



A Comparison on Supervised and Semi-Supervised Machine Learning Classifiers for Diabetes Prediction

Lokesh Kola
Vigneshwar Muriki

This thesis is submitted to the Faculty of Engineering at Blekinge Institute of Technology in partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Science. The thesis is equivalent to 10 weeks of full time studies.

The authors declare that they are the sole authors of this thesis and that they have not used any sources other than those listed in the bibliography and identified as references. They further declare that they have not submitted this thesis at any other institution to obtain a degree.

Contact Information:

Author(s):

Lokesh Kola

E-mail: lokl20@student.bth.se

Vigneshwar Muriki

E-mail: vimu20@student.bth.se

University advisor:

Shahrooz Abghari

Department of Computer Science

Faculty of Engineering
Blekinge Institute of Technology
SE-371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

Abstract

Background:

The main cause of diabetes is due to high sugar levels in the blood. There is no permanent cure for diabetes. However, it can be prevented by early diagnosis. In recent years, the hype for Machine Learning is increasing in disease prediction especially during COVID-19 times. In the present scenario, it is difficult for patients to visit doctors. A possible framework is provided using Machine Learning which can detect diabetes at early stages.

Objectives:

This thesis aims to identify the critical features which show an impact on gestational (Type-3) diabetes and experiments are performed to identify the efficient algorithm for Type-3 diabetes prediction. The selected algorithms are Decision Trees, Random Forest, Support Vector Machine, Gaussian Naive Bayes, Bernoulli Naive Bayes, Laplacian Support Vector Machine. The algorithms are compared based on the performance.

Methods:

The method consists of gathering the dataset and preprocessing the data. SelectKBest univariate feature selection was performed for selecting the important features, which influence the Type-3 diabetes prediction. A new dataset was created by binning some of the important features from the original dataset, leading to two datasets, non-binned and binned datasets.. The original dataset was imbalanced due to unequal distribution of class labels. The train-test split was performed on both datasets. Therefore, oversampling technique was performed on both training datasets to overcome imbalance nature. The selected Machine Learning algorithms were trained. Predictions were made on the test data. Hyperparameter tuning was performed on all algorithms to improve the performance. Predictions were made again on the test data and accuracy, precision, recall, and f1-score were measured on both binned and non-binned datasets.

Results:

Among selected Machine Learning algorithms, Laplacian Support Vector Machine attained higher performance with 89.61% and 86.93% on non-binned and binned datasets respectively. Hence, it is the efficient algorithm for Type-3 diabetes prediction. The second best algorithm is Random Forest with 74.5% and 72.72% on non-binned and binned datasets. The non-binned dataset performed well for majority of selected algorithms.

Conclusions:

Laplacian Support Vector Machine scored high performance among the other algorithms on both binned and non-binned datasets. The non-binned dataset showed the best performance in almost all Machine Learning algorithms except Bernoulli Naive Bayes. Therefore, the non-binned dataset is more suitable for the Type-3 diabetes prediction.

Keywords: Machine Learning, Semi-supervised Learning, Supervised Learning, Diabetes Prediction

Acknowledgments

We would like to give acknowledgements to our supervisor Shahrooz Abghari for his constant support and guidance during the thesis. Your feedback helped us to think more and sharpen up our skills. And also we like to thank our friends for helping us throughout the thesis.

We thank our examiner Prashant Goswami for his support and feedback during the project plan and his valuable lectures regarding the research methodology aiding us to a better understanding of carrying out a bachelor thesis in a systematic manner.

Authors:

Lokesh Kola

Vigneshwar Muriki

Contents

Abstract	i
Acknowledgments	ii
1 Introduction	1
1.1 Aim and Objectives	2
1.2 Research Questions	3
2 Background	4
2.1 Machine Learning	4
2.2 Machine Learning Algorithms	5
2.3 Libraries	6
2.4 Feature Selection	7
2.5 Binning	7
2.6 One-Hot Encoding	7
2.7 Oversampling	8
2.8 k -Fold Cross-Validation	8
2.9 GridSearchCV (GSCV)	8
2.10 Evaluation Metrics	8
3 Related Work	10
4 Method	12
4.1 Data Collection and Manipulation	12
4.2 Feature Selection	16
4.3 Binning	18
4.4 One-Hot Encoding	19
4.5 Train-Test Split and Handling Imbalance Data	19
4.6 Training and Testing The Models	20
4.7 Hyper Parameter Tuning	20
5 Results	21
6 Discussion	24
7 Conclusions and Future Work	29
7.1 Conclusion	29
7.2 Future Work	29

