical data from Urine Analysis Data laboratory test	RMSE is 3.68 MAE is 2.77, Application time 0. 298 of GRNN-based input doubling method
(Han, et.al, 2021)	Problem in the RF training step	Random forests (RF)	Enhancing random forest forecasts from two-phase sampling designs in small datasets	Immunologic marker dataset from the Human Immunodeficiency Virus (HIV) Vaccine Trials Network (HVTN) Community.	When variable screening is not used, class balancing enhances the performance of random forest prediction.
(Riekert, et.al, 2021)	achieve high accuracy and how the training sets should be sized to efficiently use annotation labor.	SVM and NBSVM	creating accurate text classifiers using machine learning is small training sets	ACL IMDB dataset	high classification accuracy can be achieved using a manually annotated dataset of only 300 examples.
(Al-Najjar, et.al, 2021)	Early warning systems for predicting financial crises using machine learning methods	Logistic regression, SVM, Neural network	predicting need for hospitalization in infections against the small dataset	macroeconomic dataset covering 17 countries between 1870 and 2015	Accuracy of 2004 to 2017 Forecasting period of logistic is 0.867, SVM is 0.867, Neural is 0.872
(Boukhatem, et.al, 2022)	predicting heart diseases	Multilayer Perceptron (MLP), Support Vector Machine	designing smart systems to accurately diagnose them based on electronic health	using data of major health factors from patients	The SVM model performed best with 91.67% accuracy.

		(SVM), Random Forest (RF), and Naïve Bayes (NB)	data, with the aid of machine learning algorithms		
(Anil Kumar, et.al, 2022)	optimize the process of detection from the lung cancer dataset	Support vector machines (SVMs) and SMOTE methods	construct a sustainable prototype model for the treatment of lung cancer using the current developments in computational intelligence without negatively impacting the environment	Several cancer datasets from the University of California, Irvine, library	The proposed method gets a 98.8% of accuracy rate
(Ranjan, et.al, 2022)	If the prognosis of cancer is done well in advance, it is curable	Random Forest Classifier, Convolutional Neural Network, and ResNet50	categorize different forms of cancer based on criteria including age, gender, and race/ethnicity	Kaggle online system, the TCIA informed on kidney cancer, the HCC Liver Cancer dataset, the Brain Cancer dataset	Lung, Liver, Kidney, Breast, and Brain cancers had accuracy rates of 99%, 75.75%, 88.09%, 96%, and 81%, respectively.
Nuha Thesis	Prediction based Small health datasets	Decision tree, Naïve Biase, KNN, and Logistic regression	Select the suitable prediction algorithm	Heart Disease Dataset, Lung Cancer Dataset, Cancer Patients Dataset	94 % for the Naive Bayes, 100 % for the Decision tree, 100 % for the K-Nearest Neighbour, 99.9 % for the Logistic Regression

1.5 Organization of the Thesis.

This thesis is organized into five chapters as follows:

In addition to chapter one:

Chapter Two: present the theory section will detail the research design and methods used to collect and analyze data.

Chapter Three: Discusses the proposed models and how the technologies can be integrated into a prediction process.

Chapter Four: evaluates the proposed model performance by using python programing language; Studying the results and analyzing them fully.

Chapter Five: includes the thesis conclusion and future work directions.

CHAPTER TWO

Theoretical Background

Chapter Two
Theoretical Background

2.1 Introduction

In this chapter, the terminology and tools that were used in this research to implement the proposed model and show the results to be analyzed will be clarified. The chapter contains several sequential sections according to a method that illustrates the sequence of implementation and use of those tools that helped build the proposed model.

2.2 Small Dataset

A small dataset is a collection of data points that has a relatively low number of records, rows, or observations compared to larger datasets. The number of data points and the number of features or variables linked to each data point are frequently used to estimate the size of a dataset (Gebru, et.al, 2021) (Mesaros, et.al, 2018). Small datasets can be easier to manage, process, and analyze due to their limited size. However, they may also present some challenges (Harper, et.al, 2015) (Ramsey, C. B.,2017) (Reddy, et.al, 2020):

- **Overfitting**: A small dataset raises the danger of overfitting in machine learning, a condition where a model learns the noise in the data and performs well on training data but badly on new, untainted data.
- **Lack of representativeness**: small datasets may not capture the full range of variability present in the population, leading to biased or less accurate insights and predictions.
- **Statistical power**: With a small dataset, it can be difficult to detect significant patterns, trends, or relationships in the data, limiting the conclusions that can be drawn from the analysis.

Several strategies can be used to overcome these obstacles as the following (Harper, et.al, 2015) (Mesaros, et.al, 2018) (Zhang, et.al, 2018):

- Data augmentation: Increasing the dataset size by adding new data points by transforming the existing data, for as by rotating, translating, or flipping picture datasets.

- Transfer learning: Making use of previously trained models from larger datasets to boost the effectiveness of machine learning models on smaller datasets.
- Regularization: Represent a technique that can be used to calibrate machine learning models in order to minimize the adjusted loss function as lasso regression (L1) or ridge regression (L2) regularization to limit the model's complexity and lower the likelihood of overfitting.
- Cross-validation: By splitting the dataset into several training and testing sets and averaging the performance indicators, techniques like k-fold cross-validation may be utilized to make better use of sparse data.
- Ensemble methods: integrating the findings from several models to prevent overfitting and boost performance overall.

2.3 Data Preprocessing

In order to make raw data appropriate for analysis and modeling, raw data must be cleaned, transformed, and prepared as part of the data preparation stage of machine learning (Zhang, et.al, 2018). Data preprocessing is regarded as a crucial step in the machine-learning pipeline since the caliber of the data utilized to train a machine-learning model can have a substantial impact on the accuracy and dependability of the outputs (Zebari, et.al, 2020). Figure 2.1 shows the knowledge discovery steps.

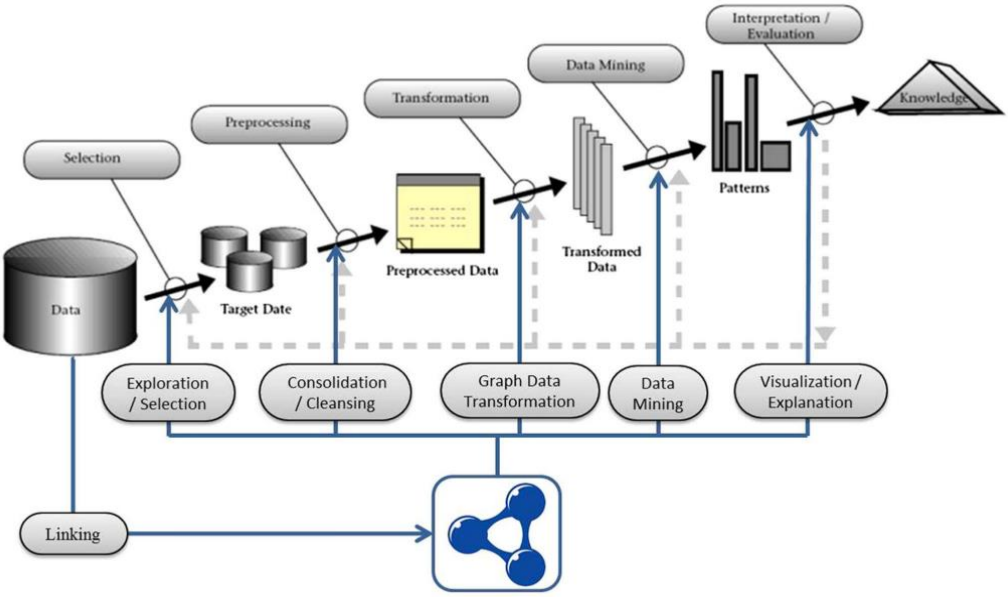


Figure 2.1 Knowledge Discovery Steps (Ristoski, P., & Paulheim, H., 2016)

2.4 Feature Selection

Choosing the most relevant features (variables, predictors, and characteristics) to include in a machine-learning model by locating and selecting them from the original dataset is referred to as feature selection (Chen, et.al, 2020). Removing pointless or unnecessary characteristics, reducing the number of dimensions in the data, and enhancing model performance are the main objectives (Ramsey, C. B.,2017). There are a number of feature selection techniques, which are often divided into three groups: filter techniques, wrapper techniques, and embedding techniques (Phyu, et.al, 2018).

1. **Filter Techniques**: Filter techniques are predicated on the inherent characteristics of the data. Without using a specific machine learning technique, they assess each feature's importance by examining how it relates to the target variable. Typical filtering techniques include (Zebari, et.al, 2020) (Chen, et.al, 2020):
   - Pearson's association Coefficient.: Measures the linear association between two continuous variables
   - Mutual Information: This measure, applicable to both continuous and discrete data, quantifies the mutual dependency between two variables.
- Chi-square Test: Determines whether two categorical variables are correlated.

Filter approaches are less prone to overfitting and are computationally efficient. They don't take into account potential relationships between features, instead focusing on each feature alone.

2. **Wrapper Methods**: Wrapper approaches generate several models with various feature subsets and compare the results to determine the relative importance of each model. The most widely used wrapper techniques are (Phyu, et.al, 2018) (Zebari, et.al, 2020):
   - Recursive Feature Elimination (RFE): This technique begins with all features and systematically eliminates the least significant feature until the required number of features is reached.
   - Forward Selection: Starts with a blank feature set and incrementally adds the most pertinent feature up until the point when performance no longer improves.

- Backward Elimination: Starting with all features, the least important feature is gradually removed until the performance begins to suffer.

  Given that they take feature interactions into account, wrapper approaches can identify superior feature subsets. In contrast to filter approaches, they are more prone to overfitting and expensive to compute (Phyu, et.al, 2018).

3. **Embedded Methods**: By including feature selection within the machine learning algorithm's training process, the advantages of both filter and wrapper techniques are combined in embedded methods. Typical embedded strategies include (Kowshalya, et.al, 2019) (Zebari, et.al, 2020) (Chen, et.al, 2020):
   - LASSO (Least Absolute Shrinkage and Selection Operator): This linear regression technique selects features by applying L1 regularization to bring some coefficients to zero.
   - Ridge Regression: Reduces the coefficients using L2 regularization, but not to zero. For better feature selection, it can be used with LASSO (Elastic Net) and aids in the detection of multicollinearity.
   - Decision Trees and Random Forests: These algorithms automatically determine the best split based on a criterion (such as Gini impurity or information gain) at each node.

   Embedded methods are generally more efficient than wrapper methods and take into account feature interactions. Depending on the particular problem, dataset, and machine learning algorithm being utilized, the best feature selection technique must be chosen. It's essential to experiment with different methods and validate their effectiveness through cross-validation or other evaluation techniques (Phyu, et.al, 2018) (Zebari, et.al, 2020). Figure 2.2 show the stages in feature selection process (Venkatesh, B., & Anuradha, J., 2019).

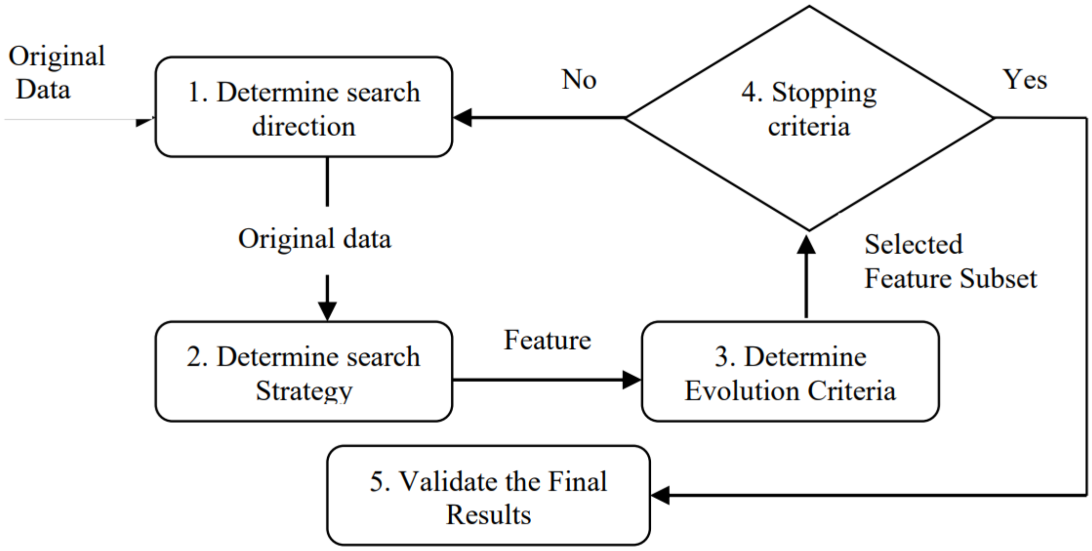

Original Data
1. Determine search direction
No
4. Stopping criteria
Yes
Original data
Selected Feature Subset
2. Determine search Strategy
Feature
3. Determine Evolution Criteria
5. Validate the Final Results

Figure 2.2 Stages in feature selection process (Venkatesh, B., & Anuradha, J., 2019)

2.4.1 Feature Ranking

Feature ranking is the process of analyzing and rating the relevance of input characteristics in a dataset to evaluate their relative contribution to the model's predicted performance. This helps identify the most influential variables and can aid in dimensionality reduction, model interpretability, and increased prediction accuracy (He, S., Guo, F., & Zou, Q., 2020).

Once the features have been ranked, the least significant features can be eliminated from the dataset, which can enhance model performance by lowering overfitting and boosting interpretability, reduced computational complexity, and increased interpretability. To acquire a thorough understanding of the data, feature ranking should be used in conjunction with other approaches like data visualization and domain expertise. However, it's crucial to remember that feature ranking is not always appropriate or necessary (Phyu, et.al, 2018).

2.4.2 Correlation

A statistical measure known as correlation expresses how closely two variables are related. Understanding their linear relationship, which shows whether an increase or decrease in one variable is accompanied by an equivalent rise or fall in the other, is helpful (Kowshalya, et.al, 2019).

Pearson's correlation coefficient (equation 2.1), denoted by the letter "r," is the most often used correlation coefficient. The range of the Pearson correlation coefficient is from -1 to 1 (Akoglu, H., 2018) (Schwartz, R., & Stanovsky, G., 2022).

$$\boldsymbol{r} = \frac{\boldsymbol{n}(\sum \boldsymbol{xy}) - (\sum \boldsymbol{x})(\sum \boldsymbol{y})}{\sqrt{[\boldsymbol{n}(\sum \boldsymbol{x^2}) - (\sum \boldsymbol{x^2})][\boldsymbol{n}(\sum \boldsymbol{y^2}) - (\sum \boldsymbol{y^2})]}} \ldots \ldots\ldots\ldots \quad \textbf{2-1}$$

- Perfect negative correlation (r = -1): This means that while one variable rises, the other one falls in exactly the same proportion.
- r = 0: No correlation, indicating that the two variables do not have a linear relationship.
- r = 1: Perfect positive correlation, which means that as one variable rises, the other rises perfectly proportionately as well.

Between -1.0 and -0.7 there is a strong negative correlation, between -0.7 and -0.3 there is a moderate negative correlation, between 0.3 and 0.7 there is a weak or no relationship, and between 0.7 and 1.0 there is a strong positive correlation.

It is important to remember that correlation does not imply causality. Even while two variables may have a strong correlation, this does not always mean that one causes the other to change. The link could just be a coincidence, or perhaps a third factor is involved (Kowshalya et al., 2019).

In addition to Pearson's correlation coefficient, there are other correlation metrics, including Spearman's rank correlation coefficient (equation 2.2) and Kendall's Tau rank correlation coefficient (equation 2.3).

$$\boldsymbol{p} = \mathbf{1} - \frac{\mathbf{6}\sum \boldsymbol{d^2}}{\boldsymbol{n}(\boldsymbol{n^2} - \mathbf{1})} \ldots\ldots\ldots\ldots \quad \textit{(2-2)}$$

Where P = Spearman, d = difference in ranks, and n = number of pairs of data.

$$\boldsymbol{T = \frac{C - D}{C + D}} \textbf{...........} \textit{\textbf{(2-3)}}$$

T = Kendall's Tau, C = of total concordant pairs, and D = of total discordant pairs

These are non-parametric measures, which are more resistant to outliers and appropriate for non-linear connections since they evaluate the relationship between two variables based on their ranks rather than their actual values (Akoglu, H., 2018).

2.5 Balance and Imbalance Dataset

The balance of a dataset refers to its distribution of classes or categories, which is essential in machine learning and predictive modeling. Radical imbalance problems in datasets can severely influence predictive results. There are some methods to determine the balance of a dataset (Akoglu, H., 2018) (Schwartz, R., & Stanovsky, G., 2022):

1. **Frequency Distribution**: Frequency distribution is the most straightforward approach to determining the balance of a dataset. This strategy shows how many observations or instances are assigned to each category or class. The more balanced the categories, the more uniform the frequency of observations associated with each class.
2. **Visual Examination**: Another strategy to assess dataset balance is to plot the distributions of each category on a graph. Common visualizations include bar charts, pie charts, and histograms. Examining the plot can highlight imbalances in dataset categories.
3. **Statistical Test**: One statistical test that can be used to determine the dataset balance is the chi-square test. The chi-square test compares the frequency distribution of the dataset's categories to a theoretical, balanced distribution and determines the likelihood that the distribution occurs by chance.

$$\boldsymbol{X^2 = \sum \frac{(O - E)^2}{E}} \textbf{...........} \textit{\textbf{2-4)}}$$

O = the frequencies observed

E = the frequencies expected

2.6 Confusion Matrixes

To evaluate how effectively a classification model is doing, a confusion matrix is employed. It compares the actual labels on the data with the labels the model predicts would be there. The four cells that make up the matrix are true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). (Luque, et.al, 2019) (Li, et.al, 2023).

- **True positives (TP)**: A positive class that the model accurately predicted.
- **False positives (FP)**: A positive class was predicted by the model wrongly.
- **True negatives (TN)**: A negative class was successfully predicted by the model.
- **False negatives (FN)** are negative classes that the model mistakenly predicted.

The confusion matrix may be used to compute evaluation measures such as accuracy, precision, recall, and F1 score. It is a useful technique for identifying the benefits and drawbacks of a classification model and aids in guiding model adjustments (Sayyad et al., 2021).

The rows of the confusion matrix correspond to the real (true) classes, and the columns to the expected classes. The diagonal of the matrix represents the number of accurate predictions, while the cells off the diagonal reflect the number of wrong estimates .The model's overall accuracy is calculated by dividing the sum of the diagonal cells by the entire number of predictions (Luque et al., 2019). While accuracy assesses the proportion of true positives among all predicted positives, recall assesses the proportion of real positives among all actual positives. The harmonic mean of recall and accuracy is known as the F1 score, which is a balanced assessment of the model's performance. Confusion matrices may be represented using heatmaps, which can make it easier to see patterns and trends in the model's behavior. It's vital to remember that confusion matrices may be utilized to solve problems with multi-class classification as well as binary classification problems by expanding the matrix to include all possible classes. (Li, et.al, 2023) (Sayyad, et.al, 2021).

2.7 Machine Learning

Machine learning, a subset of artificial intelligence (AI), is the study of creating algorithms and computational models that enable computers to learn from data and make predictions or judgements based on that data (Zhang, et al., 2018). By analyzing and identifying patterns in the data, machine learning techniques allow a computer to "learn" how to carry out a certain task without having to be explicitly programmed. (Al-Mhiqani,, et.al, 2020).

Machine learning is used extensively across a wide range of industries. Some examples include picture and audio recognition, natural language processing, recommendation systems, medical diagnosis, financial modeling, and self-driving automobiles. offers an example of how the use of machine learning in sentiment analysis, a branch of natural language processing, may be advantageous (Al-Mhiqani et al., 2020).

To determine the sentiment (positive, negative, or neutral) communicated in a text, such as a product review, social media post, or customer feedback, sentiment analysis is a possible technique (Ibrahim, et.al, 2016) (Zhang, et.al, 2018). Using machine learning for sentiment analysis, one can quickly identify patterns and trends, address issues, and make data-driven decisions to improve their certain services (Ibrahim, et.al, 2016).

2.7.1 Naive Bayes

Based on Bayes' theorem, a key idea in probability theory, Naive Bayes is a well-known and simple machine learning algorithm. When trying to categorize an input data point into one of several classes or categories, the algorithm is especially helpful (Dey, et.al, 2016).

The term "naive" refers to the presumption that, given the class label, the features of the input data are independent of one another. Although this reduction makes the technique simple to use and computationally efficient, it may not always hold true in practical applications. Despite this drawback, the Naive Bayes classifier frequently exhibits surprisingly good performance in actual use (Al-Mhiqani, et.al, 2020).

Naive Bayes algorithm works (Dey, et.al, 2016) (Surya, et.al, 2019):

1. The approach determines the prior probabilities of each class label given a dataset with cases that have been tagged. No of the feature values, these are the chances that each class will appear in the dataset.