A	Best parameters for the selected algorithms	34
A.1	Best parameters for DT	34
A.2	Best parameters for RF	34
A.3	Best parameters for SVM	35
A.4	Best parameters for BNB	35
A.5	Best parameters for GNB	35
A.6	Best parameters for LapSVM	36

List of Figures

4.1	Dataframe	13
4.2	Box plot for Pregnancies	14
4.3	Box plot for Glucose	14
4.4	Box plot for Diastolic Blood Pressure	14
4.5	Box plot for Skin Thickness	15
4.6	Box plot for Insulin	15
4.7	Box plot for BMI	15
4.8	Box plot for Diabetes Pedigree Function	16
4.9	Box plot for Age	16
4.10	The data after removing non-essential features	17
4.11	Binning Age	18
4.12	Binning BMI	18
6.1	Accuracies for non-binned and binned datasets	25
6.2	Precision for non-binned and binned datasets	26
6.3	Recall for non-binned and binned datasets	27
6.4	F1-score for binned and non-binned datasets	27

List of Tables

4.1	Top five Features	17
5.1	Accuracies on test datasets	22
5.2	Precision scores on test datasets	22
5.3	Recall scores on test datasets	22
5.4	F1-scores on test datasets	22

List of Acronyms

ML	Machine Learning
AI	Artificial Intelligence
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
DT	Decision Tree
WHO	World Health Organization
RF	Random Forest
SVM	Support Vector Machine
BNB	Bernoulli Naive Bayes
GNB	Gaussian Naive Bayes
LapSVM	Laplacian Support Vector Machine
GSCV	Grid Search with Cross Validation

Diabetes is one of the deadliest chronic diseases in the world. As per the results from World Health Organization (WHO), around 422 million people in the world are suffering from diabetes [4], particularly in low and middle-income countries. Every year 1.6 million people die from diabetes. This is attributed to late diagnosis. Research [27] has shown that the occurrence of diabetes is more likely in adults aged 18 and above has risen from 4.7% to 8.5% from 1980 to 2014. By 2030, diabetes [35] will be declared as the 7th major cause of death. Statistical results in 2017 showed that 451 million people are suffering from diabetes. If diabetes goes untreated, the count increases to 693 million by 2045 [26].

There are mainly three types of diabetes. Type-1 diabetes is due to a lack of insulin. Type-2 diabetes is due to the ineffective use of insulin. Type-3 also called gestational diabetes is only seen in pregnant women without any previous history of diabetes. If diabetes goes untreated it leads to blindness, heart stroke, nerve damage, chronic kidney diseases, etc. There is no permanent cure for diabetes. However, it can be prevented by early diagnosis.

Type-1 diabetes [4] is due to the pancreas not producing insulin. Of all the diabetes cases 10% of them suffer from Type-1 diabetes. The symptoms include increased thirst and urination, weight loss, appetite, blurred vision. It cannot be cured completely but can be prevented by giving regular insulin injections. This type of diabetes can be found genetically. When the body does not produce enough insulin, it uses the body fats as a substitute which in turn releases chemicals in the blood. Neglecting insulin injections, the harmful chemicals combine and lead to long-term complications.

Type-2 diabetes [4] is due to ineffective use of insulin or incapable of producing insulin. It is usually seen in adults aged 45 years and above often seen in children currently. It is almost certain in individuals with previous family history. The risk of Type-2 diabetes is directly proportional to age. The primary symptoms include heart attack, blurred vision, ulcers, etc. It is not curable however can be forestalled by early diagnosis and having a healthy diet.

Type-3 also called gestational [31] diabetes is only seen in pregnant women without any previous history of diabetes. It is generally seen in women between the 24-28th week of pregnancy. All women should be tested for gestational diabetes at the early stages of pregnancy. To diagnose gestational diabetes women should fast overnight and get tested for their glucose levels. Then they will be given a sugar drink and a blood test will be taken after two hours of consumption of drink and results will be declared based on the blood glucose levels above or below the normal

range at the time of fasting.

Significant research [22] [41] [40] has been performed in diabetes predictions. As time passes, challenges keep increasing to build a system to detect diabetes systematically. The hype for Machine Learning is increasing day to day to analyze medical data to diagnose a disease. Diagnosing diabetes is a herculean task considering the complications. To make predictions accurate, effective analysis needs to be performed on medical data. This is possible by Data Mining and Machine Learning. Depending entirely on technology is not the right way to diagnose a disease. Medical expertise should be involved in the diagnosis of diabetes as there are a lot of factors to be taken into consideration.

Most of the papers [33] [44] [39] used algorithms like Decision Tree, Random Forest, Support Vector Machine, Bernoulli Naive Bayes, Gaussian Naive Bayes, and Laplacian Support Vector Machine. Laplacian Support Vector Machine algorithm for diabetes prediction has produced good accurate results. Therefore those algorithms were selected. Laplacian Support Vector Machine has been performed as semi-supervised learning and the rest are supervised.

The dataset used in this thesis was PIMA Indians Diabetes Dataset [8]. The dataset consists of women aged 21 years and above and is suffering from Type-3 diabetes. So, in this thesis we focused on predicting Type-3 diabetes. The dataset has been preprocessed and Feature selection was performed on the dataset to select the important features. A new dataset has been created from the existing one by binning some of the important features. The two datasets (non-binned and binned) were used separately for training and testing the Machine Learning algorithms. Additionally, hyperparameter tuning was performed to improve the performance of each model. The comparison between algorithms was performed based on accuracy, precision, recall, and f1-score.

1.1 Aim and Objectives

The thesis aims to perform a comparison on the selected supervised and semi-supervised Machine Learning classifiers to predict gestational diabetes (Type-3) on two datasets, of which one dataset is derived from the original dataset. The classifiers are Decision Trees, Random Forest, Support Vector Machine, Gaussian Naive Bayes, Bernoulli Naive Bayes, and Laplacian Support Vector Machine. The algorithms will be compared based on their performance on the two built datasets.

The objectives of the thesis are as follows:

1. Performing data pre-processing on the dataset.
2. Performing feature selection on the dataset.
3. Creating a new dataset from the existing one by binning Age and BMI attributes. Therefore, there will be two datasets.
4. Training and observing the predictions of the models on two datasets and perform hyperparameter tuning to improve the performance.

5. Comparing the performances of classifiers on the test datasets and identifying the efficient algorithm for each dataset by using evaluation metrics.

1.2 Research Questions

The below mentioned research questions have been defined to accomplish the aim.

RQ1: What are the important features which influence in prediction of gestational (Type-3) diabetes?

Motivation: The motivation behind this research question to identify the important features which influence prediction and also helps for experimenting with the RQ2 to find a suitable algorithm. The feature selection also helps to find and remove redundancy and irrelevant features from the dataset to lower computational power and higher the resulting quality.

RQ2: Which is the efficient Machine Learning algorithm among Support Vector Machine, Gaussian Naive Bayes, Bernoulli Naive Bayes, Decision Trees, Random Forest, and Laplacian Support Vector Machine for each dataset to predict gestational (Type-3) diabetes?

Motivation: The motivation behind this research question is to find the efficient Machine Learning classifier among the mentioned classifiers for each dataset. The features obtained from RQ1 and the two datasets help the prediction of Type-3 diabetes. The suitable Machine Learning algorithm will be identified by using evaluation metrics namely accuracy, precision, recall, and f1-score.

2.1 Machine Learning

Machine Learning (ML) [38] is the science of programming computers that learn from the data. ML is a subset of Artificial Intelligence (AI), which is defined as the replication of human intelligence into machines that are customized to think like humans and copy their activities. The term may likewise be applied to any machine that shows attributes related to a human brain, for example, learning, problem-solving, critical thinking etc. Nowadays ML is being used in many industries like healthcare, government, marketing and sales, E-commerce websites. Based on the availability of a labeled data, ML methods can be divided into three learning categories: supervised, unsupervised, and semi-supervised.