2. The algorithm determines the conditional probabilities of the feature values given each class label for each feature. This entails calculating the likelihood of a specific feature value, presuming that we are aware of the actual class label.
3. The algorithm employs the Bayes theorem to determine the posterior probabilities of each class label given the feature values of a fresh, unlabeled data point.
4. The new data point is given the class label with the highest posterior probability.

The Naive Bayes algorithm's simplicity and readability are among its advantages. Text categorization, spam identification, and medical diagnosis are just a few of the many classification issues that it can be used to solve. When the premise of feature independence is broken or when dealing with continuous features that necessitate further preprocessing or feature engineering, it could not perform as effectively (Surya, et.al, 2019).

2.7.2 Decision Tree

The decision tree is a flexible and clear machine learning method used in both classification and regression applications. It makes a decision based on the dominant class or average value in each subset after recursively segmenting the input data into subsets according to the values of the input characteristics (Al-Mhiqani et al., 2020). The result is a tree-like structure, with each leaf node denoting the final result or prediction and each internal node reflecting a judgment based on a characteristic.

How the decision tree algorithm works (Gulati, et.al, 2016) (Ingre, et.al, 2018):

1. At the root node, start with the complete dataset.
2. Select the most effective data splitting feature. based on a criterion that measures how well a characteristic distinguishes different groups or reduces the variation in the target variable, such as Gini impurity or information gain.
3. Create child nodes for each unique value of the selected feature and split the data accordingly.
4. Repeat steps 2-3 recursively for each child node until one of the following stopping conditions is met:
   - All instances in the node belong to the same class (for classification) or have the same target value (for regression).

  - The maximum tree depth is reached.
  - The minimum number of instances required for a split is not met.
5. Assign a class label or a target value to each leaf node based on the majority class or the average target value, respectively.

Decision trees are popular because they are easier to grasp and visualize, making them more interpretable than many other machine learning methods. They can handle both category and numerical variables and are relatively resilient to outliers and missing values (Gulati, et.al, 2016). However, decision trees have some downsides, including a tendency to overfit the training data, resulting in poor generalization to new data. This issue can be overcome by pruning the tree, restricting the maximum depth, or employing ensemble approaches like random forests or gradient boosting, which combine the predictions of numerous decision trees to enhance overall performance (Ingre, et.al, 2018).

2.7.3 K-Nearest Neighbor (KNN)

KNN is a simple and intuitive machine-learning technique used for classification and regression tasks. The primary notion behind KNN is that instances with comparable feature values tend to have similar output values (Dey, et.al, 2016). In other words, KNN assumes that instances that are close to each other in the feature space are likely to belong to the same class or have similar target values.

How the KNN algorithm works (Dey, et.al, 2016) (Prasatha, et.al, 2017) (Sha'Abani,, et.al, 2020):

1. Choose the number of neighbors (k) to consider. This is a hyperparameter that specifies how many nearest neighbors to take into account when creating a prediction.
2. Calculate the distances between the new, unlabeled instance and all instances in the training dataset using some distance metric, such as Euclidean, Manhattan, or Minkowski distance.
3. Identify the k training examples that are closest to the new instance.
4. Based on the majority class (for classification) or average target value (for regression) of the k closest neighbors, make a forecast.

KNN does not create an explicit model during training since it is an instance-based or lazy learning method. When a new instance has to be classified or a target value needs to be projected, it instead stores the whole training dataset and does the distance calculations and predictions immediately (Al-Mhiqani, et. al., 2020).

The benefits of KNN are its simplicity, ease of implementation, and capacity to handle noisy input. It can also adjust to changes in the data distribution if the dataset is updated over time (Prasatha, et.al, 2017). Figure 2.5 show an example of K-NN classifier.

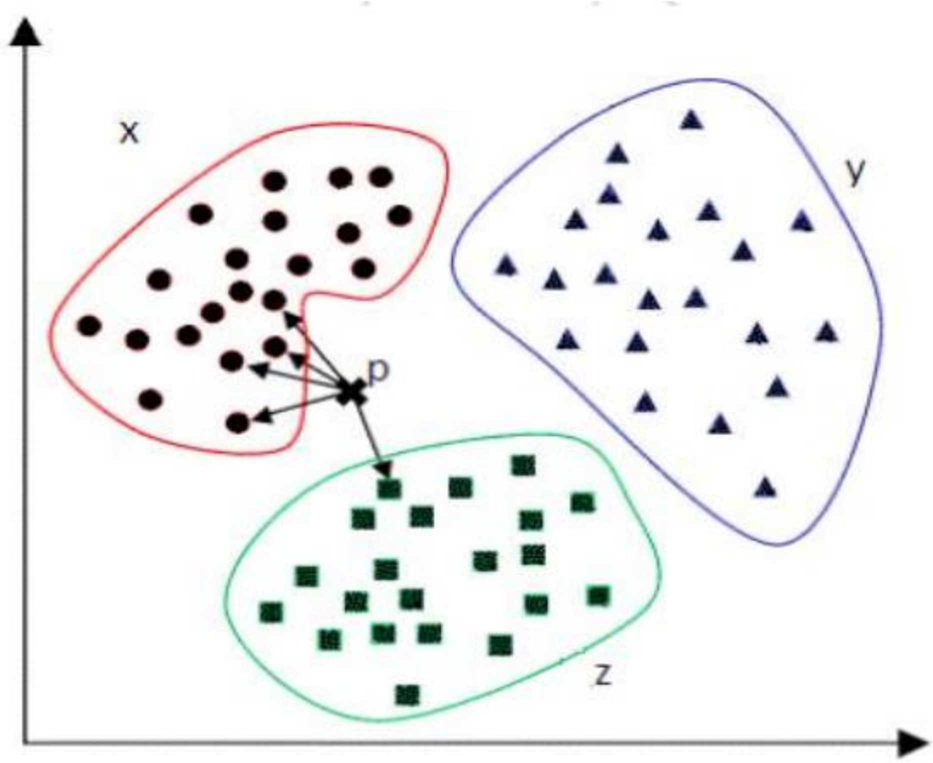


Figure 2.3 Example of K-NN classifier (Patankar, B., & Chavda, V., 2014)

However, KNN has significant downsides, such as its sensitivity to the choice of the distance measure and the value of k. Additionally, KNN can be computationally expensive for large datasets, as it needs calculating distances between the new instance and all the training examples. Scaling the features and adopting efficient data structures like k-d trees or ball trees can help relieve this difficulty. Another difficulty with KNN is that it might not operate well with high-dimensional data due to the curse of dimensionality, which makes the distance between instances less relevant. Feature selection or dimensionality reduction techniques like PCA can be utilized to address this problem (Sha'Abani,, et.al, 2020).

2.7.4 Logistic Regression

In datasets where the dependent variable is categorical and has two alternative outcomes, logistic regression is frequently utilized. The purpose of this strategy is to categorize data into one of two groups based on the predictor factors. It is very helpful for binary classification problems. By utilizing the one-vs-all or one-vs-rest technique, logistic regression can also be used to solve multinomial classification issues involving more than two categories (Bailly, et.al, 2022).

The logistic function, which converts the continuous input into a probability value, is how the logistic regression algorithm models the relationship between the dependent variable and the predictor variables. The possibility that the data point falls into one of the two categories is indicated by this probability value.

The following are some instances of datasets where logistic regression can be used (Kirasich, et.al, 2018):

- Use information about a customer's usage habits, demographics, and interactions with customer service to predict whether or not they will leave a telecom firm.
- estimating a patient's probability of developing a specific disease based on criteria like age, gender, family history, and lifestyle.
- Use attributes like keywords, email content, and metadata to categorize emails as spam or not spam.
- assessing a loan applicant's credit history, income, employment status, and loan amount to determine whether they will miss payments.

Logistic regression might be a good option for modeling the relationship between the predictor factors and the responder variable in the dataset if the objective is to predict a categorical outcome with two possible values or to perform binary classification (Bailly, et.al, 2022).

2.8 Synthetic Minority Over-sampling Technique

SMOTE is a well-known method for addressing class imbalance in datasets for machine learning. Class imbalance arises when one class has disproportionately more instances than another, resulting in biased models that benefit the majority class and perform poorly on the minority class (Mansourifar, et al., 2020).

SMOTE balances the dataset by creating synthetic samples that increase the number of minority class instances. These are the stages that the algorithm takes (Tarawneh et al., 2020):

1. Select a minority class instance and locate its k-nearest neighbors (usually, k=5) in the feature space.
2. Choose a random neighbor from the k-nearest neighbors.
3. Compute the difference between the feature values of the chosen instance and the selected neighbor.
4. Multiply this difference by a random integer between 0 and 1.
5. Add the result to the feature values of the chosen instance.

This technique is continued until the desired number of synthetic samples is obtained. The generated samples are subsequently added to the original dataset, enhancing the representation of the minority class (Mansourifar, et.al, 2020).

SMOTE provides a number of benefits, including increasing dataset variety without replicating already existing instances, which can assist decrease overfitting. As it presumes that the minority class instances are situated in the same feature space as the majority class, it might not be appropriate for all unbalanced datasets. Alternative methods, including cost-sensitive learning or alternate sample procedures, could be better suitable in some circumstances (Tarawneh et al., 2020).

Figure 2.6 depicts the design for the traditional SMOTE method for a two-dimensional feature collection. In the illustration's example, a few sample points from group "A" are interpolated linearly to create additional sample points from neighboring points "b", "c", and "e". The shown synthetic sample points are all freshly synthesized samples, and a visual representation of the SMOTE synthesis's interpolation of the minority class in imbalanced emotional samples is provided.

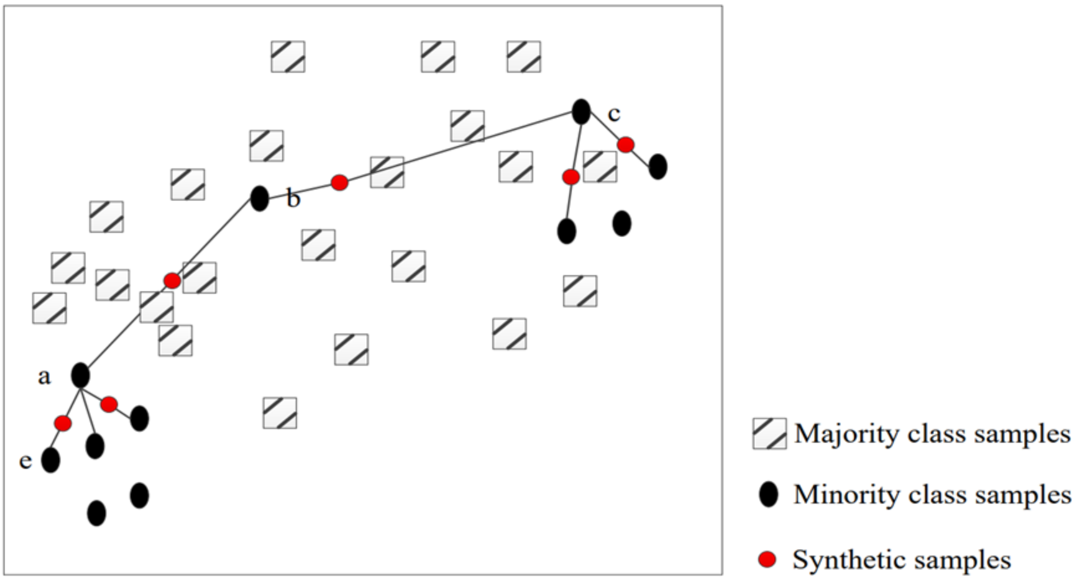


Figure 2.4 Diagram of the traditional SMOTE algorithm (Liu, et.al, 2020)

2.9 Prediction Modeling

Prediction modeling is a method in which machine learning algorithms are used to forecast future events or trends based on past data. It is frequently utilized across numerous industries, including as finance, marketing, healthcare, and sports, to generate data-driven judgments and projections (Mohamadou, et.al, 2020).

The steps involved in building a prediction model typically include (Asghar, et.al, 2016) (Banerjee, et.al, 2018) (Sayad, et.al, 2019):

1. Data collection: Gather historical data relevant to the problem you wish to address. This data should comprise both input properties (independent variables) and output values (dependent variables) that you wish to forecast.
2. Data preprocessing: Clean, preprocess, and convert the data to a suitable format for the chosen machine learning technique. This may entail managing missing data, scaling features, encoding category variables, or decreasing dimensionality.
3. Feature selection: Identify the most significant traits that aid in producing accurate forecasts. This may be done using techniques like correlation analysis, recursive feature reduction, or regularization methods like Lasso.

4. Model selection: Based on the nature of the problem, the type of data, and the desired level of interpretability, pick the appropriate machine learning algorithm(s) for the prediction task. Common algorithms include support vector machines, decision trees, neural networks, and linear regression.
5. Model training: Train the selected algorithm(s) on the preprocessed data by feeding it the input characteristics and output values, allowing the algorithm to understand the underlying patterns and connections in the data.
6. Model evaluation: Evaluate the trained model's performance using the relevant evaluation metrics, such as accuracy, precision, recall, and F1-score for classification tasks or mean squared error for regression tasks. The dataset is frequently divided into training and testing sets, or cross-validation methods are used, to do this.
7. Model tuning: Tweak the model's hyperparameters to enhance output. Approaches like grid search, random search, or Bayesian optimization can be used for this.
8. Model deployment: Use the trained and improved model in a real-world setting to generate predictions about brand-new, ambiguous data.
9. Model maintenance: Continuously evaluate the model's performance and update it with new data as needed to guarantee its accuracy and relevance.

Prediction modeling allows organizations to leverage historical data to make informed decisions, improve processes, and optimize resources. However, it's important to remember that prediction models are not perfect and should be used as a tool to support decision-making rather than as the sole basis for decisions (Mohamadou, et.al, 2020).

2.10 Evaluation Metrics

Evaluation metrics are used to measure the performance of machine learning models, notably in classification tasks. A brief explanation of each metric coupled with a mathematical example (Mesaros, 2018) (Luque, et.al, 2019) (Zeng, et.al, 2020):

2.10.1 Precision

Precision is the fraction of true positive predictions (TP) out of all positive predictions generated by the model. It assesses how effectively the model properly recognizes positive cases (Mesaros, 2018):

Precision = TP / (TP + FP) ……………. 2-5

Example: If a model produces 100 positive predictions, and 80 of them are actual positives, the accuracy is 0.8 (80 / 100).

2.10.2 Accuracy

Accuracy is the fraction of correct predictions (including true positives and true negatives) out of all predictions generated by the model. It gauges the overall accuracy of the model (Luque, et.al, 2019):

Accuracy = (TP + TN) / (TP + TN + FP + FN) ……………. 2-6

Example: If a model makes 200 predictions in total, with 80 true positives, 90 true negatives, 20 false positives, and 10 false negatives, the accuracy is 0.85 [(80 + 90) / 200].

2.10.3 Recall

Recall, also known as sensitivity or true positive rate, is the fraction of true positive forecasts out of all real positive cases. It assesses the model's ability to detect good cases (Zeng, et.al, 2020).

Recall = TP / (TP + FN) ……………. 2-7

Example: If there are 100 actual positive instances, and the model correctly identifies 80 of them, the recall is 0.8 (80 / 100).

2.10.4 F1-score

F1-score is the harmonic mean of accuracy and recall. It gives a balanced measure between accuracy and recalls when working with unbalanced datasets (Zeng, et.al, 2020).

F1-score = 2 * (Precision * Recall) / (Precision + Recall) ……………. 2-8

Example: If the precision is 0.8 and recall is 0.9, the F1-score is 0.848 [(2 * 0.8 * 0.9) / (0.8 + 0.9)].

2.10.5 Detection Rate (DR)

The detection Rate is the same as recall, which quantifies the fraction of genuine positive predictions out of all real positive cases (Mesaros, 2018).

2.10.6 False Alert Rate (FAR)

False Alert Rate, also known as false positive rate, is the fraction of erroneous positive predictions out of all real negative cases. It assesses the model's likelihood of mistakenly recognizing negative cases as positive (Luque, et.al, 2019).

$$FAR = FP / (FP + TN) \text{ …………….. } 2\text{-}9$$

Example: If there are 100 actual negative instances, and the model incorrectly identifies 20 of them as positive, the FAR is 0.2 (20 / 100).

2.10.7 Error rate

Error rate is the fraction of inaccurate predictions (including false positives and false negatives) out of all predictions generated by the model. It's the complement of accuracy (Luque, et.al, 2019) (Zeng, et.al, 2020).

$$\text{Error rate} = (FP + FN) / (TP + TN + FP + FN) \text{ …………….. } 2\text{-}10$$

Example: In the accuracy example above, the error rate would be 0.15 [(20 + 10) / 200].

CHAPTER THREE

The Proposed Model

Chapter Three
The Proposed Model

3.1 Introduction

This chapter explains the steps of implementing the model suggested for prediction problems in small datasets applications. The first part of the suggested model is the data processing by applying features selection, ranking and correlation. The second part Applying multiple algorithms such as the Decision Tree, Naïve Bayes and KNN. As well as using azimuths to collect several results for analysis and evaluation.

3.2 Methodology

The suggested method for finishing this study begins with the download of an open-source UCI data collection. After the dataset has been verified, the next stage is preprocessing, which includes data cleaning, data transformation, data reduction, binning, and select attributes. and utilized the correlation, prediction model, and ranking feature. The primary approach, feature selection, is used after all these techniques have been applied to the downloaded dataset. Later, the data is subjected to the applications of Decision Tree, Naive Bayes, and KNN. Following application of algorithms and strategies, we compare the outcomes and talk about the conclusion. The flow of these techniques is shown in Figure 3.1.

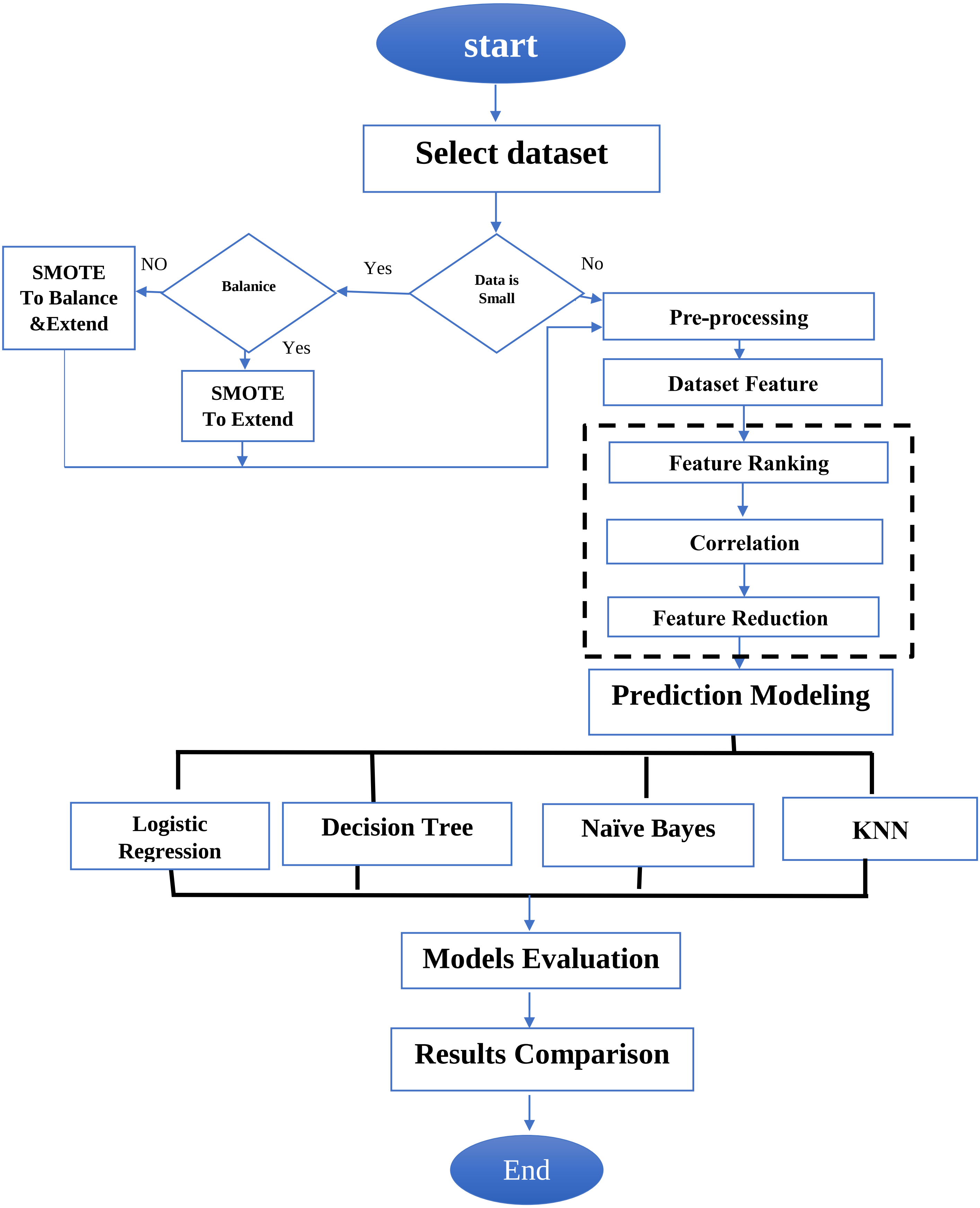


Figure 3.1 The flow chart for the proposed model

3.3 Select Dataset

To apply the proposed model, three types of small datasets are utilized. These datasets are analyzed and processed to be used in prediction process. These datasets are:

3.3.1 Heart Disease Dataset

Predicting heart disease is a significant issue that can be helped by machine learning approaches. However, it might be difficult to forecast cardiac disease when dealing with a little sample. A heart disease dataset downloaded from:
https://www.kaggle.com/datasets/rishidamarla/heart-disease-prediction
This dataset is composed of 270 records and 15 attributes; it is important to improve the chances of creating a machine learning model for heart disease prediction that works well on a small dataset. However, it's crucial to remember that working with a small dataset has its drawbacks and that the model's performance could be constrained by the quantity of data available.