1. Supervised Learning [16] methods are used in the presence of a labeled data where the model learns to predict the target variable based on input data. The model learns from labeled training data and aids to predict outcomes for unseen data. Supervised learning is further classified into two categories.
 - Classification is defined as a predictive modeling technique, where the target variable is categorical.
 - Regression is defined as a predictive modeling technique, where the target variable is numerical.
2. Unsupervised Learning [28] involves having only input data and no corresponding output data. The data is unlabeled. Unsupervised ML helps to find all kinds of unknown patterns in the data. Models discover patterns from the data by themselves which is unnoticed previously. The task complexity is more and unpredictable compared to supervised learning problems. Some of the commonly used unsupervised methods are anomaly detections, clustering, neural networks, etc.
3. Semi-Supervised Learning [18] uses the concepts of both supervised and unsupervised learning. Semi-Supervised learning combines a small amount of labeled data with a large amount of unlabeled data during the training of the model. A semi-supervised model will learn from a small amount of labeled data and tries to discover patterns in the unlabeled data and make predictions.

2.2 Machine Learning Algorithms

In this section, the ML algorithms, which were used in this thesis are briefly explained below.

Decision Tree (DT) ¹

DT [1] is a supervised learning technique used in both classification and regression problems. But DTs are mostly preferred in solving classification problems. The DT is a tree-structured classifier whereas the root node represents the entire population or samples, internal nodes represent the features of the dataset, branches represent the rules and each leaf node represents the outcome.

Random Forest (RF) ²

RF [10] is a popular ML algorithm that belongs to supervised learning category. It can be used in both classification and regression problems. It is based on the concept of ensemble learning, which is defined as multiple classifiers that will be combined for efficient predictions. Multiple classifiers are combined to solve a complex issue and in turn, increase the performance of the model. It consists of many DTs in various subsets of the provided dataset and takes the average of those to improve the accuracy of the dataset.

Naive Bayes (NB)

NB [3] is a supervised learning algorithm and is primarily used for solving classification problems. The reason behind it calling naive is because the occurrence of a certain feature is independent of the occurrence of other features. Predictions are done based on the probability of an event. The parameters in the dataset will be used and the occurrences of the symptoms then predictions will be made on the probability of the events.

1. **Gaussian Naive Bayes** ³ - This is an extended version of Naive Bayes. It is easy to work with because it only requires mean and standard deviation from the training data.
2. **Bernoulli Naive Bayes** ⁴ - Bernoulli Naive Bayes works on binomial distribution and is used for discrete data.

Support Vector Machine (SVM) ⁵

SVM [2] is also a supervised ML algorithm that is used for both classification and regression problems. This classifier will plot each data item as a point in an n -

¹<https://scikit-learn.org/stable/modules/tree.html#classification>

²[https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html)

[RandomForestClassifier.html](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html)

³https://scikit-learn.org/stable/modules/naive_bayes.html#gaussian-naive-bayes

⁴https://scikit-learn.org/stable/modules/naive_bayes.html

⁵<https://scikit-learn.org/stable/modules/svm.html#classification>

dimensional space (where n is the number of features or attributes) with the value of each feature being the value of a particular coordinate. Then it performs classification by finding the best suitable hyperplane that separates the two classes.

Laplacian Support Vector Machine (LapSVM)

LapSVM [20] is an extended version of SVM. It is a semi-supervised learning algorithm where a small number of labeled data is given to a model for training. Then, the partially trained model is trained on a large amount of unlabeled data. The unlabeled data learn from the patterns of the partially trained model.

2.3 Libraries

The below libraries are used in this thesis for dealing with the PIDD data.

NumPy⁶

Numerical Python, also called NumPy [36] is also an open-source python package focusing on arrays. It is written in C language. It primarily focuses on numerical data. This library is mainly used for providing objects for multi-dimensional arrays. NumPy library occupies less memory when compared to pandas. The current version is numpy 1.20.3.

Pandas⁷

Pandas [37] is an open-source python package widely used for data analysis and ML tasks. It is built on top of another package called NumPy which supports multi-dimensional arrays. Pandas are used to read csv, json, excel files, etc. Additionally, it is capable of data cleaning, remove outliers, duplicates from the data. In advanced ways, it is used for correlation between the variables and plotting graphs. Pandas can handle a large amount of data. The current version is pandas 1.2.4.

Matplotlib⁸

Matplotlib [7] [43] is an open-source plotting tool used for data visualization. It was originally written by John.B. Hunter [30]. Most of it is written in python. Few parts are written in C, Objective C. The current version is matplotlib 3.4.2.

Scikit-learn⁹

Scikit-learn [9] is an open-source and effective tool built on top of NumPy, matplotlib, and scipy. It is a simple tool for performing predictive data analysis. Scikit-learn

⁶<https://numpy.org/>

⁷<https://pandas.pydata.org/>

⁸<https://matplotlib.org/>

⁹<https://scikit-learn.org/>

provides all kinds of ML algorithms and evaluating a model. It is not a suitable tool for reading and analyzing data. The current version is scikit-learn 0.24.2.

Seaborn¹⁰

Seaborn [6] is a data visualization library built on top of matplotlib. It is capable of handling pandas dataframes better compared to matplotlib. It provides better graphics than matplotlib. The current version is seaborn 0.11.1.

2.4 Feature Selection

Feature selection [24] [32] is the process of selecting the features automatically or manually which influence the target variable. Having irrelevant features leads to a negative impact on the performance of the model. Feature selection should be performed before designing a model. The advantages of performing feature selection are that it reduces overfitting, improves accuracy, and reduces training time. The Univariate feature selection was used in this thesis.

Univariate Feature Selection

This method uses statistical tests and selects the features which are strongly related to the output or target variable. The SelectKBest method was used in this thesis. This method will generate a score for each attribute based on the chi-square test.

The chi-square test is used to find the independence between the target variable and the dependent variable. A higher chi-square value means that the features have the strongest relationship with the output variable.

2.5 Binning

Binning refers [45] to grouping data into bins (or buckets). It converts numerical values into categorical values. It includes binning by distance and binning by frequency. The range of values belonging to the interval will be converted into a single value or a label name is associated with each category. The advantage of using is that it improves the accuracy of the model.

2.6 One-Hot Encoding

One-Hot Encoding [23] is applied on columns with multiple categories and can be categorized in the form of binary values (1 means True and 0 means False). It makes the model easier to understand without any hierarchy in them. Each category is converted into a new column and is assigned a binary value (0 or 1). The only disadvantage is that the number of columns increase and shows an impact on the performance of the model.

¹⁰<https://seaborn.pydata.org/>

2.7 Oversampling

Oversampling [25] is one of the approaches in Random Resampling to handle the imbalance in the target variable. The PIDD dataset used in this thesis is imbalanced. Non-diabetic patients comprise 65% which is the majority class and the remaining 35% people are diabetic which is the minority class showing the imbalanced nature. Oversampling functions by randomly duplicating samples in the minority class and creating a new version of the dataset. Hence, the number of samples will increase than before and so the target variable will be equally distributed. The model will have more data to deal with and can learn more and make better predictions.

2.8 k -Fold Cross-Validation

Cross-validation is used to find the stability of an ML model on unseen data. It is a simple and widely used technique where it has a single parameter k which defines the number of splits to be performed on a dataset. Not choosing the right value of k leads to poor performance of the model or the model can be highly biased on a particular class label. The maximum value of k is 10. The only disadvantage is the class ratio is not preserved in each split as in the original dataset.

Stratified k -Fold Cross-Validation

This is an improvised version of k -fold cross-validation especially used in classification problems. The splits are not completely random, each fold preserves the closest class ratio of the target variable as in the original dataset. This technique is used when the dataset contains few training samples. It is imported using the scikit-learn library.