3.3.2 Lung Cancer Dataset

The dataset for lung cancer is frequently utilized in studies and analyses of the disease. This information is used by researchers and medical professionals to investigate the causes of lung cancer, evaluate treatment results, pinpoint risk factors, and create novel treatments for the condition. Predictive models that aid in early diagnosis and prognosis prediction can also be created using the dataset. lung cancer dataset is extremely important to the field of lung cancer research in terms of knowledge advancement and patient care improvement. It contains 309 records and 15 attributes. A lung cancer dataset is downloaded from:
https://www.kaggle.com/datasets/jillanisofttech/lung-cancer-detection

3.3.3 Cancer Patients Dataset

The cancer patient dataset can be used for various tasks, such as predicting the survival outcomes of cancer patients based on their demographics, tumor characteristics, and treatment information. It can also be used for identifying risk factors associated with cancer, evaluating the effectiveness of different treatments, and developing new diagnostic or therapeutic strategies. It contains 1000 records with 26 attributes. A Cancer patient dataset downloaded from:
https://www.kaggle.com/datasets/thedevastator/cancer-patients-and-air-pollution-a-new-link

3.4 Data Preprocessing

To achieve best results from the prediction model, the data format must be in a proper manner. The data set should be formatted in such a way that more than one algorithm is executed in one data set, and best out of them is chosen. In this thesis datasets, Data Transformation which involves converting the data into a suitable format for analysis is implemented with normalization. Normalization is used to scale the data to a common range. Algorithm 3.1 shows the Preprocessing Dataset.

Algorithm 3.1 Preprocessing Dataset.

```
Input:  Two dimensional array Dataset [$n*m$]
Output:  Processed Dataset.
Step 1:  Checking the Missing Values of Dataset.
Step 2: transforming Date nominal feature to numeric feature
Step 3: Normalization of Dataset
End.
```

```
Input:  Two dimensional array Dataset [$n*m$]
Output:  Processed Dataset.
Step 1:  Checking the Missing Values of Dataset.
Step 2: transforming Date nominal feature to numeric feature
Step 3: Normalization of Dataset
```

3.5 Features Selection

Feature selection on a small dataset can be challenging, but there are still several effective approaches can be used on the selected dataset. In the proposed model, important features will be selected accordingly:

- Correlation-based feature selection: calculated the correlation coefficient between each pair of features in each dataset, and selected only those features that have high correlation Either between two features or between the features and the target.
- Recursive feature elimination: involves training the proposed model on all features in dataset, and then eliminating the feature with the lowest

importance score. then repeat the process until reaching only the most important features in the dataset.

The types and number of features that are eliminated differ according to the type of dataset used to train the model. For example, there are many important features that can cause heart disease, and also there are many important features that can cause general cancer diseases, but these features are less effective in determining the type of cancer, such as lung cancer, which is affected by factors associated with the lung. In the same vein, the most important features of the dataset vary according to its type. For example, pressure is considered the most important features for heart diseases, and smoking is the most important features for lung cancer diseases. These are the most important features that are useful in knowing the prediction of dataset.

Algorithm 3.2 shows the Feature Selection process.

Algorithm 3.2 Feature Selection

```
Input: Dataset[n*m].
Output:  features for Dataset[n*m].
  for i = 1 to m
       for j = 1 to n
           Call correlation coefficient Method R[i], R[j]
           if R[i] >R[j]
Select and Order the features (Fi) with maximum correlation (Ri)
The selected features are sorted by the correlation coefficient
values in descending order
         end if
      end for j
  end for i
End.
```

3.6 Features Ranking

Features ranking on a small dataset can be challenging as there may not be enough data to adequately capture the relationships between the features. The data will be arranged in the dataset examples according to the following steps:

- Determine the target of the dataset: Before arranging the dataset, it is important to determine the target of the dataset and which features are the most important for achieving that target.
- Identify the features: Identify all the features in the dataset and determine which features are the most important.
- Prioritize the features: Prioritize the features based on how significant they are in achieving the target of the dataset.
- Sort the dataset based on the prioritized features
- After reviewing each dataset, and following the method of arranging the features, it will be noted that each dataset differs in the order and number of its features based on the type of data and the importance of each feature and the impact of its presence on the prediction accuracy of the dataset.

Overall, feature ranking on small datasets requires a careful balance between statistical methods and expert judgment.

It is possible to implement the feature ranking manually or through mathematical statistics through multiple methods, also by programs designed for this purpose or programming languages. In the proposed model, the function **sort_values()** was used (which is one of the functions in the Python libraries that performs the process of feature ranking automatically when it is called to any data) in order to ranking the features for each dataset and note the difference in the ranking of each of them. Algorithm 3.1 shows how to implement the feature ranking using a function in Python.

3.7 Correlation

When dealing with small datasets, it may be challenging to apply correlation as the analysis requires a large number of data points to detect a meaningful relationship between features.

In the proposed model, the following steps will apply to perform the correlation on the small datasets:

- Arrange the features carefully: Careful and accurate arrange of the features helps to reduce measurement errors, which can cause false correlations.
- Check for outliers: Since small datasets are susceptible to outliers, it is important to check for them and either remove or address them.
- Choosing the method of calculating the correlation: To perform the correlation and extract the results, it is done through several methods (described in Section 2.4.3). Each method has its own mathematical equations shown in 2.1, 2.2, and 2.3.
- Interpreted the results: Interpreted the results cautiously, If the relationship between the two features is a 0 relationship, this means that one of the features can be replaced by the other, while if the relationship is weak, this means that one of the features cannot be replaced by the other.
- To study and analyzing the relation and the effects of each feature on the others, the process starts by selecting the one feature and test its correlation coefficient with the second, third, …… till the last one and so on for other features, and then eliminated one of the strong features.
- Then the correlation is calculated between each feature and the target, and then eliminated the weak feature.

In the proposed model, the function **corr()** was used (which is one of the functions in the Python libraries that calculate and extract the correlation automatically when it is called to any data) in order to extract the correlation for each dataset. Algorithm 3.3 shows how to implement the correlation.

Algorithm 3.3 correlation coefficient Method in python

```
Input:Two dimensional array Dataset [n*m]
Output: correlation coefficient for Dataset [n*m]
for $i = 1$ to n
    for $j = 1$ to m
Compute the correlation coefficient betweenfeature Fi and
feature Fj according to the equation (2.1)
    end for j
end for i
END
```

3.8 Check Balance and Imbalance Dataset

Balance in a dataset is crucial for unbiased analysis, accurate predictions, robust models, and generalizability to real-world scenarios. It makes ensuring that various groups or categories are fairly represented, which produces insights and decision-making processes that are more trustworthy. In the proposed model, the data set that shows imbalance is balanced to make the proposed model more efficient and accurate in prediction. To do this, the following two steps are followed:

- Understand the dataset: it is important to understand the features of the dataset before proceeding with balancing or imbalance techniques.
- Collected more data: collected more data to balance the dataset. In this model, the SMOTE principle will be used to balance the dataset.

3.9 Synthetic Minority Over-Sampling Technique (SMOTE)

The SMOTE can be applied to a small dataset when the dataset has a class imbalance problem. If the dataset has too few instances of a certain class, SMOTE could be used to artificially increase the number of instances of that class. The following steps will be applied SMOTE to the small dataset:

- Splitting the dataset into two parts: a training set and a test set.
- Applying SMOTE only to the training set. This means that the synthetic samples generated through SMOTE will not be added to the test set.
- Once SMOTE has been applied to the training set, train the classification model.
- Testing the model on the test set.

To illustrate how SMOTE works, suppose the following data points:

Class A (majority class):

Point A1: (2, 3)5

Point A2: (4, 2)

Point A3: (3, 1)

Class B (minority class):

Point B1: (1, 2)

applying SMOTE with k = 1 (1 nearest neighbor) to generate one synthetic sample for the minority class.

Step 1: Calculate the distance between point B1 and its nearest neighbor(s) in the minority class. In this case, there's only one point in the minority class, so we'll find its nearest neighbor in the majority class.

Euclidean distance between B1 and A1: $\sqrt{((1-2)^2 + (2-3)^2)} = \sqrt{2}$

Euclidean distance between B1 and A2: $\sqrt{((1-4)^2 + (2-2)^2)} = \sqrt{9} = 3$

Euclidean distance between B1 and A3: $\sqrt{((1-3)^2 + (2-1)^2)} = \sqrt{5}$

The nearest neighbor to point B1 is point A1 with a distance of $\sqrt{2}$.

Step 2: Generate the synthetic sample. We'll randomly choose a value for the interpolation factor r between 0 and 1. Let's say r = 0.5 for this example.

New synthetic point coordinates:

x = x1 + r * (x2 - x1) = 1 + 0.5 * (2 - 1) = 1.5

y = y1 + r * (y2 - y1) = 2 + 0.5 * (3 - 2) = 2.5

The synthetic point generated by SMOTE is (1.5, 2.5).

Now, the updated dataset looks like this:

Class A (majority class):

Point A1: (2, 3)

Point A2: (4, 2)

Point A3: (3, 1)

Class B (minority class):

Point B1: (1, 2)

Synthetic Point B2: (1.5, 2.5)

By applying SMOTE, we've generated a synthetic sample for the minority class, helping to balance the dataset and improve classification performance.

It is important to note that SMOTE should only be applied when the dataset has a class imbalance problem. If the dataset does not have a class imbalance problem, SMOTE can cause overfitting, which can negatively impact the performance of the model.

There are multiple programs to apply the SMOTE on the dataset to balance them, one of those programs is applied to implement the SMOTE on the three types of datasets that were taken in this thesis to implement the proposed model, two of these datasets showed the need for balancing, while the third dataset showed the balance and did not need to use the SMOTE.

3.10 Prediction Modeling

There are many ways to predict the data, in this proposed model using the method that depends on the use of algorithms Naive Bayes, Decision Tree, KNN, and Logistic Regression:

3.10.1 Naive Bayes Algorithm

Prediction modeling using the Naive Bayes algorithm can be a good choice on a small dataset for several reasons:

- Naive Bayes is a simple and efficient algorithm that requires a small amount of training data to make accurate predictions.
- Naive Bayes assumes that features are independent of each other, which makes it easy to handle small datasets where the number of features is limited.
- Naive Bayes has been shown to work well on small and medium datasets, particularly in text classification and spam filtering.

In the proposed model, the following steps are used to apply the Naive Bayes algorithm on the dataset:

- ➢ splitting dataset into training and testing sets.
- ➢ Train the Naive Bayes model on the training data. This involves calculating the probability of each class and each feature given a class.
- ➢ Use the trained model to make predictions on the testing data. The model calculates the probability of each test instance belonging to each class and assigns the instance to the class with the highest probability.
- ➢ It's important to note that Naive Bayes may not perform well on datasets with highly correlated features or noisy data. In such cases, other algorithms like decision trees or support vector machines may be more appropriate.

The Naive Bayes algorithm is implemented on the dataset through the **GaussianNB()** function (which is one of the functions in the Python libraries to implement the Naive Bayes algorithm when it is called to any data). Algorithm 3.4 describes the steps used in the Python programming language to implement the Naive Bayes algorithm on Dataset using its libraries and functions, While the Algorithm 3.5 represents the steps of the programming algorithm in Python.

Algorithm 3.4 implementing of the Naive Bayes algorithm

```
Input: dataset D
Output: x
Begin
Step1: dataset Dataset → Matrix (x, y)
Step2: Split Dataset by Training and Testing
Step3: Training → x_train, y_train
          Testing → x_test, y_test
Step4: model → function GaussianNB()
Step5: y_predicted → model.fit (Tr).x_predicted
Step6: Accuracy_Score(y_test, y_predicted)
End
```

Algorithm 3.5 The Naive Bayes Algorithm

```
Input: Training data
Output: Class of a test sample x
for q = 1... w // loop for each mining models element
μ[q] = 0; // initialization of mining models elements
end for;
for j = 1... m // loop for each row
μ[d[j,p]]++; // increment number of row for value xj.p of object xj;
for k = 1 ... p-1 // loop for each column
μ[(k-1)+(d[j, k]-1)-(0)+ d[j, p]]++; // increment number of rows with value xj.k // and value xj.p, where o(k)=s+4=1(Tq|·s)
end for;
end for;
```

3.10.2 Decision Tree Algorithm

Decision Tree algorithm can be effective in predicting outcomes on small datasets. They are a type of supervised learning algorithm that uses a tree-like structure to make decisions about the datasets.

In the proposed model, the following steps are used to apply the Decision Tree algorithm on the dataset:

- creates a tree model where each node represents a decision point, and the edges represent the outcomes or results of that decision.
- splitting the data into smaller subsets based on the attributes, and it continues to make decisions by splitting the subsets until it reaches a point where it can make a prediction.
- to avoid overfitting, it is essential to use techniques such as pruning and cross-validation.

The Decision Tree algorithm is implemented on the dataset through the **DecisionTreeClassifier()** function (which is one of the functions in the Python libraries to implement the Decision Tree algorithm when it is called to any data). The algorithm 3.6 describes the steps used in the Python programming language to implement the Decision Tree algorithm on Dataset using its libraries and functions, While the Algorithm 3.7 represents the steps of the programming algorithm in Python.

Algorithm 3.6 implementing of the Decision Tree algorithm on the dataset

```
Input: dataset, y
Output: x
Begin
Step1: dataset Dataset → Matrix (x, y)
Step2: Split Dataset by Training and Testing
Step3: Training → x_train, y_train
          Testing → x_test, y_test
Step4: model → function DecisionTreeClassifier()
Step5: y_predicted → model.fit (Tr).x_predicted
```

```
Step6: Accuracy_Score(y_test, y_predicted)
End
```

Algorithm 3.7 The Decision Tree Algorithm

```
Input: Training data
Output: Class of a test sample x
Create a root node N;
IF (T belongs to same category C)
{leaf node=N;
Mark N as class C;
Return N;
}
For i=1 to n
{Calculate Information_gain (Ai);}
ta= testing attribute;
N.ta = attribute having highest information_gain;
if (N.ta== continuous)
{find threshold;}
For (Each T in splitting of T)
if (T is empty)
{child of N is a leaf node;}
else
{child of N= dtree T)}
calculate classification error rate of node N;
return N;
```

3.10.3 KNN Algorithm

KNN works on the principle of finding the nearest K neighbors in the data set and making predictions based on their classifications and values. In this proposed model, use KNN algorithm for prediction modeling on a small dataset. Small datasets pose a challenge when it comes to prediction modeling, particularly with the KNN algorithm, as it requires a good amount of data for accurate predictions. However, can apply the KNN algorithm on small datasets using the following steps:

- ensuring that the dataset is clean and properly formatted: This has been done by implementing the previous steps: feature selection, ranking, and correlation.
- splitting the small dataset into training and testing sets to validate the accuracy of the predictions. A commonly used ratio is 80:20, where 80% of the data is used for training and 20% is used for testing.
- The value of K determines the number of nearest neighbors that will be used to make predictions. Will need to experiment with different values of K to determine the optimal value that provides the best accuracy.

The KNN algorithm is implemented on the dataset through the **KNeighborsClassifier()** function (which is one of the functions in the Python libraries to implement the KNN algorithm when it is called to any data). The algorithm 3.8 describes the steps used in the Python programming language to implement the KNN algorithm on Dataset using its libraries and functions, While the Algorithm 3.9 represents the steps of the programming algorithm in Python.

Algorithm 3.8 implementing of the KNN algorithm on the dataset

```
Input: dataset, y
Output: x
Begin
Step1: dataset Dataset → Matrix (x, y)
Step2: Split Dataset by Training and Testing
Step3: Training → x_train, y_train
       Testing → x_test, y_test
```

```
Step4: model → function KNeighborsClassifier()
Step5: y_predicted → model.fit (Tr).x_predicted
Step6: Accuracy_Score(y_test, y_predicted)
End
```

Algorithm 3.9 The KNN Algorithm

```
Input: X: training data, Y: class labels of X, K: number of nearest neighbors.
Output: Class of a test sample x.
Start
Classify (X,Y,x)
for each sample x do
Calculate the distance: $d(x,X) = \sqrt{}(x_i — X_i)^2$ end for
Classify x in the majority class: C(x) = argmax; $Ex_1$€™NN C(Xj, Yx)
End
```

3.10.4 Logistic Regression

Applying the Logistic Regression for the proposed prediction modeling on the datasets is illustrated in the following steps:

- Create training and testing sets from the dataset. The testing set will be used to assess the performance of the Logistic Regression model after it has been constructed using the training set.
- Specifying the independent variables (features) and the dependent variable (target).
- Create an instance of the LogisticRegression class by importing it from sklearn.linear_model. Using the fit() method, fit the model to the training data.
- generate predictions: Using the predict() method, use the trained model to generate predictions about the testing set.

Note: Understanding the fundamental ideas and presumptions of Logistic Regression as well as being aware of its limits is crucial. Additionally, additional preparation or model modifications may be necessary when working with unbalanced datasets or highly dimensional data.

The Logistic Regression is implemented on the dataset through the **LogisticRegression()** function (which is one of the functions in the Python libraries to implement the Logistic Regression when it is called to any data). The algorithm 3.10 describes the steps used in the Python programming language to implement the Logistic Regression on Dataset using its libraries and functions, While the Algorithm 3.11 represents the steps of the programming algorithm in Python.

Algorithm 3.10 implementing of the Logistic Regression on the dataset

```
Input: dataset, y
Output: x
Begin
Step1: dataset Dataset → Matrix (x, y)
Step2: Split Dataset by Training and Testing
Step3: Training → x_train, y_train
        Testing → x_test, y_test
Step4: model → function LogisticRegression()
Step5: y_predicted → model.fit (Tr).x_predicted
Step6: Accuracy_Score(y_test, y_predicted)
End
```

Algorithm 3.11 The Logistic Regression Algorithm

```
Input: Training data
Output: Class of a test sample x
Begin
For i = 1 to k
For each training data instance d¡.
Set the target value for the regression to z; = yi-P(1d) [P(1|d;)(1−P(1|dj))]
Initialize the weight of instance dj to [P(1d;)(1 − P(1d;))]
Finalize a f(j) to the data with class value (Z;) and weight (w;)
Classical label decision
Assign (class label: 1) if Pid > 0.5, otherwise (class label: 2)
End
```

CHAPTER FOUR

Result and Discussions

Chapter Four
Results and Discussions

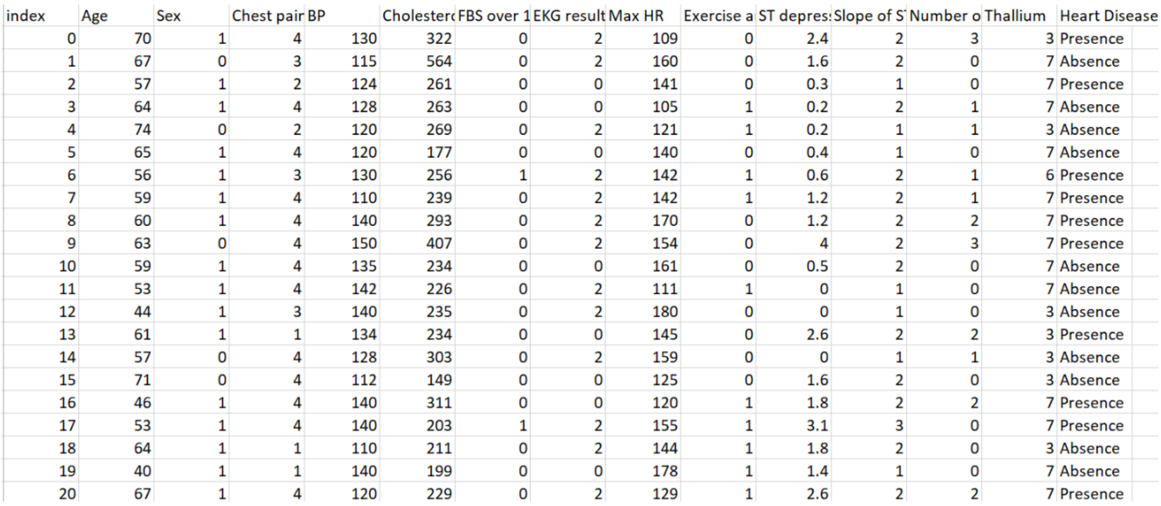
4.1 Introduction

In this chapter, the results obtained from the application of the proposed model will be presented by applying the steps as mentioned in Chapter 3. The results are for the two datasets that were taken to implement the model. Then these results are analyzed and compared with other proposed works.