2.9 GridSearchCV (GSCV)

GSCV is a hyperparameter tuning technique. It helps in iterating through predefined set of hyperparameters and fit the model or estimator to the training dataset. In the end, the best parameters can be selected from the hyperparameters listed. Additionally, it provides cross-validation functionality in which it can be specified the number of times the cross validation to be performed for each set of hyperparameters. The cross-validation can either be k -fold cross-validation or stratified k -fold cross-validation.

2.10 Evaluation Metrics

Confusion Matrix

Confusion matrix [12] is a performance metric widely used in classification problems with two or more output class labels. It is a matrix of four different combinations of predicted and actual values. Accuracy, precision, f1-score, and recall are calculated

using a confusion matrix. To understand the confusion matrix, it is essential to understand the following definitions.

True positive (TP): The number of predicted values are positive and expected values are true.

True Negative (TN): The number of predicted values are negative and so the actual values are also true.

False Positive (FP): Also called Type 1 Error. The number of predicted values are positive but the actual values are false.

False Negative (FN): Also called Type 2 Error. The predicted values are negative but the actual values are false.

Recall

Recall [15] is defined as how well the predictions are out of all the positive classes. Recall should be as high as possible.

$$Recall = \frac{TP}{TP + FN} \quad (2.1)$$

Precision

Precision [14] is defined as the number of classes that are positive from correctly predicted positive classes.

$$Precision = \frac{TP}{TP + FP} \quad (2.2)$$

Accuracy

Accuracy [11] is computed based on the predicted values and true values. Higher the accuracy, better the performance of the model.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (2.3)$$

F1-score

F1-score [13] is used to find precision and recall at the same time. It is difficult to compare models with low precision and high recall and vice-versa. To overcome such problems f1-score comes to the rescue.

$$F\text{-measure} = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (2.4)$$

Priyanka and JayaMalini [42] performed research on diabetes prediction using various ML classifiers. The dataset used was PIDD which comprises 768 instances and nine features. Data preprocessing was performed to convert the raw data into structures data. The proposed algorithms in the research paper were DT, SVM, GNB, Artificial Neural Networks (ANN). The evaluation metrics used for model selection are accuracy, precision, recall, and f1-score. The results show that ANN was the preferred approach for diabetes prediction as it gave better prediction compared to other ML techniques.

Aeshah and Mezher [21] performed diabetes prediction using ML algorithms. The data was collected from diabetic patients at the Security Force Primary Health Care in Tabuk, Saudi Arabia. Data preprocessing was performed to detect outliers, duplicates, empty or missing values. The algorithms used in this research are SVM and RF. The performance of the proposed algorithms was analyzed using ROC Curve and confusion matrix. A confusion matrix was used to find accuracy, precision, and recall for each model. The Area Under Receiver Operating Characteristic Curve (AUROC) for SVM and RF was found to be 0.93% and 0.99% which shows that the RF model predicts diabetes well compared to SVM.

Hasan and Alam [29] performed diabetes prediction using various ML classifiers namely K-Nearest Neighbours (KNN), RF, DT, NB, SVM, AdaBoost (AB), XGBoost (XGB), and Multilayer Perceptron (MLP). AB and XGB was the best ensembling model in terms of performance. PIDD dataset used in the paper. Training and Testing of models were performed for different ML classifiers and suitable evaluation metrics namely accuracy, precision, recall, False Omission Rate (FOR), and Diagnostic Odds Ratio (DOR) were applied for the model selection. ROC and Area Under ROC (AUROC) is also used to measure the predictions rather than the absolute values. Stratified 5-fold cross-validation was used for training the models. (AB+XB) ensembling classifier has maximum AUROC value and hence it is the most efficient algorithm for diabetes prediction.

Nai-Arun and Moungrmai [39] performed diabetes prediction using the same PIDD dataset. The algorithms used were NB, SVM, and DT. Experiments were performed using 10-fold cross-validation. Performance metrics used were accuracy, precision, recall, and Receiver Operating Curve (ROC). The results show that NB is the most efficient compared to other ML algorithms.

Khanam and Foo [33] predicted diabetes using ML algorithms namely DT, KNN, RF, NB, AB, Logistic Regression (LR), SVM. Data preprocessing was performed

using the WEKA ¹ tool. The results shown that LR attained an accuracy of 78.85% when compared to other ML algorithms. Deep Learning techniques were also used with 1,2 and 3 hidden layers and Neural Network with 2 hidden layers attained an accuracy of 86% for all varying epochs (200,400, and 600).