4.2 Heart Disease Prediction Dataset

This dataset contains 14 attributes and 270 record. Heart_Disease_prediction to identify which patients are most likely to have a cardiac condition soon, a dataset is employed. Table 4.1 present sample from the dataset.

Table 4.1 Heart_Disease prediction dataset sample



index	Age	Sex	Chest pair	BP	Cholestero	FBS over 1	EKG result	Max HR	Exercise a	ST depres	Slope of S	Number o	Thallium	Heart Disease
0	70	1	4	130	322	0	2	109	0	2.4	2	3	3	Presence
1	67	0	3	115	564	0	2	160	0	1.6	2	0	7	Absence
2	57	1	2	124	261	0	0	141	0	0.3	1	0	7	Presence
3	64	1	4	128	263	0	0	105	1	0.2	2	1	7	Absence
4	74	0	2	120	269	0	2	121	1	0.2	1	1	3	Absence
5	65	1	4	120	177	0	0	140	0	0.4	1	0	7	Absence
6	56	1	3	130	256	1	2	142	1	0.6	2	1	6	Presence
7	59	1	4	110	239	0	2	142	1	1.2	2	1	7	Presence
8	60	1	4	140	293	0	2	170	0	1.2	2	2	7	Presence
9	63	0	4	150	407	0	2	154	0	4	2	3	7	Presence
10	59	1	4	135	234	0	0	161	0	0.5	2	0	7	Absence
11	53	1	4	142	226	0	2	111	1	0	1	0	7	Absence
12	44	1	3	140	235	0	2	180	0	0	1	0	3	Absence
13	61	1	1	134	234	0	0	145	0	2.6	2	2	3	Presence
14	57	0	4	128	303	0	2	159	0	0	1	1	3	Absence
15	71	0	4	112	149	0	0	125	0	1.6	2	0	3	Absence
16	46	1	4	140	311	0	0	120	1	1.8	2	2	7	Presence
17	53	1	4	140	203	1	2	155	1	3.1	3	0	7	Presence
18	64	1	1	110	211	0	2	144	1	1.8	2	0	3	Absence
19	40	1	1	140	199	0	0	178	1	1.4	1	0	7	Absence
20	67	1	4	120	229	0	2	129	1	2.6	2	2	7	Presence

4.2.1 Small Dataset Results

After applying the steps of the proposed model, have the following results:

Step1: Attribute Ranking

After applying the proposed model using python programing language by ranking the small dataset (heart disease) and observing its attributes. It composed of 14 attributes. Table 4.2 presents the sorted list of the ranked attributes:

Table 4.2 attributes ranking for the small heart disease dataset

Rank	Attribute name	Old No.
1	Thallium	14
2	Number of vessels fluor	13
3	Exercise angina	10
4	Max HR	9
5	Depression St	11
6	Chest pain type	4
7	Slope of St	12
8	sex	3
9	age	2
10	EKG results	8
11	Bp	5
12	Cholesterol	6
13	FBS over120	7
14	Index	1

Table 4.2 shows that the attribute ranking of the extended data is very close that for the small dataset.

Step2: Attributes Correlation

The correlation results are recorded in a matrix of (14 x 14). The strong correlation coefficient value (which is greater or equal 0.7) is selected and attended. Table 4.3 shows a sample of the created correlation matrix.

Table 4.3 correlation matrix for the small heart disease dataset

	Index	Age	Sex	Chest pain type	Bp	Chole sterol	FBS over1 20	EKG results	Max HR	Exercise angina	Depressi on St	Slope of St
Thallium	0.01	0.10	0.39	0.263	0.132	0.028	0.049	0.0073	0.253	0.321	0.324	0.284
Number of vessels fluor	0.06	0.35	0.08	0.226	0.08 57	0.127	0.124	0.114	0.265	0.153	0.255	0.109
Slope of St	0.02	0.16	0.05	0.137	0.14	0.005	0.044	0.161	0.387	0.256	0.61	
Depression St	0.08	0.19	0.09	0.167	0.22	0.027	0.025	0.12	0.349	0.275		0.61
Exercise angina	0.06	0.09	0.18	0.353	0.08	0.078	0.004	0.095	0.381		0.275	0.256
Max HR	0.01	0.40	0.07	0.318-	0.03	0.018	0.025	0.074		0.381-	0.349-	0.387-
EKG results	0.03	0.12	0.03	0.074	0.11	o.168	0.053		0.074	0.0951	0.12	0.161
FBS over120	0.01	0.12	0.04	0.098	0.15	0.025		0.0535	0.022	0.0041	0.0255-	0.044
Cholesterol	0.04	0.22	0.20	0.09	0.17		0.025	o.168	0.018	0.0782	0.0277	0.005
Bp	0.04	0.27	0.06	0.043		0.173	0.156	0.116	0.039	0.0828	0.223	0.142
Chest pain type	0.01	0.09	0.03		0.04	0.090	0.098	0.0743	0.318	0.353	0.167	0.137
sex	0.04	0.09		0.034	0.06	0.202	0.042	0.0393	0.076	0.18	0.0974	0.050
age	0.04		0.09	0.09	0.27	0.22	0.123	0.128	0.402	0.098	0.194	0.16
index		0.04	0.04	0.01	0.04	0.04	0.01	0.03	0.01	0.06	0.08	0.02

Number of vessels fluor	Thallium
0.06	0.01
0.356	0.106
0.0868	0.391
0.226	0.263
0.0857	0.132
0.127	0.0288
0.124	0.0492
0.114	0.0073
0.265-	0.253-
0.153	0.321
0.255	0.324
0.109	0.284
	0.256
0.256	

Step3: Predictive Modeling

To perform a predictive modeling approach, four prediction models are utilized in this thesis (Naïve bayes, Decision tree, KNN and Logistic).

Table 4.4 presents the specific measures for each of the used prediction models. These measures are (precision, Recall, F-Measure, accuracy) detailed by class.

Table 4.4 specific measures for each of the used prediction models for the small heart disease dataset

Algorithm type	Precision	Recall	F- score	accuracy	Class
Naïve Bayes	0.85 0.81 0.83 0.83	0.88 0.88 0.82 0.83	0.86 0.86 0.83 0.83	0.83	Presence (1) Absence (0) Macro avg Weighted Avg
Decision tree	0.54 0.81 0.67 0.71	0.75 0.62 0.68 0.67	0.63 0.70 0.66 0.67	0.69	Presence (1) Absence (0) Macro avg Weighted Avg
KNN	0.55 0.72 0.63 0.65	0.57 0.70 0.63 0.65	0.56 0.71 0.63 0.65	0.64	Presence (1) Absence (0) Macro avg Weighted Avg
Logistic	0.75 0.91 0.83 0.86	0.83 0.86 0.85 0.85	0.89 0.79 0.84 0.85	0.85	Presence (1) Absence (0) Macro avg Weighted Avg

Table 4.5 presents the specific measures (Accuracy, Error, precision, Recall) for each of the used prediction models.

Table 4.5 presents the specific measures (Accuracy, Error, precision, Recall) for the small heart disease dataset

Algorithm type	Accuracy	Error	Precision	Recall
Naïve Bayes	0.83	0	0.87	0.84
Decision tree	0.69	0.3	0.73	0.60
KNN	0.64	0.3	0.75	0.54
Logistic	0.85	0	0.83	0.75

Confusion matrices are a way to evaluate the performance of a classification model by analyzing the number of true positives, true negatives, false positives, and false negatives. For each of prediction models can be evaluated using confusion matrices as shown in Tables 4.6, 4.7, 4.8 and 4.9.

Table 4.6 Confusion Matrixes (Naïve Bayes) for the small heart disease dataset

	Absence (0)	Presence (1)
Absence (0)	17	4
Presence (1)	5	28

Table 4.7 Confusion Matrixes (Decision tree) for the small heart disease dataset

	Absence (0)	Presence (1)
Absence (0)	20	6
Presence (1)	11	17

Table 4.8 Confusion Matrixes (KNN) for the small heart disease dataset

	Absence (0)	Presence (1)
Absence (0)	23	9
Presence (1)	10	12

Table 4.9 Confusion Matrixes (Logistic Regression) for the small heart disease dataset

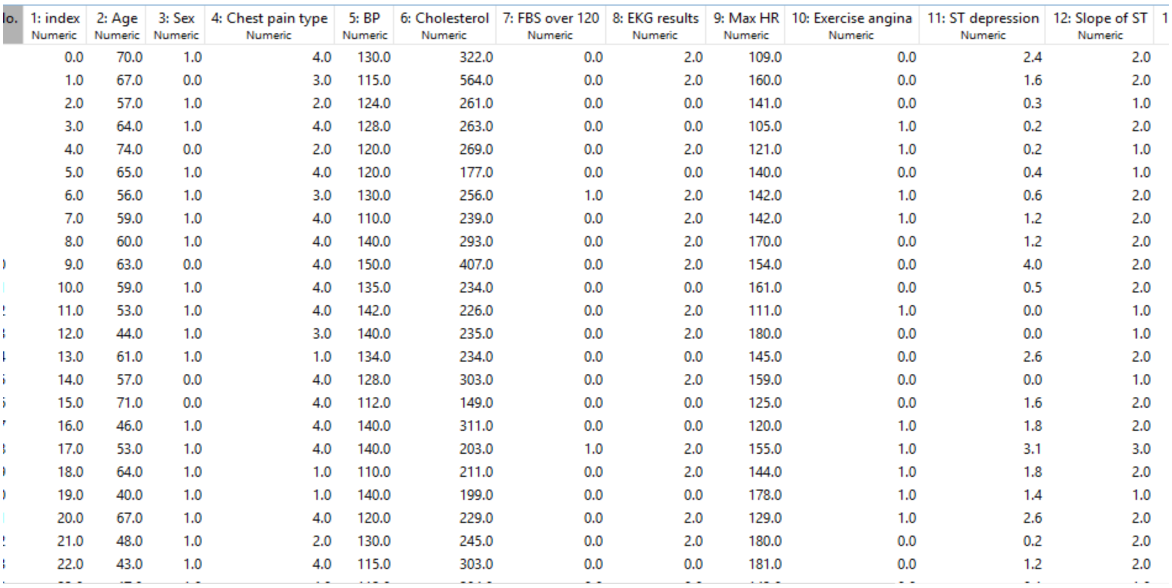
	Absence (0)	Presence (1)
Absence (0)	31	3
Presence (1)	5	15

After obtaining the results of the algorithms, tried to delete some of the least important features according to the ranking in a Table 4.2, and then monitor the effect of the deletion on each algorithm accuracy. Where when deleting (from rank 7 to 14 features) in Naïve Bayes and Logistic, index feature in KNN, and (rank 8 to 14 features) in Decision tree not effected on dataset accuracy. While remaining features when removing it the accuracy became less.

4.2.2 Data Extension Result

After applying the proposed model to the dataset, the results showed an imbalance of the dataset, so the dataset will be enlarged. A Synthetic Minority Oversampling Technique (SMOTE) is used to generate new data to extend the small dataset as the features expanded from 270 to 1080. Table 4.10 Present a sample from the generated data.

Table 4.10 a sample of the generated data.



1: index Numeric	2: Age Numeric	3: Sex Numeric	4: Chest pain type Numeric	5: BP Numeric	6: Cholesterol Numeric	7: FBS over 120 Numeric	8: EKG results Numeric	9: Max HR Numeric	10: Exercise angina Numeric	11: ST depression Numeric	12: Slope of ST Numeric
0.0	70.0	1.0	4.0	130.0	322.0	0.0	2.0	109.0	0.0	2.4	2.0
1.0	67.0	0.0	3.0	115.0	564.0	0.0	2.0	160.0	0.0	1.6	2.0
2.0	57.0	1.0	2.0	124.0	261.0	0.0	0.0	141.0	0.0	0.3	1.0
3.0	64.0	1.0	4.0	128.0	263.0	0.0	0.0	105.0	1.0	0.2	2.0
4.0	74.0	0.0	2.0	120.0	269.0	0.0	2.0	121.0	1.0	0.2	1.0
5.0	65.0	1.0	4.0	120.0	177.0	0.0	0.0	140.0	0.0	0.4	1.0
6.0	56.0	1.0	3.0	130.0	256.0	1.0	2.0	142.0	1.0	0.6	2.0
7.0	59.0	1.0	4.0	110.0	239.0	0.0	2.0	142.0	1.0	1.2	2.0
8.0	60.0	1.0	4.0	140.0	293.0	0.0	2.0	170.0	0.0	1.2	2.0
9.0	63.0	0.0	4.0	150.0	407.0	0.0	2.0	154.0	0.0	4.0	2.0
10.0	59.0	1.0	4.0	135.0	234.0	0.0	0.0	161.0	0.0	0.5	2.0
11.0	53.0	1.0	4.0	142.0	226.0	0.0	2.0	111.0	1.0	0.0	1.0
12.0	44.0	1.0	3.0	140.0	235.0	0.0	2.0	180.0	0.0	0.0	1.0
13.0	61.0	1.0	1.0	134.0	234.0	0.0	0.0	145.0	0.0	2.6	2.0
14.0	57.0	0.0	4.0	128.0	303.0	0.0	2.0	159.0	0.0	0.0	1.0
15.0	71.0	0.0	4.0	112.0	149.0	0.0	0.0	125.0	0.0	1.6	2.0
16.0	46.0	1.0	4.0	140.0	311.0	0.0	0.0	120.0	1.0	1.8	2.0
17.0	53.0	1.0	4.0	140.0	203.0	1.0	2.0	155.0	1.0	3.1	3.0
18.0	64.0	1.0	1.0	110.0	211.0	0.0	2.0	144.0	1.0	1.8	2.0
19.0	40.0	1.0	1.0	140.0	199.0	0.0	0.0	178.0	1.0	1.4	1.0
20.0	67.0	1.0	4.0	120.0	229.0	0.0	2.0	129.0	1.0	2.6	2.0
21.0	48.0	1.0	2.0	130.0	245.0	0.0	2.0	180.0	0.0	0.2	2.0
22.0	43.0	1.0	4.0	115.0	303.0	0.0	0.0	181.0	0.0	1.2	2.0

Step1: Attribute Ranking

After applying the SMOTE, apply the ranking for the attributes for the new dataset and observe its attributes. Table 4.11 Presents the ranking order of the generated data compared with the small data ranking order.

Table 4.11 the ranking order of the generated data compared with the small data ranking order

Rank	SMOT		Small dataset	
	Attribute name	Old No.	Attribute name	Old No.
1	Thallium	14	Thallium	14
2	Number of vessels fluor	13	Number of vessels fluor	13
3	ST depression	11	Exercise angina	10
4	Exercise angina	10	Max HR	9
5	Max HR	9	Depression St	11
6	Chest pain type	4	Chest pain type	4
7	Slope of St	12	Slope of St	12
8	sex	3	Sex	3
9	age	2	Age	2
10	EKG results	8	EKG results	8
11	Bp	5	Bp	5
12	Cholesterol	6	Cholesterol	6
13	FBS over120	7	FBS over120	7
14	Index	1	Index	1

Table 4.11 shows that the attribute ranking of the extended data is very close that for the small dataset. Such results are a good indication that the generated virtual data behavior is very close or similar to the original real small or limited data. One advantage of attribute ranking is to indicate the most important attributes. The least important attributes can be eliminated or avoided in any reduction process. Also, the significant attributes must be considered in any prediction model.

Step2: Attributes Correlation

Re-execute the correlation process for the new Dataset with the same previous mechanism to test its correlation coefficient. The correlation results are recorded in a matrix of (14 x 14). Table 4.12 shows a sample of the created correlation matrix.

Table 4.12 correlation matrix for the extension heart disease dataset

	Index	age	sex	Chest pain type	Bp	Cholesterol	FBS over120	EKG results	Max HR	Exercise angina
Thallium	0.01	0.106	0.391	0.263	0.132	0.0288	0.0492	0.0073	0.253-	0.321
Number of vessels fluor	0.00613-	0.356	0.0868	0.226	0.0857	0.127	0.124	0.114	0.265-	0.153
Slope of St	0.0234	0.16	0.0505	0.137	0.142	0.00576	0.0441	0.161	0.387-	0.256
St Depression	0.0845	0.194	0.0974	0.167	0.223	0.0277	0.0255-	0.12	0.349-	0.275
Exercise angina	0.0646-	0.0983	0.18	0.353	0.0828	0.0782	0.00411-	0.0951	0.381-	
Max HR	0.103	0.402-	0.0761-	0.318-	0.0391	0.0187-	0.0225	0.0746-		0.381-
EKG results	0.0398-	0.128	0.0393	0.0743	0.116	o.168	0.0535		0.0746-	0.0951
FBS over120	0.0147-	0.123	0.0421	0.0985-	0.156	0.025		0.0535	0.0225	0.00411-
Cholesterol	0.0435	0.22	0.202-	0.0905	0.173		0.025	o.168	0.0187-	0.0782
Bp	0.0383	0.273	0.0627-	0.0432-		0.173	0.156	0.116	0.0391	0.0828
Chest pain type	0.0069-	0.0969	0.0346		0.0432-	0.0905	0.0985-	0.0743	0.318-	0.353
sex	0.0339	0.0944-		0.0346	0.0627-	0.202-	0.0421	0.0393	0.0761-	0.18
age	0.0442-		0.0944-	0.0969	0.273	0.22	0.123	0.128	0.402-	0.0983
Index		0.0442-	0.0339	0.0069-	0.0383	0.0435	0.0147-	0.0398-	0.103	0.0646-

Depression St	**0.0845**	**0.194**	**0.0974**	**0.167**	**0.223**	**0.0277**	**0.0255-**	**0.12**	**0.349-**	**0.275**		**0.61**	**0.255**	**0.324**
Slope of St	**0.0234**	**0.16**	**0.0505**	**0.137**	**0.142**	**0.00576**	**0.0441**	**0.161**	**0.387-**	**0.256**	**0.61**		**0.109**	**0.284**
Number of vessels fluor	**0.00613-**	**0.356**	**0.0868**	**0.226**	**0.0857**	**0.127**	**0.124**	**0.114**	**0.265-**	**0.153**	**0.255**	**0.109**		**0.256**
Thallium	**0.01**	**0.106**	**0.391**	**0.263**	**0.132**	**0.0288**	**0.0492**	**0.0073**	**0.253-**	**0.321**	**0.324**	**0.284**	**0.256**	

Step3: Predictive Modeling

Applying the prediction models (Naïve bayes, Decision tree, KNN, and Logistic). Table 4.12 presents the specific measures for each of the used prediction models. These measures are (precision, Recall, F-Measure, and accuracy) detailed by class.

Table 44.12. specific measures for each of the used prediction models for the extension heart disease dataset

Algorithm type	Precision	Recall	F- score	accuracy	Class
Naïve Bayes	0.92 0.89 0.90 0.90	0.87 0.93 0.90 0.90	0.89 0.91 0.90 0.90	0.91	Presence (1) Absence (0) Macro avg Weighted Avg
Decision tree	0.83 0.92 0.87 0.88	0.90 0.85 0.90 0.88	0.86 0.89 0.87 0.88	0.88	Presence (1) Absence (0) Macro avg Weighted Avg
KNN	0.93 0.96 0.94 0.94	0.95 0.94 0.94 0.94	0.94 0.95 0.94 0.94	0.94	Presence (1) Absence (0) Macro avg Weighted Avg

Logistic	0.88 0.93 0.91 0.91	0.89 0.92 0.91 0.91	0.89 0.93 0.91 0.91	0.91	Presence (1) Absence (0) Macro avg Weighted Avg

Table 4.13 presents the specific measures (Accuracy, Error, precision, Recall) for each of the used prediction models.

Table 4.13 presents the specific measures (Accuracy, Error, precision, Recall) for the extension heart disease dataset

Algorithm type	Accuracy	Error	Precision	Recall
Naïve Bayes	0.91	0	0.91	0.87
Decision tree	0.88	0.12	0.81	0.89
KNN	0.94	0.5	0.92	0.94
Logistic	0.91	0	0.88	0.89

For each of prediction models can be evaluated using confusion matrices as shown in Tables 4.14, 4.15, 4.16 and 4.17.

Table 4.14 Confusion Matrixes (Naïve Bayes) for the extension heart disease dataset

	Absence (0)	Presence (1)
Absence (0)	108	13
Presence (1)	8	87

Table 4.15 Confusion Matrixes (Decision tree) for the extension heart disease dataset

	Absence (0)	Presence (1)
Absence (0)	103	10
Presence (1)	19	84

Table 4.16 Confusion Matrixes (KNN) for the extension heart disease dataset

	Absence (0)	Presence (1)
Absence (0)	117	5
Presence (1)	7	87

Table 4.17 Confusion Matrixes (Logistic Regression) for the extension heart disease dataset

	Absence (0)	**Presence (1)**
Absence (0)	121	9
Presence (1)	10	76

After obtaining the results of the algorithms, treeing to delete some of the least important features according to the ranking in a Table 4.11, and then monitor the effect of the deletion on each algorithm accuracy. Where when deleting (from rank 12, 13 14 features) in Naïve Bayes and index feature in Logistic not effected on dataset accuracy. While remaining features when removing it the accuracy became less.

4.2.3 Compare result between small and extend dataset

After calculating the Standard Deviation, Average, Max, and Min for the small dataset, and after balancing and extend using the (SMOTE) method, the results were plotted by a graphical analysis shown in the figure 4.1. Where the orange line indicates the SMOTE, while the blue line indicates the data after the small dataset. Through the analysis, it was noticed that the behavior of the data is completely similar, only it became an increase in values, but the behavior is the same.

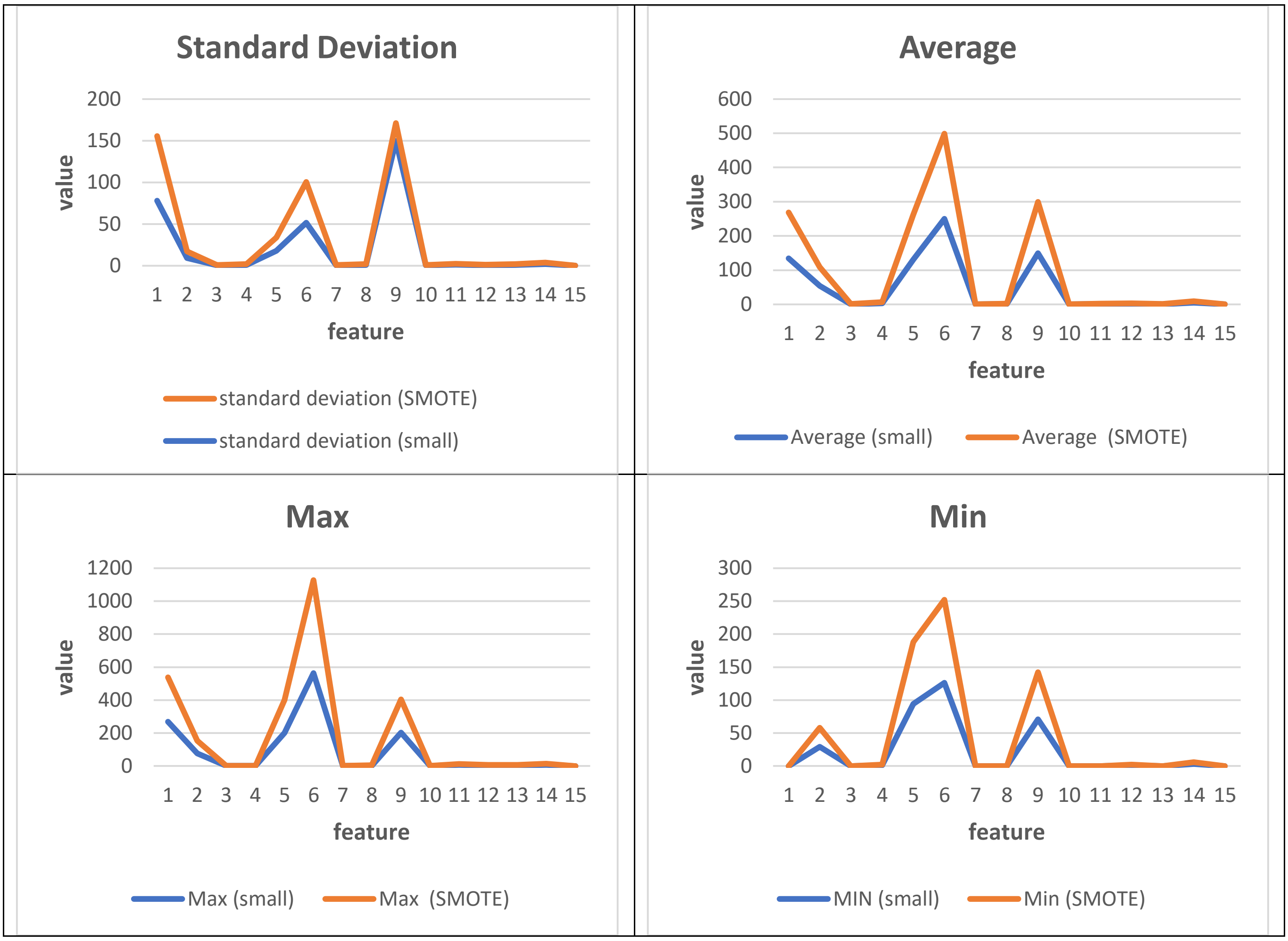


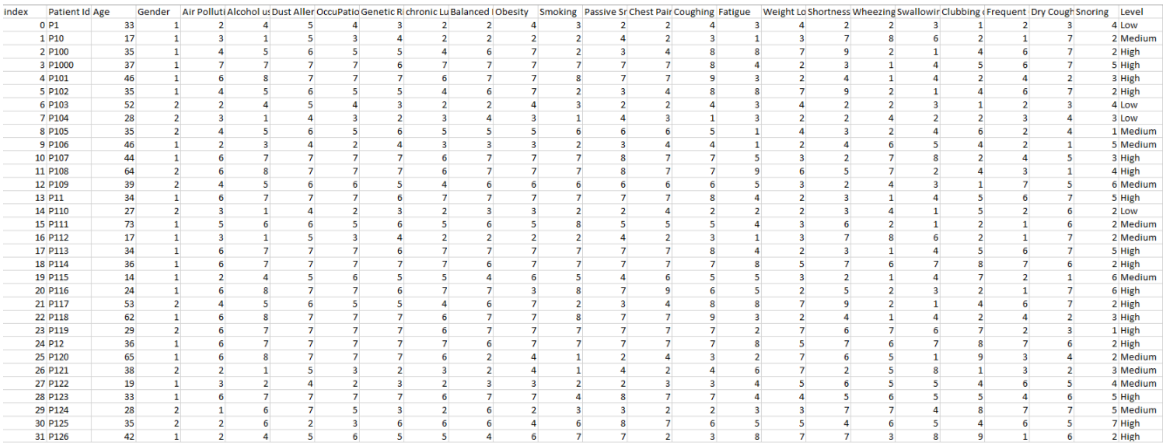
Figure 4.1 Standard Deviation, Average, Max, and Min for the heart disease dataset

According to Tables 4.5 and 4.13, the prediction results showed an improvement in the accuracy, precision, and recall values when extension and balancing the data in the dataset. And the KNN algorithm showed the highest achieved results with a percentage of 94% for accuracy, 92% for precision, and 94% for recall.

4.3 Lung Cancer Prediction Dataset

The lung cancer dataset is often used for research and analysis related to lung cancer. Table 4.18 present sample from the dataset.

Table 4.18 lung cancer dataset sample



index	Patient Id	Age	Gender	Air Polluti	Alcohol u	Dust Aller	OccuPatio	Genetic R	chronic Lu	Balanced I	Obesity	Smoking	Passive Sr	Chest Pair	Coughing	Fatigue	Weight Lo	Shortness	Wheezing	Swallowir	Clubbing	Frequent	Dry Cough	Snoring	Level
0	P1	33	1	2	4	5	4	3	2	2	4	3	2	2	4	3	4	2	2	3	1	2	3	4	Low
1	P10	17	1	3	1	5	3	4	2	2	2	2	4	2	3	1	3	7	8	6	2	1	7	2	Medium
2	P100	35	1	4	5	6	5	5	4	6	7	2	3	4	8	8	7	9	2	1	4	6	7	2	High
3	P1000	37	1	7	7	7	7	6	7	7	7	7	7	7	8	4	2	3	1	4	5	6	7	5	High
4	P101	46	1	6	8	7	7	7	6	7	7	8	7	7	9	3	2	4	1	4	2	4	2	3	High
5	P102	35	1	4	5	6	5	5	4	6	7	2	3	4	8	8	7	9	2	1	4	6	7	2	High
6	P103	52	2	2	4	5	4	3	2	2	4	3	2	2	4	3	4	2	2	3	1	2	3	4	Low
7	P104	28	2	3	1	4	3	2	3	4	3	1	4	3	1	3	2	2	4	2	2	3	4	3	Low
8	P105	35	2	4	5	6	5	6	5	5	5	6	6	6	5	1	4	3	2	4	6	2	4	1	Medium
9	P106	46	1	2	3	4	2	4	3	3	3	2	3	4	4	1	2	4	6	5	4	2	1	5	Medium
10	P107	44	1	6	7	7	7	7	6	7	7	7	8	7	7	5	3	2	7	8	2	4	5	3	High
11	P108	64	2	6	8	7	7	7	6	7	7	7	8	7	7	9	6	5	7	2	4	3	1	4	High
12	P109	39	2	4	5	6	6	5	4	6	6	6	6	6	6	5	3	2	4	3	1	7	5	6	Medium
13	P11	34	1	6	7	7	7	6	7	7	7	7	7	7	8	4	2	3	1	4	5	6	7	5	High
14	P110	27	2	3	1	4	2	3	2	3	3	2	2	4	2	2	2	3	4	1	5	2	6	2	Low
15	P111	73	1	5	6	6	5	6	5	6	5	8	5	5	5	4	3	6	2	1	2	1	6	2	Medium
16	P112	17	1	3	1	5	3	4	2	2	2	2	4	2	3	1	3	7	8	6	2	1	7	2	Medium
17	P113	34	1	6	7	7	7	6	7	7	7	7	7	7	8	4	2	3	1	4	5	6	7	5	High
18	P114	36	1	6	7	7	7	7	7	6	7	7	7	7	7	8	5	7	6	7	8	7	6	2	High
19	P115	14	1	2	4	5	6	5	5	4	6	5	4	6	5	5	3	2	1	4	7	2	1	6	Medium
20	P116	24	1	6	8	7	7	6	7	7	3	8	7	9	6	5	2	5	2	3	2	1	7	6	High
21	P117	53	2	4	5	6	5	5	4	6	7	2	3	4	8	8	7	9	2	1	4	6	7	2	High
22	P118	62	1	6	8	7	7	7	6	7	7	8	7	7	9	3	2	4	1	4	2	4	2	3	High
23	P119	29	2	6	7	7	7	7	6	7	7	7	7	7	7	2	7	6	7	6	7	2	3	1	High
24	P12	36	1	6	7	7	7	7	7	6	7	7	7	7	7	8	5	7	6	7	8	7	6	2	High
25	P120	65	1	6	8	7	7	7	6	2	4	1	2	4	3	2	7	6	5	1	9	3	4	2	Medium
26	P121	38	2	2	1	5	3	2	3	2	4	1	4	2	4	6	7	2	5	8	1	3	2	3	Medium
27	P122	19	1	3	2	4	2	3	2	3	3	2	2	3	3	4	5	6	5	5	4	6	5	4	Medium
28	P123	33	1	6	7	7	7	7	6	7	7	4	8	7	7	4	4	5	6	5	5	4	6	5	High
29	P124	28	2	1	6	7	5	3	2	6	2	3	3	2	2	3	3	7	7	4	8	7	7	5	Medium
30	P125	35	2	2	6	2	3	6	6	6	4	6	8	7	6	5	5	4	6	5	4	6	5	7	High
31	P126	42	1	2	4	5	6	5	5	4	6	7	7	2	3	8	7	7	3	8	9	1	6	2	High

4.3.1 Small Dataset Results

After applying the steps of the proposed model, have the following results:

<u>Step1: Attribute Ranking</u>

After applying the proposed model using python programing language by ranking the small dataset (lung cancer) and observing its attributes. Table 4.19 presents the sorted list of the ranked attributes:

Table 4.19 attributes ranking for the small lung cancer dataset

Rank	Attribute name	Old No.
1	Allergy	9
2	Alcohol consuming	11
3	Swallowing difficulty	14
4	Wheezing	10
5	Coughing	12
6	Chest pain	15
7	Peer pressure	6
8	Yellow fingers	4
9	Fatigue	8
10	Anxiety	5
11	Chronic disease	7

12	Age	2
13	Gender	1
14	Shortness of breath	13
15	Smoking	3

Step2: Attributes Correlation

The correlation results are recorded in a matrix of (15 x 15). The strong correlation coefficient value (which is greater or equal 0.7) is selected and attended. Table 4.20 shows a sample of the created correlation matrix.

Table 4.20 correlation matrix for the small lung cancer dataset

	GENDER	AGE	SMOKING	YELLOW	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL	COUGHING	SHORTNESS OF BREATH	SWALLOWING DIFFICULTY	CHEST PAIN
GENDER		0.0213	0.0363	0.213	0.152	0.276	0.205	0.0836	0.154	0.141	0.454	0.133	0.0649	0.0782	0.363
AGE	0.0213		0.0845-	0.0052	0.0532	0.0187	0.0126-	0.0126	0.028	0.055	0.059	0.17	0.0175-	0.00127-	0.0181-
SMOKING	0.0363	0.0845-		0.0146-	0.16	0.0428-	0.142-	0.296-	0.00191	0.129-	0.0506-	0.129-	0.0613	0.0307	0.12
YELLOW	0.213	0.0052	0.0146-		0.566	0.323	0.0411	0.118-	0.144-	0.0785-	0.289-	0.0126-	0.106-	0.346	0.105-
ANXIETY	0.152	0.0532	0.16	0.566		0,217	0.00968-	0.189-	0.166-	0.192-	0.166-	0.226-	0.144-	0.489	0.114-
PEER_PRESSURE	0.276	0.0187	0.0428-	0.323	0.217		0.0485	0.0781	0.0818-	0.0688-	0.16-	0.089-	0.22-	0.367	0.0948-
CHRONIC DISEASE	0.205	0.0126-	0.142-	0.0411	0.00968	0.0485		0.111-	0.106	0.05-	0.00215	0.175-	0.0265-	0.0752	0.0369-
FATIGUE	0.0836	0.0126	0.296-	0.118-	0.189-	0.0781	0.111-		0.00306	0.142	0.191-	0.147	0.442	0.133-	0.0108-
ALLERGY	0.154	0.028	0.00191	0.144-	0.166-	0.0818-	0.106	0.00306		0.174	0.344	0.19	0.0301-	0.0615-	0.239
WHEEZING	0.141	0.055	0.129-	0.0785-	0.192-	0.0688-	0.05-	0.142	0.174		0.266	0.374	0.0378	0.069	0.148
ALCOHOL CONSUMING	0.454	0.059	0.0506-	0.289-	0.166-	0.16-	0.00215	0.191-	0.344	0.266		0.203	0.179-	0.00929-	0.331
COUGHING	0.133	0.17	0.129-	0.0126-	0.226-	0.089-	0.175-	0.147	0.19	0.374	0.203		0.277	0.158-	0.084
SHORTNESS OF BREATH	0.0649	0.175-	0.0613	0.106-	0.144-	0.22-	0.0265-	0.442	0.0301-	0.0378	0.179-	0.277		0.161-	0.0243
SWALLOWING DIFFICULTY	0.0782	0.00127-	0.0307	0.346	0.489	0.367	0.0752	0.133-	0.0615-	0.069	0.00929-	0.158-	0.161-		0.069
CHEST PAIN	0.363	0.0181-	0.12	0.105-	0.114-	0.0948-	0.0369-	0.0108-	0.239	0.148	0.331	0.084	0.0243	0.069	

Step3: Predictive Modeling

To perform a predictive modeling approach, using prediction models (Naïve Bayes, Decision Tree, KNN, and Logistic).

Table 4.21 presents the specific measures for each of the used prediction models. These measures are (precision, Recall, F-Measure, and accuracy) detailed by class.

Table 4.21 specific measures for each of the used prediction models for the small lung cancer dataset

Algorithm type	Precision	Recall	F- score	accuracy	Class
Naïve Bayes	0.55 0.86 0.70 0.80	0.46 0.90 0.68 0.81	0.50 0.88 0.69 0.80	0.81	NO (0) YES (1) Macro avg Weighted Avg
Decision tree	0.64 0.96 0.80 0.91	0.78 0.92 0.85 0.90	0.70 0.94 0.82 0.91	0.90	NO (0) YES (1) Macro avg Weighted Avg
KNN	0.29 0.95 0.62 0.89	0.40 0.91 0.66 0.87	0.33 0.93 0.63 0.88	0.87	NO (0) YES (1) Macro avg Weighted Avg
Logistic	0.56 0.96 0.76 0.92	0.71 0.93 0.82 0.90	0.63 0.94 0.78 0.91	0.90	NO (0) YES (1) Macro avg Weighted Avg

Table 4.21 presents the specific measures (Accuracy, Error, precision, Recall) for each of the used prediction models.

Table 4.22 presents the specific measures (Accuracy, Error, precision, Recall) for the small lung cancer dataset

Algorithm type	Accuracy	Error	Precision	Recall
Naïve Bayes	0.81	0.1	0.89	0.86
Decision tree	0.90	0	0.92	0.96
KNN	0.87	0.1	0.91	0.94

Logistic	0.90	0.09	0.92	0.96

For each of prediction models can be evaluated using confusion matrices as shown in Tables 4.23, 4.24, 4.25 and 4.26.

Table 4.23 Confusion Matrixes (Naïve Bayes) for the small lung cancer dataset

	No (0)	**Yes (1)**
No (0)	6	5
Yes (1)	7	44

Table 4.24 Confusion Matrixes (Decision tree) for the small lung cancer dataset

	No (0)	**Yes (1)**
No (0)	7	4
Yes (1)	2	49

Table 4.25 Confusion Matrixes (KNN) for the small lung cancer dataset

	No (0)	**Yes (1)**
No (0)	2	5
Yes (1)	3	52

Table 4.26 Confusion Matrixes (Logistic Regression) for the small lung cancer dataset

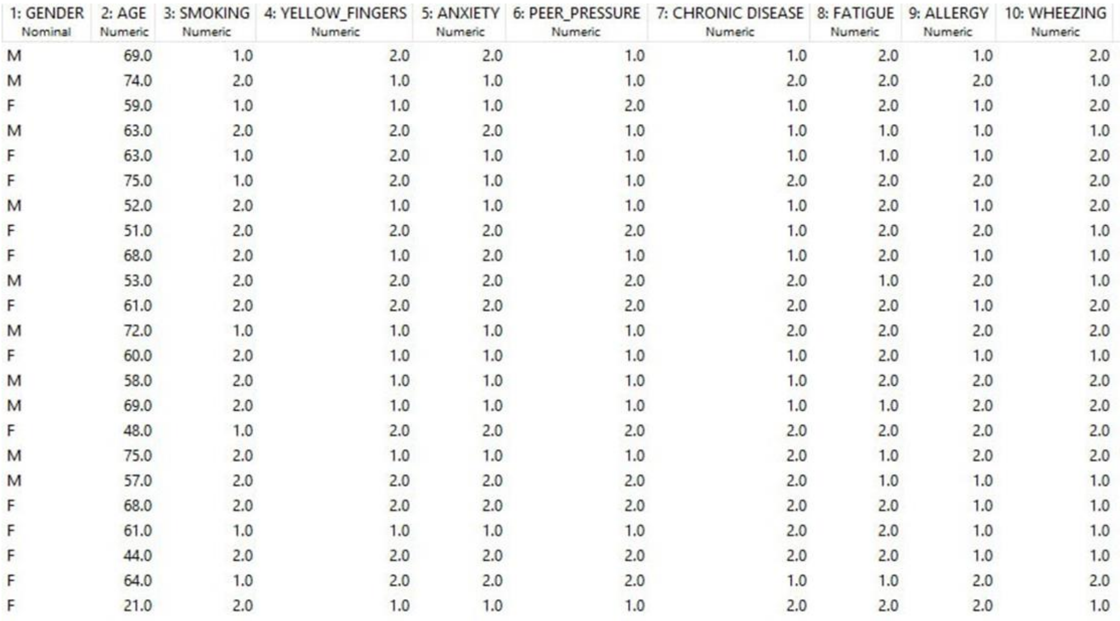
	No (0)	**Yes (1)**
No (0)	5	4
Yes (1)	2	51

After obtaining the results of the algorithms, tried to delete some of the least important features according to the ranking in a Table 4.19, and then monitor the effect of the deletion on each algorithm accuracy. Where when deleting (rank 11 to 15 features) in Naïve Bayes and (rank 14 + 15 features) in KNN not effected on dataset accuracy. While remaining features when removing it the accuracy became less.

4.3.2 Data extension Result

After applying the proposed model to the dataset, the results showed the imbalance of the dataset, so the dataset will be enlarged. A Synthetic Minority Oversampling Technique (SMOT) is used to generate new data to extend the small dataset as the features expanded from 309 to 2328. Table 4.27 Present a sample from the generated data.

Table 4.27 Present a sample from the generated data.