Jiang and Diao [44] predicted diabetes using LapSVM. Initially, LapSVM was trained as a fully supervised learning classifier and an accuracy of 79.17% was obtained. Then it was trained as a semi-supervised learning classifier by converting labeled samples to unlabeled by randomly changing the samples in the training set. Accuracy bounced up to 82.29% which is greater than the previous one.

In the above mentioned research papers, diabetes prediction was performed using supervised and semi-supervised learning algorithms. Some of the commonly used ML algorithms are SVM, NB, DT, RF, and LapSVM. In this thesis, we performed comparison on supervised and semi-supervised ML algorithms for Type-3 diabetes prediction.

¹<https://www.cs.waikato.ac.nz/ml/weka/>

The experimentation methodology has been chosen to be performed. The five ML algorithms were trained on both datasets and a comparison of classifiers performance has been performed. A new dataset has been created from the existing one by binning some of the important features. The algorithms were trained on both binned and non-binned datasets. Five ML algorithms are selected and compared based on performance metrics namely accuracy, precision, recall, and f1-score. The method is described as follows.

1. Preprocessing the data to remove outliers, null and missing values.
2. Performing feature selection on the dataset and selecting an optimal number of features that impact the target variable.
3. Creating a new dataset by binning Age and BMI attributes, which is binned dataset.
4. Performing One-Hot Encoding on the binned dataset.
5. Splitting both datasets to train and test then performing Oversampling on both Training datasets.
6. Training each ML algorithm on each dataset and analyze the results.
7. Performing hyperparameter tuning to improve the performance of the model.
8. Selecting the efficient algorithm from each dataset.

4.1 Data Collection and Manipulation

The dataset used in this thesis is PIMA Indians Diabetes Dataset (PIDD). It consists of female patients aged 21 years and above from PIMA Indians Heritage. The dataset comprises 768 instances of which 500 patients are non-diabetic (represented by outcome 0) and the rest of them are diabetic (represented by outcome 1) which shows the imbalanced nature. There are nine features in the dataset. They are described as follows:

1. Pregnancies – Denotes the number of times patient is pregnant
2. Glucose – Plasma glucose concentration 2 hours in an oral glucose tolerance

3. Blood pressure – Diastolic blood pressure (mm Hg)
4. SkinThickness – Triceps skinfold thickness (mm)
5. Insulin – 2-Hour serum insulin (μ U/ml)
6. Body Mass Index (BMI) – Body Mass Index = (weight in kg/ (height in m)²)
7. DiabetesPedigreeFunction – Diabetes history in relatives
8. Age – Age in years
9. Outcome – Class label (0 means non-diabetic or 1 means diabetic)

The PIDD dataset is obtained from Kaggle open source database [8]. The second dataset created from the existing one by categorizing Age and BMI attributes.

The dataset is loaded using the Pandas library and the first five instances are viewed using head method.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Figure 4.1: Dataframe

Data preprocessing was performed on the dataset to find missing values, null values, duplicated rows. Further, outliers were detected in each feature using box plot. An outlier is defined as the data point outside the whiskers of the box plot.

Each attribute comprises of outliers. The quantile method was used to set the minimum and maximum threshold for Pregnancies, Diabetes Pedigree Function, and Glucose attributes. After detecting and removing outliers the dataset was reduced to 765 instances which were three less than the original dataset. Most of the outliers, which are identified were not outliers. They are valid values, which were found in pregnant women rarely.