1: GENDER	2: AGE	3: SMOKING	4: YELLOW_FINGERS	5: ANXIETY	6: PEER_PRESSURE	7: CHRONIC DISEASE	8: FATIGUE	9: ALLERGY	10: WHEEZING
Nominal	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric
M	69.0	1.0	2.0	2.0	1.0	1.0	2.0	1.0	2.0
M	74.0	2.0	1.0	1.0	1.0	2.0	2.0	2.0	1.0
F	59.0	1.0	1.0	1.0	2.0	1.0	2.0	1.0	2.0
M	63.0	2.0	2.0	2.0	1.0	1.0	1.0	1.0	1.0
F	63.0	1.0	2.0	1.0	1.0	1.0	1.0	1.0	2.0
F	75.0	1.0	2.0	1.0	1.0	2.0	2.0	2.0	2.0
M	52.0	2.0	1.0	1.0	1.0	1.0	2.0	1.0	2.0
F	51.0	2.0	2.0	2.0	2.0	1.0	2.0	2.0	1.0
F	68.0	2.0	1.0	2.0	1.0	1.0	2.0	1.0	1.0
M	53.0	2.0	2.0	2.0	2.0	2.0	1.0	2.0	1.0
F	61.0	2.0	2.0	2.0	2.0	2.0	2.0	1.0	2.0
M	72.0	1.0	1.0	1.0	1.0	2.0	2.0	2.0	2.0
F	60.0	2.0	1.0	1.0	1.0	1.0	2.0	1.0	1.0
M	58.0	2.0	1.0	1.0	1.0	1.0	2.0	2.0	2.0
M	69.0	2.0	1.0	1.0	1.0	1.0	1.0	2.0	2.0
F	48.0	1.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
M	75.0	2.0	1.0	1.0	1.0	2.0	1.0	2.0	2.0
M	57.0	2.0	2.0	2.0	2.0	2.0	1.0	1.0	1.0
F	68.0	2.0	2.0	2.0	2.0	2.0	2.0	1.0	1.0
F	61.0	1.0	1.0	1.0	1.0	2.0	2.0	1.0	1.0
F	44.0	2.0	2.0	2.0	2.0	2.0	2.0	1.0	1.0
F	64.0	1.0	2.0	2.0	2.0	1.0	1.0	2.0	2.0
F	21.0	2.0	1.0	1.0	1.0	2.0	2.0	2.0	1.0

After applying the steps of the proposed model to the new dataset, have the following results:

<u>Step1: Attribute Ranking</u>

After applying the SMOTE, apply the ranking for the attributes for new dataset and observing its attributes. Table 4.28 Presents the ranking order of the generated data compared with the small data ranking order.

Table 4.28 the ranking order of the generated dataset

Rank	Attribute name	Old No.
1	Allergy	9
2	Coughing	12
3	Alcohol consuming	11
4	Wheezing	10
5	Swallowing difficulty	14
6	Fatigue	8

7	Yellow fingers	4
8	Chest pain	7
9	Chronic disease	15
10	Peer pressure	6
11	Anxiety	5
12	Shortness of breath	13
13	Age	2
14	Smoking	3
15	Gender	1

Table 4.28 shows that the attribute ranking of the extended dataset, the least important attributes can be eliminated or avoided in any reduction process. Also, the significant attributes must be considered in any prediction model.

Step2: Attributes Correlation

Re-execute the correlation process for the new Dataset with the same previous mechanism to test its correlation coefficient. The correlation results are recorded in a matrix of (15 x 15). Table 4.29 shows a sample of the created correlation matrix.

Table 4.29 Correlation matrix for the generated lung cancer dataset

	GENDER	AGE	SMOKING	YELLOW	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL CONSUMING	COUGHING	SHORTNESS OF BREATH	SWALLOWING DIFFICULTY	CHEST PAIN
GENDER		0,119	0,0393	0,237	0,0573	0,274	0,144	0,168	0,226	0,0661	0,466	0,0338	0,321	0,127	0,296
AGE	0,119		0,00913	0,0421-	0,0126-	0,0848	0,0405	0,131	0,0602	0,0728	0,0784	0,149	0,00815	0,0666-	0,0259-
SMOKING	0,0393	0,00913-		0,014-	0,136	0,0657-	0,19-	0,0478-	0,0559	0,15-	0,00602	0,0439-	0,0424	0,0548	0,11
YELLOW	0,237	0,0421-	0,014-		0,556	0,288	0,125	0,127-	0,0339-	0,0117-	0,128-	0,102	0,0418-	0,494	0,0541-
ANXIETY	0,0573	0,0126-	0,136	0,556		0,233	0,104	0,135-	0,12-	0,203-	0,0438-	0,137-	0,208-	0,539	0,12-
PEER_PRESSURE	0,274	0,0848	0,0657-	0,288	0,233		0,158	0,0905	0,0129-	0,0698	0,0412-	0,00509-	0,188-	0,321	0,0382
CHRONIC DISEASE	0,144	0,0405	0,19-	0,125	0,104	0,158		0,0128-	0,158	0,0373	0,102	0,0357-	0,0383-	0,138	0,0584-
FATIGUE	0,168	0,131	0,0478-	0,127-	0,135-	0,0905	0,0128-		0,108	0,247	0,117-	0,225	0,468	0,118-	0,0199
ALLERGY	0,226	0,0602	0,0559	0,0339-	0,12-	0,0129-	0,158	0,108		0,335	0,537	0,357	0,0181-	0,0216	0,348
WHEEZING	0,0661	0,0728	0,15-	0,0117-	0,203-	0,0698	0,0373	0,247	0,335		0,345	0,49	0,129	0,123	0,254
ALCOHOL CONSUMING	0,466	0,0784	0,00602	0,128-	0,0438-	0,0412-	0,102	0,117-	0,537	0,345		0,307	0,185-	0,0869	0,389
COUGHING	0,0338	0,149	0,0439-	0,102	0,137-	0,00509-	0,0357-	0,225	0,357	0,49	0,307		0,297	0,045-	1,135
SHORTNESS OF BREATH	0,321	0,00815-	0,0424	0,0418-	0,208-	0,188-	0,0383-	0,468	0,0181-	0,129	0,185-	0,297		0,123-	0,0217-
SWALLOWING DIFFICULTY	0,127	0,0666-	0,0548	0,494	0,539	0,321	0,138	0,118-	0,0216	0,123	0,0869	0,045-	0,123-		0,0584
CHEST PAIN	0,296	0,0259-	0,11	0,0541-	0,12-	0,0382	0,0584-	0,0199	0,348	0,254	0,389	1,135	0,0217-	0,0584	

Table 4.30 specific measures for each of the used prediction models for the generated lung cancer dataset

Algorithm type	Precision	Recall	F- score	accuracy	Class
Naïve Bayes	0.92	0.95	0.93	0.93	NO (0)
	0.94	0.90	0.92		YES (1)
	0.93	0.93	0.93		Macro avg
	0.93	0.93	0.93		Weighted Avg
Decision tree	1.00	0.96	0.98	0.98	NO (0)
	0.96	1.00	0.98		YES (1)
	0.98	0.98	0.98		Macro avg
	0.98	0.98	0.98		Weighted Avg
KNN	1.00	0.99	1.00	1.00	NO (0)
	0.99	1.00	1.00		YES (1)
	1.00	1.00	1.00		Macro avg
	1.00	0.94	1.00		Weighted Avg
Logistic	0.98	0.97	0.97	0.97	NO (0)
	0.97	0.97	0.97		YES (1)
	0.97	0.97	0.97		Macro avg
	0.97	0.97	0.97		Weighted Avg

Table 4.31 presents the specific measures (Accuracy, Error, precision, Recall) for each of the used prediction models.

Table 4.31 presents the specific measures (Accuracy, Error, precision, Recall) for the generated lung cancer dataset

Algorithm type	Accuracy	Error	Precision	Recall
Naïve Bayes	0.93	0.07	0.90	0.93
Decision tree	0.98	0.02	0.99	0.96
KNN	0.99	0	1.0	0.99
Logistic	0.97	0.02	0.97	0.96

For each of the prediction models can be evaluated using confusion matrices as shown in Tables 4.32, 4.33, 4.34, and 4.35.

Table 4.32 Confusion Matrixes (Naïve Bayes) for the generated lung cancer dataset

	No (0)	Yes (1)
No (0)	232	21
Yes (1)	13	200

Table 4.33 Confusion Matrixes (Decision tree) for the generated lung cancer dataset

	No (0)	Yes (1)
No (0)	234	1
Yes (1)	9	222

Table 4.34 Confusion Matrixes (KNN) for the generated lung cancer dataset

	No (0)	Yes (1)
No (0)	245	0
Yes (1)	2	219

Table 4.35 Confusion Matrixes (Logistic Regression) for the generated lung cancer dataset

	No (0)	Yes (1)
No (0)	249	6
Yes (1)	7	204

After obtaining the results of the algorithms, tried to delete some of the least important features according to the ranking in a Table 4.28, and then monitor the effect of the deletion on each algorithm accuracy. Where when deleting (rank 12 to 15 features) in Naïve Bayes, (rank 14 + 15 features) in KNN, (rank 13 to 15 features) in logistic, and (rank 11 to 15 features) in Decision tree not effected on dataset accuracy. While remaining features when removing it the accuracy became less.

4.3.3 Compare result between small and extend dataset

After calculating the Standard Deviation, Average, Max, and Min for the small dataset, and after balancing and extend using the (SMOTE) method, the results were plotted by a graphical analysis shown in the figure 4.2. Where the orange line indicates the SMOTE, while the blue line indicates the data after the small dataset. Through the analysis, it was noticed that the behavior of the data is completely similar, only it became an increase in values, but the behavior is the same.

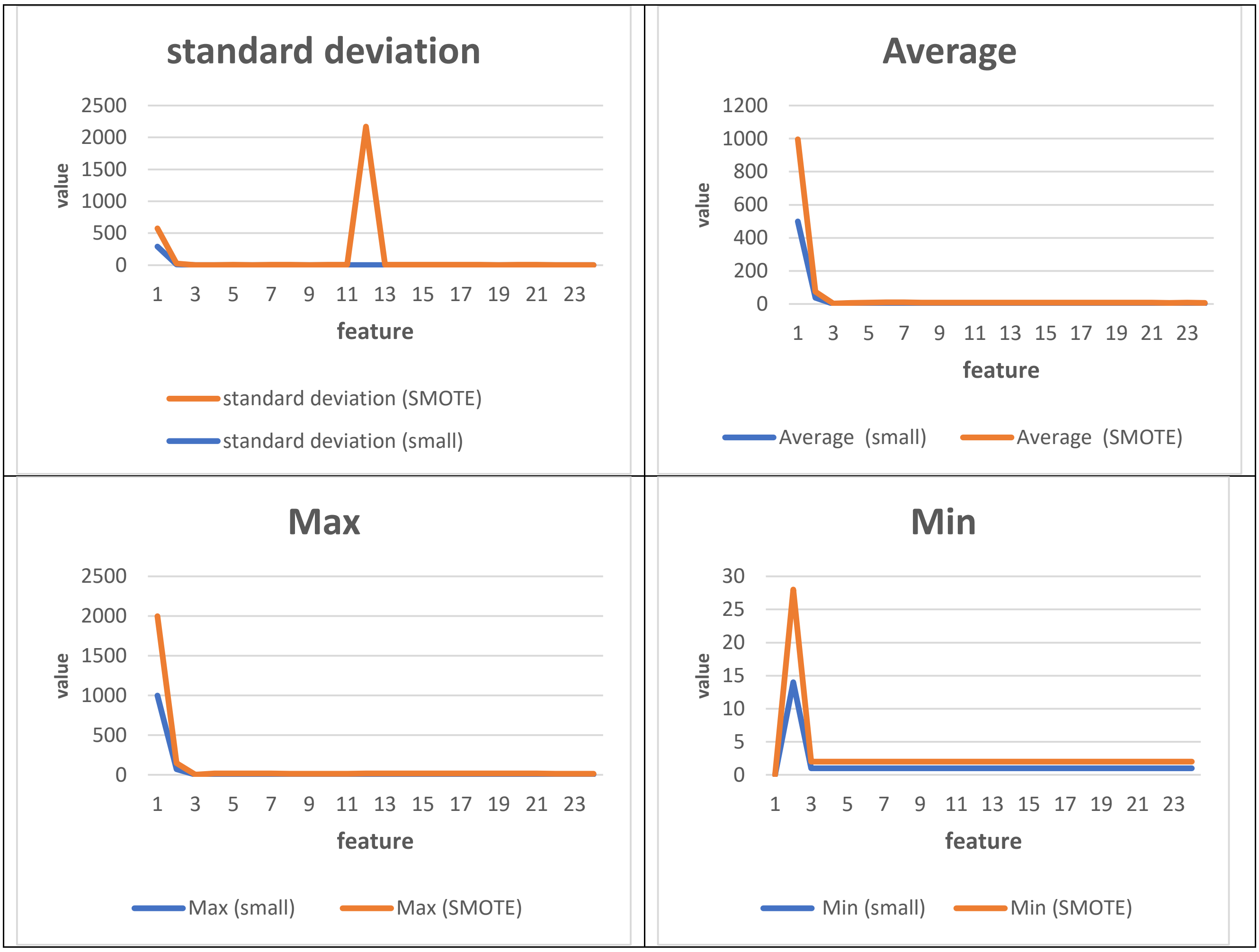


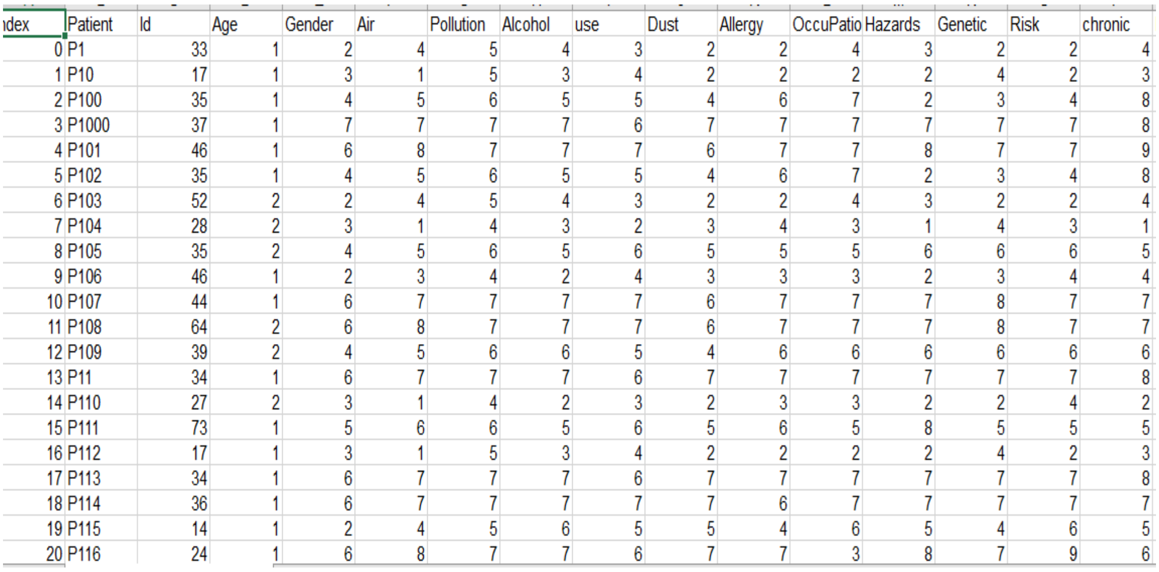
Figure 4.2 Standard Deviation, Average, Max, and Min for the lung cancer dataset

According to Tables 4.22 and 4.31, the prediction results showed an improvement in the accuracy, precision, and recall values when extension and balancing the data in the dataset. And the KNN algorithm showed the highest achieved results with a percentage of 99% for accuracy, 100% for precision, and 99% for recall.

4.4 Cancer Patient Prediction Dataset

Cancer patient Dataset contains information about patients with cancer. Table 4.36 present sample from the Cancer Patient dataset.

Table 4.36 sample from the Cancer Patient dataset



ndex	Patient	Id	Age	Gender	Air	Pollution	Alcohol	use	Dust	Allergy	OccuPatio	Hazards	Genetic	Risk	chronic
0	P1	33	1	2	4	5	4	3	2	2	4	3	2	2	4
1	P10	17	1	3	1	5	3	4	2	2	2	2	4	2	3
2	P100	35	1	4	5	6	5	5	4	6	7	2	3	4	8
3	P1000	37	1	7	7	7	7	6	7	7	7	7	7	7	8
4	P101	46	1	6	8	7	7	7	6	7	7	8	7	7	9
5	P102	35	1	4	5	6	5	5	4	6	7	2	3	4	8
6	P103	52	2	2	4	5	4	3	2	2	4	3	2	2	4
7	P104	28	2	3	1	4	3	2	3	4	3	1	4	3	1
8	P105	35	2	4	5	6	5	6	5	5	5	6	6	6	5
9	P106	46	1	2	3	4	2	4	3	3	3	2	3	4	4
10	P107	44	1	6	7	7	7	7	6	7	7	7	8	7	7
11	P108	64	2	6	8	7	7	7	6	7	7	7	8	7	7
12	P109	39	2	4	5	6	6	5	4	6	6	6	6	6	6
13	P11	34	1	6	7	7	7	6	7	7	7	7	7	7	8
14	P110	27	2	3	1	4	2	3	2	3	3	2	2	4	2
15	P111	73	1	5	6	6	5	6	5	6	5	8	5	5	5
16	P112	17	1	3	1	5	3	4	2	2	2	2	4	2	3
17	P113	34	1	6	7	7	7	6	7	7	7	7	7	7	8
18	P114	36	1	6	7	7	7	7	7	6	7	7	7	7	7
19	P115	14	1	2	4	5	6	5	5	4	6	5	4	6	5
20	P116	24	1	6	8	7	7	6	7	7	3	8	7	9	6

4.4.1 Small Dataset Results

After applying the steps of the proposed model, have the following results:

<u>Step1: Attribute Ranking</u>

After applying the proposed model using python programing language by ranking the small dataset (Cancer Patient) and observing its attributes. It composed of 25 attributes. Table 4.37 presents the sorted list of the ranked attributes:

Table 4.37 attributes ranking for the small Cancer Patient dataset

Rank	Attribute name	Old No.
1	Coughing of Blood	16
2	Obesity	12
3	Passive Smoker	14
4	Balanced Diet	11
5	Air Pollution	5
6	Smoking	13
7	Alcohol use	6

8	Chest Pain	15
9	Dust Allergy	7
10	Genetic Risk	9
11	Chronic Lung Disease	10
12	Occupational Hazards	8
13	Fatigue	17
14	Shortness of Breath	19
15	Frequent Cold	23
16	Weight Loss	18
17	Clubbing of Finger Nails	22
18	Wheezing	20
19	Dry Cough	24
20	Snoring	25
21	Swallowing Difficulty	21
22	Gender	4
23	Age	3
24	Index	2
25	Patient Id	1

Step2: Attributes Correlation

The correlation results are recorded in a matrix of (26 x 26). The strong correlation coefficient value (which is greater or equal 0.7) is selected and attended. Table 4.38 shows a sample of the created correlation matrix.

Table 4.38 correlation matrix for the small Cancer Patient dataset

	Index	Patient Id	Age	Gender	Air pollution	Alcohol use	Dust Allergy	Occupational Hazards	Genetic Risk	Chronic Lung Disease	Balanced Diet	Obesity	Smoking
Snoring	0,00296	0,026	0,0047-	0,182-	0,0213-	0,123	0,0528		0,0568-				
Dry Cough	0,00379	0,0277	0,0121	0,123-	0,261	0,211	0,3	0,229	0,194				
Frequent Cold	0,0457	0,0266	0,0127-	0,000526-	0,175	0,181	0,219	0,16	0,0871				
Clubbing of Finger Nails	0,0157	0,0248	0,0393	0,0342-	0,241	0,415	0,346	0,0772	0,358				
Swallowing Difficult	0,00557	0,0266	0,106-	0,0583-	0,0809-	0,114-	0,0311	0,366	0,0629-				
Wheezing	0,0151	0,0275	0,0954-	0,0763-	0,0554	0,181	0,305	0,00285-	0,205				
Shortness of Breath	0,028	0,0277	0,0353	0,046	0,27	0,436	0,519	0,179	0,458				
Weight Loss	0,0264	0,0285	0,107	0,058-	0,258	0,208	0,322	0,366	0,272				
Fatigue	0,0423	0,025	0,0951	0,116-	0,212	0,237	0,332	0,176	0,231				
Coughing	0,0494	0,0283	0,053	0,147-	0,608	0,668	0,625	0,268	0,632				
Chest Pain	0,0222	0,0283	0,0129	0,218-	0,218-	0,717	0,64	0,646	0,832				
Passive Smoker	0,0195	0,0278	0,00491	0,185-	0,185-	0,593	0,56	0,776	0,609				
Smoking	0,0184	0,0286	0,0753	0,207-	0,482	0,547	0,359	0,498	0,543				
Obesity	0,0506	0,0286	0,0343	0,124-	0,601	0,669	0,701	0,722	0,73				
Balanced Diet	0,0307	0,0297	0,00486	0,0997-	0,525	0,653	0,647	0,692	0,68			0,707	0,645
Chronic Lung Disease	0,0252	0,0284	0,129	0,205-	0,627	0,764	0,62	0,858	0,836		0,623	0,602	0,579
Genetic Risk	0,0307	0,0294	0,0732	0,223-	0,701	0,877	0,788	0,893		0,836	0,68	0,73	0,543
Occupational Hazards	0,0324	0,0282	0,0622	0,192-	0,592	0,879	0,836		0,893	0,858	0,692	0,722	0,498
Dust Allergy	0,38	0,0276	0,0352	0,204-	0,597	0,819		0,836	0,788	0,62	0,647	0,701	0,359
Alcohol use	0,0414	0,0292	0,152	0,228-	0,721		0,819	0,879	0,877	0,764	0,653	0,669	0,547
Air pollution	0,0533	0,0285	0,0995	0,247-		0,721	0,597	0,592	0,701	0,627	0,525	0,601	0,482
Gender	0,0257-	0,031	0,202-		0,247-	0,228-	0,204-	0,192-	0,223-	0,205-	0,0997-	0,124-	0,207-
Age	0,00267	0,0249		0,202-	0,0995	0,152	0,0352	0,0622	0,0732	0,129	0,00486	0,0343	0,0753
Patient Id	0,0274		0,0249	0,031	0,0285	0,0292	0,0276	0,0282	0,0294	0,0284	0,0297	0,0286	0,0286
Index		0,0274	0,0026	0,0257-	0,0533		0,038	0,0324	0,0307	0,0252	0,0307	0,0506	0,0184

Passive Smoker	0,0195	0,0278	0,00491	0,185-	0,185-	0,593	0,56	0,555	0,609	0,573	0,725	0,682	0,762												
Chest Pain	0,0222	0,0283	0,0129	0,218-	0,218-	0,717	0,64	0,776	0,832	0,783	0,798	0,672	0,647	0,696											
Coughing	0,0494	0,0283	0,053	0,147-	0,608	0,668	0,625	0,646	0,632	0,603	0,745	0,798	0,55	0,636	0,712										
Fatigue	0,0423	0,025	0,0951	0,116-	0,212	0,237	0,332	0,268	0,231	0,248	0,410	0,553	0,2	0,378	0,251	9,482									
Weight Loss	0,0264	0,0285	0,107	0,058-	0,258	0,208	0,322	0,176	0,272	0,104	0,006	0,313	0,213-	0,0583	0,001	0,106	0,47								
Shortness of Breath	0,028	0,0277	0,0353	0,046	0,27	0,436	0,519	0,366	0,458	0,182	0,344	0,406	0,0233	0,062	0,237	0,319	0,399	0,574							
Wheezing	0,0151	0,0275	0,0954-	0,0763-	0,0554	0,181	0,305	0,179	0,205	0,057	0,063	0,0943	0,0471	0,2	0,107	0,085-	0,174	0,331	0,20						
Swallowing Difficults	0,00557	0,0266	0,106-	0,0583-	0,0809-	0,114-	0,0311	0,00285-	0,0629-	0,0072	0,046	0,127	0,0471-	0,,349	0,0718	0,086	0,15	0,053	0,2-	0,393					
Clubbing of Finger Nails	0,0157	0,0248	0,0393	0,0342-	0,241	0,415	0,346	0,366	0,358	0,29	0,042	0,149	0,0411-	0,035	0,081	0,066	0,040	0,37		0,33	0,12-				
Frequent Cold	0,0457	0,0266	0,0127-	0,000526-	0,175	0,181	0,219	0,0772	0,0871		0,264	0,288	0,0396	0,105	0,04	0,0664-	0,408	0,16	0,351	0,098	0,132	0,24			
Dry Cough	0,00379	0,0277	0,0121	0,123-	0,261	0,211	0,3	0,16	0,194	0,114	0,332	0,201	0,0101	0,121	0,142	0,148	0,27	0,189	0,49	0,054	0,055	0,307	0,516		
Snoring	0,00296	0,026	0,0047-	0,182-	0,0213-	0,123	0,0528	0,0229	0,0568-	0,043	0,15	0,039	0,189	0,248	0,14	0,0879	0,23	0,189-	0,159-	0,116	0,211	0,017	0,33	0,176	1

Table 4.39 presents the specific measures for each of the used prediction models. These measures are (precision, Recall, F-Measure, accuracy) detailed by class.

Table 4.39 specific measures for each of the used prediction models for the small Cancer Patient dataset

Algorithm type	Precision	Recall	F- score	accuracy	Class
Naïve Bayes	0.951 0.883 0.891 0.910	0.968 0.930 0.915 0.941	0.937 0.929 0.964 0.940	89	LOW MEDIUM HIGH AVG
Decision tree	0.997 0.994 0.996 0.996	0.998 0.997 0.997 0.997	0.998 0.995 0.999 0.997	100	LOW MEDIUM HIGH AVG
KNN	1.000 1.000 1.000 1.000	1.000 1.000 1.000 1.000	1.000 1.000 1.000 1.000	99.8	LOW MEDIUM HIGH AVG
Logistic	0.978 0.962 0.977 0.971	0.986 0.978 0.982 0.982	0.982 0.986 0.974 0.982	99.9	LOW MEDIUM HIGH AVG

For each of prediction models can be evaluated using confusion matrices as shown in Tables 4.40, 4.41, 4.42 and 4.43.

Table 4.40 Confusion Matrixes (Naïve Bayes)

	Low	Medium	High
Low	10	20	273
Medium	60	272	0
High	345	20	0

Table 4.41 Confusion Matrixes (Decision tree)

	Low	Medium	High
Low	0	0	303
Medium	0	332	0
High	365	0	0

Table 4.42 Confusion Matrixes (KNN)

	Low	Medium	High
Low	0	1	303
Medium	0	331	0
High	365	0	0

Table 4.43 Confusion Matrixes (Logistic Regression)

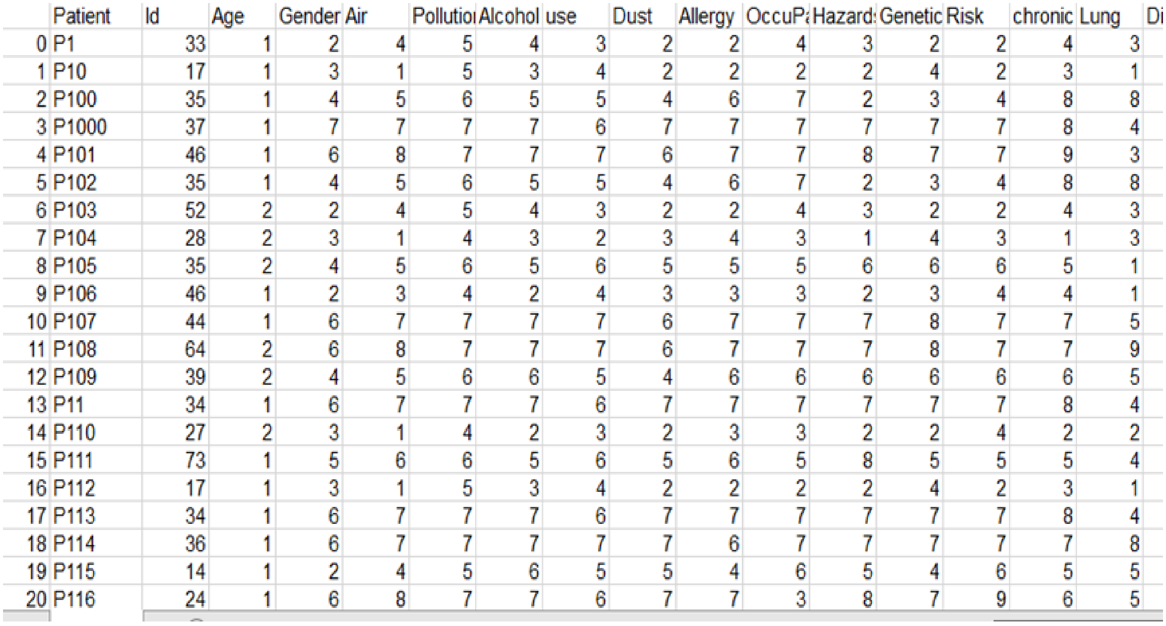
	Low	Medium	High
Low	0	1	302
Medium	0	332	0
High	365	0	0

After obtaining the results of the algorithms, tried to delete some of the least important features according to the ranking in a Table 4.37, and then monitor the effect of the deletion on each algorithm accuracy. Where when deleting (rank 21 to 25 features) in Naïve Bayes, (rank 5 to 25 features) in KNN, (rank 11 to 25 features) in logistic, and (rank 5 to 25 features) in Decision tree not effected on dataset accuracy. While remaining features when removing it the accuracy became less.

4.4.2 Data extension Result

A Synthetic Minority Oversampling Technique (SMOTE) is used to generate new data to extend the small dataset as the features expanded from 1000 to 3270. Table 4.44 Present a sample from the generated data.

Table 4.44 Sample of the extend Cancer Patient dataset



	Patient	Id	Age	Gender	Air	Pollutio	Alcohol	use	Dust	Allergy	OccuPa	Hazard	Genetic	Risk	chronic	Lung	Di
0	P1	33	1	2	4	5	4	3	2	2	4	3	2	2	4	3	
1	P10	17	1	3	1	5	3	4	2	2	2	2	4	2	3	1	
2	P100	35	1	4	5	6	5	5	4	6	7	2	3	4	8	8	
3	P1000	37	1	7	7	7	7	6	7	7	7	7	7	7	8	4	
4	P101	46	1	6	8	7	7	7	6	7	7	8	7	7	9	3	
5	P102	35	1	4	5	6	5	5	4	6	7	2	3	4	8	8	
6	P103	52	2	2	4	5	4	3	2	2	4	3	2	2	4	3	
7	P104	28	2	3	1	4	3	2	3	4	3	1	4	3	1	3	
8	P105	35	2	4	5	6	5	6	5	5	5	6	6	6	5	1	
9	P106	46	1	2	3	4	2	4	3	3	3	2	3	4	4	1	
10	P107	44	1	6	7	7	7	7	6	7	7	7	8	7	7	5	
11	P108	64	2	6	8	7	7	7	6	7	7	7	8	7	7	9	
12	P109	39	2	4	5	6	6	5	4	6	6	6	6	6	6	5	
13	P11	34	1	6	7	7	7	6	7	7	7	7	7	7	8	4	
14	P110	27	2	3	1	4	2	3	2	3	3	2	2	4	2	2	
15	P111	73	1	5	6	6	5	6	5	6	5	8	5	5	5	4	
16	P112	17	1	3	1	5	3	4	2	2	2	2	4	2	3	1	
17	P113	34	1	6	7	7	7	6	7	7	7	7	7	7	8	4	
18	P114	36	1	6	7	7	7	7	7	6	7	7	7	7	7	8	
19	P115	14	1	2	4	5	6	5	5	4	6	5	4	6	5	5	
20	P116	24	1	6	8	7	7	6	7	7	3	8	7	9	6	5	

After applying the steps of the proposed model to the new dataset, have the following results:

<u>Step1: Attribute Ranking</u>

After applying the SMOTE, apply the ranking for the attributes for new dataset and observing its attributes. Table 4.45 Presents the ranking order of the generated data.

Table 4.45 the ranking order of the generated dataset for the extend Cancer Patient dataset

Rank	Attribute name	Old No.
1	Coughing of Blood	16
2	Obesity	12
3	Passive Smoker	14
4	Balanced Diet	11
5	Air Pollution	5
6	Smoking	13
7	Alcohol use	6

8	Chest Pain	15
9	Dust Allergy	7
10	Genetic Risk	9
11	Chronic Lung Disease	10
12	Occupational Hazards	8
13	Fatigue	17
14	Shortness of Breath	19
15	Frequent Cold	23
16	Weight Loss	18
17	Clubbing of Finger Nails	22
18	Wheezing	20
19	Dry Cough	24
20	Snoring	25
21	Swallowing Difficulty	21
22	Gender	4
23	Age	3
24	Index	2
25	Patient Id	1

Table 4.45 shows that the attribute ranking of the extended dataset, the least important attributes can be eliminated or avoided in any reduction process. Also, the significant attributes must be considered in any prediction model.

Step2: Attributes Correlation

Re-execute the correlation process for the new Dataset with the same previous mechanism to test its correlation coefficient. The correlation results are recorded in a matrix of (25 x 25). Table 4.46 shows a sample of the created correlation matrix.

Table 4.46 correlation matrix for the extend Cancer Patient dataset

	Index	Patient Id	Age	Gender	Air pollution	Alcohol use	Dust Allergy	Occupational Hazards	Genetic Risk	Chronic Lung Disease	Balanced Diet	Obesity	Smoking
Index	1	0,0385	0,025	0,0454-	0,0435	0,0439	0,0369	0,0353	0,0298	0,00832	0,0138	0,0484	0,00125-
Patient Id	0,0385	1	0,034	0,0247	0,0295	0,0332	0,0327	0,0311	0,0329	0,029	0,0291	0,0306	0,0297
Age	0,025	0,034	1	0,198-	0,107	0,169	0,0172	0,0613	0,0593	0,172	0,0251-	0,00613	0,0995
Gender	0,0454-	0,0247	0,198-	1	0,228-	0,199-	0,187-	0,151-	0,218-	0,171-	0,121-	0,136-	0,226-
Air pollution	0,0435	0,0295	0,107	0,228-	1	0,721	0,597	0,592	0,701	0,618	0,516	0,617	0,492
Alcohol use	0,0439	0,0332	0,169	0,199-	0,721	1	0,786	0,854	0,859	0,746	0,626	0,681	0,499
Dust Allergy	0,0369	0,0327	0,0172	0,187-	0,597	0,786	1	0,811	0,781	0,605	0,641	0,733	0,299
Occupational Hazards	0,0353	0,0311	0,0613	0,151-	0,592	0,854	0,811	1	0,861	0,815	0,674	0,738	0,423
Genetic Risk	0,0298	0,0329	0,0593	0,218-	0,701	0,859	0,781	0,861	1	0,795	0,659	0,736	0,49
Chronic Lung Disease	0,00832	0,0291	0,172	0,171-	0,618	0,746	0,605	0,815	0,795	1	0,601	0,624	0,526
Balanced Diet	0,0138	0,0306	0,0251-	0,121-	0,516	0,626	0,641	0,674	0,659	0,601	1	0,708	0,599
Obesity	0,0484	0,0285	0,00613	0,136-	0,617	0,681	0,733	0,738	0,736	0,624	0,708	1	0,474
Smoking	0,00125-	0,0297	0,0995	0,226-	0,492	0,499	0,299	0,423	0,49	0,526	0,599		1
Passive Smoker	0,014	0,0297	0,00883	0,192-	0,615	0,557	0,555	0,519	0,577	0,537	0,693		
Chest Pain	O, o207	0,0297	0,0293	0,232-	0,593	0,7	0,655	0,76	0,807	0,742	0,776		
Coughing	0,0461	0,0287	0,0782	0,172-	0,598	0,659	0,611	0,646	0,631	0,624	0,712		
Fatigue	0,0504	0,0283	0,977	0,119-	0,253	0,254	0,394	0,298	25	0,279	0,414		
Weight Loss	0,042	0,0283	0,132	0,104-	0,297	0,273	0,413	0,25	0,319	0,171	0,0623		
Shortness of Breath	0,0422	0,0308	0,0165	0,0995-	0,311	0,522	0,586	0,449	0,519	0,255	0,384		
Wheezing	0,0128	0,0296	0,115-	0,0803-	0,0554	0,211	0,393	0,234	0,226	0,0719	0,0999		
Swallowing Difficult	0,00618-	0,0304	0,0792-	0,0543-	0,101-	0,138-	0,0621	0,0243-	0,0917-	0,035-	0,034		
Clubbing of Finger Nails	0,0342	0,0277	0,0569	0,0128-	0,275	0,493	0,413	0,444	0,41	0,337	0,131		
Frequent Cold	0,0429	0,0275	0,0103	0,0441-	0,183	0,216	0,266	0,124	0,123	0,0587	0,271		
Dry Cough	0,0158	0,0256	0,0142-	0,137-	0,309	0,306	0,39	0,243	0,281	0,147	0,385		
Snoring	0,00656-	0,0284	0,0136	0,02-	0,0357-	0,0956	0,142	0,0408	0,0204-	0,021	0,151		

Passive Smoker	Chest Pain	Coughing	Fatigue	Weight Loss	Shortness of Breath	Wheezing	Swallowing Difficults	Clubbing of Finger Nails	Frequent Cold	Dry Cough	Snoring
0,00125-	0,0207	0,0461	0,0504	0,042	0,0422	0,0128	0,00618-	0,0342	0,0429	0,0158	0,00656-
0,0287	0,0283	0,0287	0,0283	0,0283	0,0308	0,0296	0,0304	0,0277	0,0275	0,0256	0,0284
0,00883	0,0293	0,0782	0,977	0,132	0,0165	0,115-	0,0792-	0,0569	0,0103	0,0142-	0,0136
0,192-	0,232-	0,172-	0,119-	0,104-	0,0995-	0,0803-	0,0543-	0,0128-	0,0441-	0,137-	0,02-
0,615	0,593	0,598	0,253	0,297	0,311	0,0554	0,101-	0,275	0,183	0,309	0,0357-
0,557	0,7	0,659	0,254	0,273	0,522	0,211	0,138-	0,493	0,216	0,306	0,0956
0,555	0,655	0,611	0,394	0,413	0,586	0,393	0,0621	0,413	0,266	0,39	0,142
0,519	0,76	0,646	0,298	0,25	0,449	0,234	0,0243-	0,444	0,124	0,243	0,0408
0,577	0,807	0,631	0,25	0,319	0,519	0,226	0,0917-	0,41	0,123	0,281	0,0204-
0,537	0,742	0,624	0,279	0,171	0,255	0,0719	0,035-	0,337	0,0587	0,147	0,021
0,693	0,776	0,712	0,414	0,0623	0,384	0,0999	0,034	0,131	0,271	0,385	0,151
0,703	0,715	0,796	0,578	0,354	0,443	0,17	0,147	0,218	0,281	0,261	0,128
0,713	0,713	0,563	0,226	0,15-	0,0319	0,0949-	0,124	0,0229-	0,0826	0,0784	0,13
1	0,67	0,662	0,436	0,139	0,148	0,169	0,297	0,002	0,148	0,192	0,218
	1	0,711	0,306	0,11	0,325	0,131	0,036	0,174	0,083	0,238	0,141
		1	0,529	0,18	0,339	0,0002	0,141	0,033	0,245	0,192	0,121
			1	0,473	0,352	0,22	0,299	0,051	0,42	0,268	0,293
				1	0,514	0,356	0,168	0,326	0,213	0,20	0,047
					1	0,33	0,104	0,18	0,396	0,515	0,009
						1	0,385	0,37	0,209	0,175	0,224
							1	0,12-	0,209	0,068	0,311
								1	0,26	0,335	0,084
									1	0,469	0,381
										1	0,212
											1

Step3: Predictive Modeling

Applying the prediction models (Naïve bayes, Decision tree, KNN and Logistic). Table 4.47 presents the specific measures for each of the used prediction models. These measures are (precision, Recall, F-Measure, accuracy) detailed by class.

Table 4.47 specific measures for each of the used prediction models for the extend Cancer Patient dataset

Algorithm type	Precision	Recall	F- score	accuracy	Class
Naïve bayes	0.951 0.883 0.891 0.910	0.968 0.930 0.915 0.941	0.937 0.929 0.964 0.940	94.0061	LOW MEDIUM HIGH AVG
Decision tree (J48)	0.997 0.994 0.996 0.996	0.998 0.997 0.997 0.997	0.998 0.995 0.999 0.997	99.7248	LOW MEDIUM HIGH AVG
KNN	1.000 1.000 1.000 1.000	1.000 1.000 1.000 1.000	1.000 1.000 1.000 1.000	100	LOW MEDIUM HIGH AVG
Logistic	0.978 0.962 0.977 0.971	0.986 0.978 0.982 0.982	0.982 0.986 0.974 0.982	98.1651	LOW MEDIUM HIGH AVG

For each of prediction models can be evaluated using confusion matrices as shown in Tables 4.48, 4.49, 4.50 and 4.51.

Table 4.48 Confusion Matrixes (Naïve Bayes) for the extend Cancer Patient dataset

	Low	Medium	High
Low	11	65	1136
Medium	94	1234	0
High	704	26	0

Table 4.49 Confusion Matrixes (Decision tree) for the extend Cancer Patient dataset

	Low	Medium	High
Low	0	2	1210
Medium	3	1322	3
High	729	1	0

Table 4.50 Confusion Matrixes (KNN) for the extend Cancer Patient dataset

	Low	Medium	High
Low	0	0	1212
Medium	0	1328	0
High	730	0	0

Table 4.51 Confusion Matrixes (Logistic Regression) for the extend Cancer Patient dataset

	Low	Medium	High
Low	0	22	1190
Medium	7	1309	12
High	711	19	0

After obtaining the results of the algorithms, tried to delete some of the least important features according to the ranking in a Table 4.37, and then monitor the effect of the deletion on each algorithm accuracy. Where when deleting (rank 14 to 25 features) in KNN, (rank 15 to 25 features) in logistic, and (rank 13 to 25 features) in Decision tree not effected on dataset accuracy. While remaining features when removing it the accuracy became less.

By comparing the results before and after using the smote, the results showed more efficiency for Dataset before using the smote. This means that the dataset is balanced and does not need to use smote to balance or enlarge it and can be predicted directly.

4.4.3 Compare result between small and extend dataset

After calculating the Standard Deviation, Average, Max, and Min for the small dataset, and after balancing and extend using the (SMOTE) method, the results were plotted by a graphical analysis shown in the figure 4.3. Where the orange line indicates the SMOTE, while the blue line indicates the data after the small dataset. Through the analysis, it was noticed that the behavior of the data is completely similar, only it became an increase in values, but the behavior is the same.

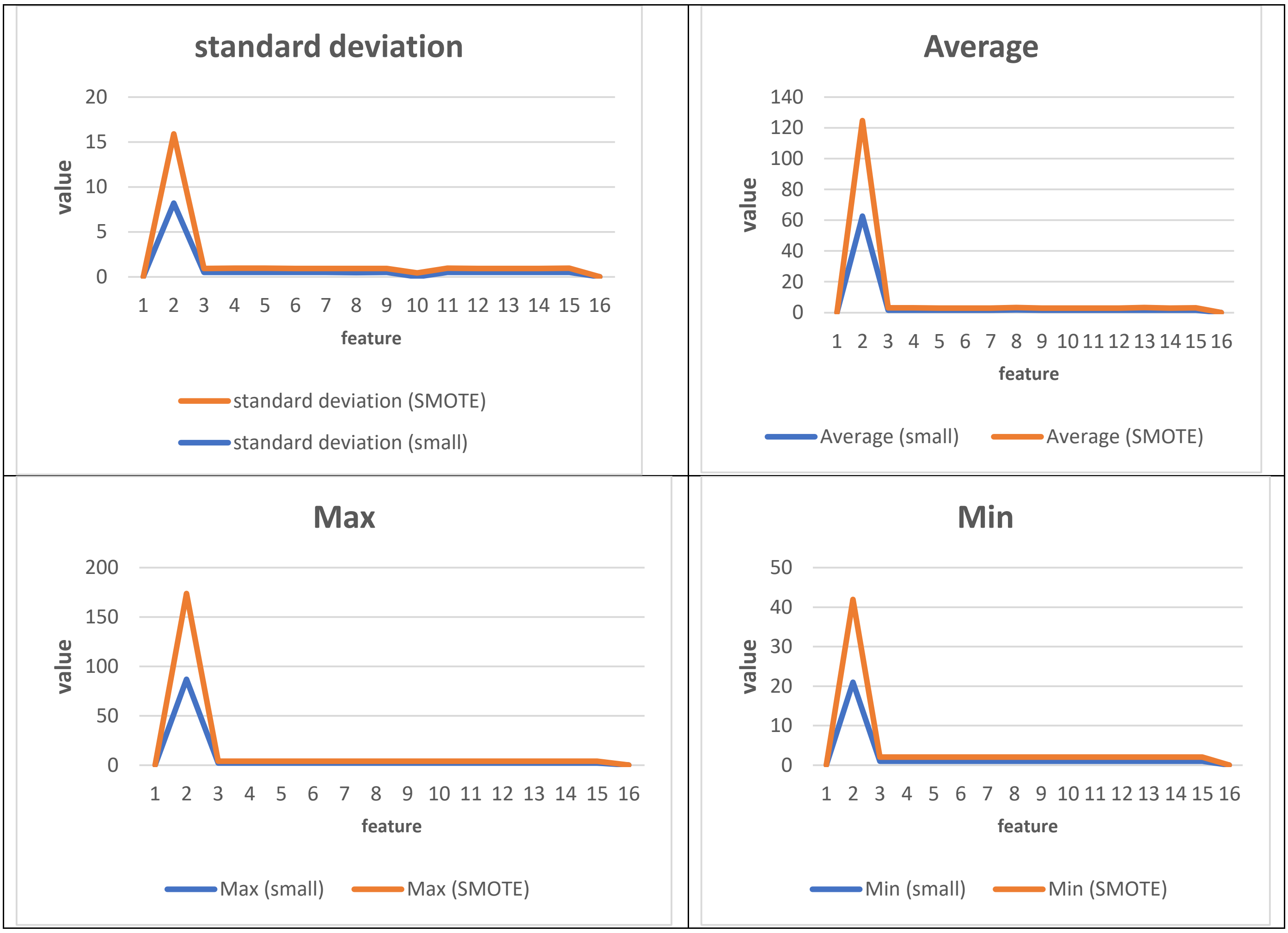


Figure 4.3 Standard Deviation, Average, Max, and Min for the cancer dataset

According to Tables 4.39 and 4.47, the prediction results showed an improvement in the accuracy, precision, and recall values when extension and balancing the data in the dataset. And the KNN algorithm showed the highest achieved results with a percentage of 99% for accuracy, 100% for precision, and 99% for recall.

4.5 Compare the proposed model with related work

Table 4.52 shows a comparison between the results of previous work for the same databases used in this thesis and the results obtained by applying the proposed model, as the results show that the proposed model developed and improved the work of previous models for predicting small datasets.

Table 4.52 Compare the proposed model with related work

Ref.	Dataset	Technique used	Ref. result	Proposed model result
(Boukhatem, et.al, 2022)	Heart Disease dataset	Multilayer Perceptron (MLP), Support Vector Machine (SVM), Random Forest (RF), and Naïve Bayes (NB), to build the prediction models.	SVM model performed best with 91.67% accuracy	KNN had accuracy rates of 94%
(Anil Kumar, et.al, 2022)	lung cancer dataset	compared with the existing SVM and SMOTE methods	98.8% of accuracy rate	KNN had accuracy rates of 99%
(Ranjan, et.al, 2022)	Cancer patient dataset	Random Forest and Deep Learning Techniques	Random Forest Classifier had accuracy rates of 99%	Decision tree had accuracy rates of 100%

CHAPTER FIVE

Conclusions and Future Works

Chapter Five
Conclusions and future works

5.1 Conclusions

The following are some conclusions based on the implementation of the proposed model:

1. SMOTE gave an advantage in balancing and extension the size of the dataset to make the training samples larger, and as a result, the prediction accuracy improved.
2. The Correlation impalement helps determine which features are elimination or kept when the model is executed on the dataset.
3. Ranking helps determine which of the features is least important (eliminate) and most important (remained) in the dataset.
4. The results of using the four algorithms showed different improved results before and after using the SMOTE on the dataset.
5. It was noticed that the four prediction algorithms that were used differ in accuracy according to the size of the dataset. For example, when the small dataset, the KNN works more efficiently, but when the medium and large dataset, the Decision tree works more efficiently.
6. Datasets whose size is more than 1000 are considered medium or large datasets, and most algorithms work efficiently on them and do not need to apply SMOTE.
7. The results of this study can help to overcome the limitations of techniques geared towards prediction of small and important dataset.

5.2 Future Works

Based on the results of this thesis some suggestion future work includes:

1. Implementing the proposed model using a new method for balancing and scaling the data, instead of SMOTE.
2. Use statistical methods to calculate accuracy.
3. Suggest a new method for prediction like time series.
4. Applying the proposed model on other datasets has additional problems with the small data problem.
5. implementing the proposed model to other machine-learning algorithms.

References

Akoglu, H. (2018). User's guide to correlation coefficients. Turkish journal of emergency medicine, 18(3), 91-93.

Al-Mhiqani, M. N., Ahmad, R., Zainal Abidin, Z., Yassin, W., Hassan, A., Abdulkareem, K. H., ... & Yunos, Z. (2020). A review of insider threat detection: classification, machine learning techniques, datasets, open challenges, and recommendations. Applied Sciences, 10(15), 5208.

Al-Najjar, H. A., Pradhan, B., Sarkar, R., Beydoun, G., & Alamri, A. (2021). A new integrated approach for landslide data balancing and spatial prediction based on generative adversarial networks (GAN). Remote Sensing, 13(19), 4011.

Alotaibi, F. S. (2019). Implementation of machine learning model to predict heart failure disease. International Journal of Advanced Computer Science and Applications, 10(6).

Anil Kumar, C., Harish, S., Ravi, P., Svn, M., Kumar, B. P., Mohanavel, V., ... & Asfaw, A. K. (2022). Lung cancer prediction from text datasets using machine learning. BioMed Research International, 2022.

Asghar, N. (2016). Yelp dataset challenge: Review rating prediction. arXiv preprint arXiv:1605.05362.

Bailly, A., Blanc, C., Francis, É., Guillotin, T., Jamal, F., Wakim, B., & Roy, P. (2022). Effects of dataset size and interactions on the prediction performance of logistic regression and deep learning models. Computer Methods and Programs in Biomedicine, 213, 106504.

Banerjee, P., Dehnbostel, F. O., & Preissner, R. (2018). Prediction is a balancing act: Importance of sampling methods to balance sensitivity and specificity of predictive models based on imbalanced chemical data sets. Frontiers in chemistry, 362.

Boukhatem, C., Youssef, H. Y., & Nassif, A. B. (2022, February). Heart disease prediction using machine learning. In 2022 Advances in Science and Engineering Technology International Conferences (ASET) (pp. 1-6). IEEE.

Chen, C. W., Tsai, Y. H., Chang, F. R., & Lin, W. C. (2020). Ensemble feature selection in medical datasets: Combining filter, wrapper, and embedded feature selection results. Expert Systems, 37(5), e12553.

Dey, L., Chakraborty, S., Biswas, A., Bose, B., & Tiwari, S. (2016). Sentiment analysis of review datasets using naive bayes and k-nn classifier. arXiv preprint arXiv:1610.09982.

Fornander, M. J., & Kearney, C. A. (2020). Internalizing symptoms as predictors of school absenteeism severity at multiple levels: Ensemble and classification and regression tree analysis. Frontiers in psychology, 10, 3079.

Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Iii, H. D., & Crawford, K. (2021). Datasheets for datasets. Communications of the ACM, 64(12), 86-92.

Gulati, P., Sharma, A., & Gupta, M. (2016). Theoretical study of decision tree algorithms to identify pivotal factors for performance improvement: A review. Int. J. Comput. Appl, 141(14), 19-25.

Han, S., Williamson, B. D., & Fong, Y. (2021). Improving random forest predictions in small datasets from two-phase sampling designs. BMC medical informatics and decision making, 21(1), 1-9.

Harper, F. M., & Konstan, J. A. (2015). The movielens datasets: History and context. Acm transactions on interactive intelligent systems (tiis), 5(4), 1-19.

He, S., Guo, F., & Zou, Q. (2020). MRMD2. 0: a python tool for machine learning with feature ranking and reduction. Current Bioinformatics, 15(10), 1213-1221.

Ibrahim, H. A. H., Al Zuobi, O. R. A., Al-Namari, M. A., MohamedAli, G., & Abdalla, A. A. A. (2016, February). Internet traffic classification using machine learning approach: Datasets validation issues. In 2016 Conference of Basic Sciences and Engineering Studies (SGCAC) (pp. 158-166). IEEE.

Imran, M., Latif, S., Mehmood, D., & Shah, M. S. (2019). Student academic performance prediction using supervised learning techniques. International Journal of Emerging Technologies in Learning, 14(14).

Ingre, B., Yadav, A., & Soni, A. K. (2018). Decision tree-based intrusion detection system for NSL-KDD dataset. In Information and Communication Technology for Intelligent Systems (ICTIS 2017)-Volume 2 2 (pp. 207-218). Springer International Publishing.

Izonin, I., Tkachenko, R., Zub, K., & Tkachenko, P. (2021). A GRNN-based approach towards prediction from small datasets in medical application. Procedia Computer Science, 184, 242-249.

Jadhav, S. D., & Channe, H. P. (2016). Comparative study of K-NN, naive Bayes and decision tree classification techniques. International Journal of Science and Research (IJSR), 5(1), 1842-1845.

Kirasich, K., Smith, T., & Sadler, B. (2018). Random forest vs logistic regression: binary classification for heterogeneous datasets. SMU Data Science Review, 1(3), 9.

Kowshalya, A. M., Madhumathi, R., & Gopika, N. (2019). Correlation based feature selection algorithms for varying datasets of different dimensionality. Wireless Personal Communications, 108, 1977-1993.

Li, J., Sun, H., & Li, J. (2023). Beyond confusion matrix: learning from multiple annotators with awareness of instance features. Machine Learning, 112(3), 1053-1075.

Liu, Z. T., Wu, B. H., Li, D. Y., Xiao, P., & Mao, J. W. (2020). Speech emotion recognition based on selective interpolation synthetic minority over-sampling technique in small sample environment. Sensors, 20(8), 2297.

Luque, A., Carrasco, A., Martín, A., & de Las Heras, A. (2019). The impact of class imbalance in classification performance metrics based on the binary confusion matrix. Pattern Recognition, 91, 216-231.

Mansourifar, H., & Shi, W. (2020). Deep synthetic minority over-sampling technique. arXiv preprint arXiv:2003.09788.

Mesaros, A., Heittola, T., & Ellis, D. (2018). Datasets and evaluation. Computational Analysis of Sound Scenes and Events, 147-179.

Mohamadou, Y., Halidou, A., & Kapen, P. T. (2020). A review of mathematical modeling, artificial intelligence and datasets used in the study, prediction and management of COVID-19. Applied Intelligence, 50(11), 3913-3925.

Patankar, B., & Chavda, V. (2014). A comparative study of decision tree, naive Bayesian and k-nn classifiers in data mining. International Journal of Advanced Research in Computer Science and Software Engineering, 4(12), 776-779.

Phyu, T. Z., & Oo, N. N. (2016). Performance comparison of feature selection methods. In MATEC web of conferences (Vol. 42, p. 06002). EDP Sciences.

Prasatha, V. S., Alfeilate, H. A. A., Hassanate, A. B., Lasassmehe, O., Tarawnehf, A. S., Alhasanatg, M. B., & Salmane, H. S. E. (2017). Effects of distance measure choice on knn classifier performance-a review. arXiv preprint arXiv:1708.04321, 56.

Ramsey, C. B. (2017). Methods for summarizing radiocarbon datasets. Radiocarbon, 59(6), 1809-1833.

Ranjan, M., Shukla, A., Soni, K., Varma, S., Kuliha, M., & Singh, U. (2022, April). Cancer Prediction Using Random Forest and Deep Learning Techniques. In 2022 IEEE 11th International Conference on Communication Systems and Network Technologies (CSNT) (pp. 227-231). IEEE.

Reddy, C. K., Gopal, V., Cutler, R., Beyrami, E., Cheng, R., Dubey, H., ... & Gehrke, J. (2020). The interspeech 2020 deep noise suppression challenge: Datasets, subjective testing framework, and challenge results. arXiv preprint arXiv:2005.13981.

Riekert, M., Riekert, M., & Klein, A. (2021). Simple baseline machine learning text classifiers for small datasets. SN Computer Science, 2(3), 178.

Ristoski, P., & Paulheim, H. (2016). Semantic Web in data mining and knowledge discovery: A comprehensive survey. Journal of Web Semantics, 36, 1-22.

Sayad, Y. O., Mousannif, H., & Al Moatassime, H. (2019). Predictive modeling of wildfires: A new dataset and machine learning approach. Fire safety journal, 104, 130-146.

Sayyad, S., Shaikh, M., Pandit, A., Sonawane, D., & Anpat, S. (2021). Confusion Matrix-Based Supervised Classification Using Microwave SIR-C SAR Satellite Dataset. In Recent Trends in Image Processing and

Pattern Recognition: Third International Conference, RTIP2R 2020, Aurangabad, India, January 3–4, 2020, Revised Selected Papers, Part II 3 (pp. 176-187). Springer Singapore.

Schwartz, R., & Stanovsky, G. (2022). On the limitations of dataset balancing: The lost battle against spurious correlations. arXiv preprint arXiv:2204.12708.

Sha'Abani, M. N. A. H., Fuad, N., Jamal, N., & Ismail, M. F. (2020). kNN and SVM classification for EEG: a review. In InECCE2019: Proceedings of the 5th International Conference on Electrical, Control & Computer Engineering, Kuantan, Pahang, Malaysia, 29th July 2019 (pp. 555-565). Springer Singapore.

Surya, P. P., & Subbulakshmi, B. (2019, March). Sentimental analysis using Naive Bayes classifier. In 2019 International conference on vision towards emerging trends in communication and networking (ViTECoN) (pp. 1-5). IEEE.

Tarawneh, A. S., Hassanat, A. B., Almohammadi, K., Chetverikov, D., & Bellinger, C. (2020). Smotefuna: Synthetic minority over-sampling technique based on furthest neighbour algorithm. IEEE Access, 8, 59069-59082.

Tate, A. E., McCabe, R. C., Larsson, H., Lundström, S., Lichtenstein, P., & Kuja-Halkola, R. (2020). Predicting mental health problems in adolescence using machine learning techniques. PloS one, 15(4), e0230389.

Uçar, M. K., Nour, M., Sindi, H., & Polat, K. (2020). The effect of training and testing process on machine learning in biomedical datasets. Mathematical Problems in Engineering, 2020.

Venkatesh, B., & Anuradha, J. (2019). A review of feature selection and its methods. Cybernetics and information technologies, 19(1), 3-26.

Zebari, R., Abdulazeez, A., Zeebaree, D., Zebari, D., & Saeed, J. (2020). A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. Journal of Applied Science and Technology Trends, 1(2), 56-70.

Zeng, C., Li, S., Li, Q., Hu, J., & Hu, J. (2020). A survey on machine reading comprehension—tasks, evaluation metrics and benchmark datasets. Applied Sciences, 10(21), 7640.

Zhang, Y., & Ling, C. (2018). A strategy to apply machine learning to small datasets in materials science. Npj Computational Materials, 4(1), 25.

الخلاصة

مجموعة البيانات الصغيرة هي مجموعة بيانات تحتوي على عدد قليل من العينات. يمكن أن تشكل مجموعات البيانات الصغيرة مجموعة متنوعة من التحديات عندما يتعلق الأمر ببناء النماذج التنبؤية. يتطلب النجاح في بناء نماذج تنبؤية على مجموعات بيانات صغيرة مزيجًا من المعرفة بالمجال والأساليب الإحصائية وتقنيات التعلم الآلي.

ستركز هذه الرسالة على إنشاء نموذج تنبؤ لمجموعات البيانات الصغيرة باستخدام طرق التصنيف التقليدية مثل Logistic Regression، Decision Tree، Naive Bayes، و KNN. تتمتع هذه الطرق بدرجة عالية جدًا من دقة التنبؤ بالتصنيف. قد يستخدم الباحثون نماذج مختلفة مع بذل جهد معين لزيادة دقة التنبؤ بالتصنيف مع تزايد شعبية أساليب التعلم الآلي وإمكانية الوصول إليها. للسماح بالتدريب المناسب واختبار النماذج التي تم إنشاؤها باستخدام تقنيات التعلم الآلي، كثيرًا ما تكون البيانات مقيدة بمجموعات أصغر من الملاحظات عما هو مطلوب عادةً. أحد الأساليب لتحسين مجموعة البيانات الصغيرة هو توسيعها باستخدام تقنية الإفراط في أخذ العينات للأقلية الاصطناعية (SMOTE) .

يتم إجراء قياس إحصائي لمجموعات البيانات الأصلية والممتدة للإشارة إلى تشابهها. هناك طريقة أخرى تتمثل في التحقق مما إذا كانت مجموعة البيانات متوازنة أم لا. تعد موازنة مجموعة البيانات غير المتوازنة أمرًا ضروريًا قبل تنفيذ أي أداة للتنبؤ بالتعلم الآلي. في هذه الرسالة يتم استخدام ثلاث مجموعات بيانات. تم استخدام أربع خوارزميات للتعلم الآلي التنبؤي (Logistic Regression، Decision Tree، Naive Bayes، وKNN). يتم إنشاء مصفوفات الارتباك لهذه البيانات قبل وبعد الامتدادات. يتم حساب الدقة والدقة والاستدعاء وF-Score لكل مجموعة بيانات صغيرة أصلية ومجموعات البيانات الموسعة. كمثال؛ تُظهر نتائج مجموعة البيانات الصغيرة الخاصة بالمرض أن دقة Logistic Regression تبلغ 85%، وNaïve Baise 83%، وDecision Tree 69%، وKNN 64%. بعد التمديد، تبلغ نتائج الدقة Logistic Regression 91%، و Naïve Baise 91%، وDecision Tree 88%، و KNN 94%.

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة بابل
كلية تكنلوجيا المعلومات
قسم البرمجيات

طريقه نمذجه لمعالجه مشاكل التنبؤ في قواعد بيانات المجال الصحي الصغيره

رسالة

مقدمة إلى مجلس كلية تكنولوجيا المعلومات في جامعة بابل كجزء من متطلبات نيل درجة الماجستير في تكنولوجيا المعلومات – البرمجيات

مقدمة من قبل

نهى احمد سلمان فرج

باشراف

أ.د.سعد طالب حسون الجبوري

هـ ١٤٤٣

م٢٠٢٣