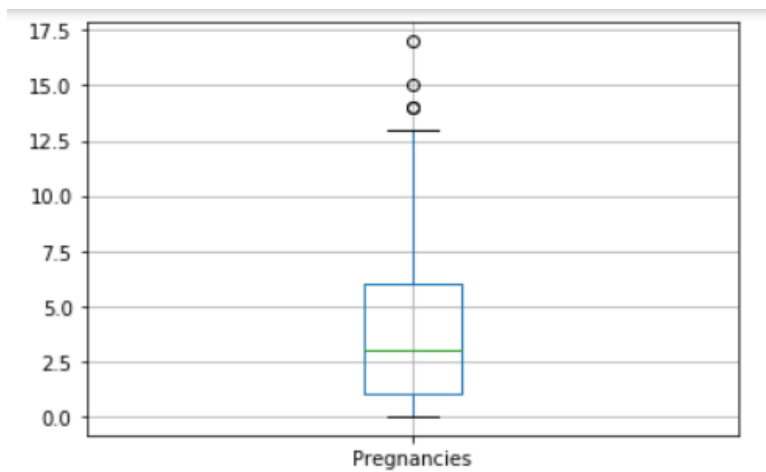


Figure 4.2: Box plot for Pregnancies

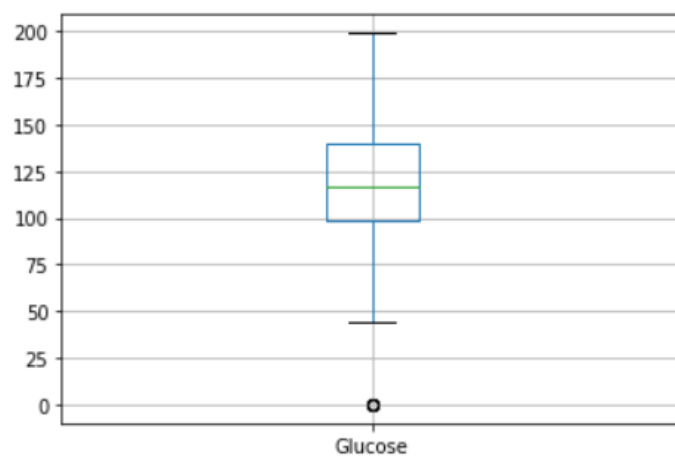


Figure 4.3: Box plot for Glucose

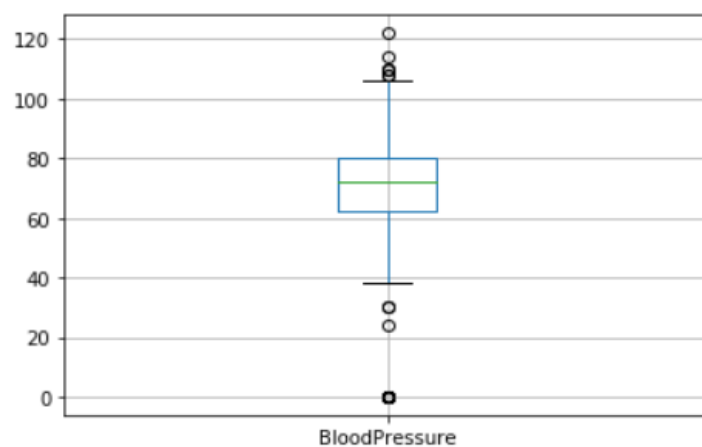


Figure 4.4: Box plot for Diastolic Blood Pressure

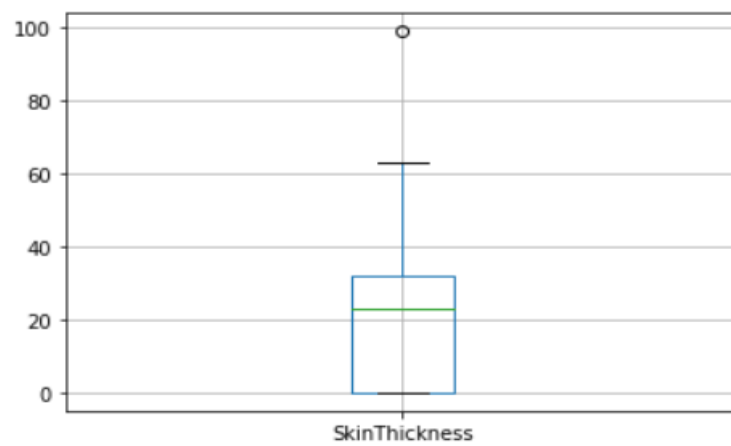


Figure 4.5: Box plot for Skin Thickness

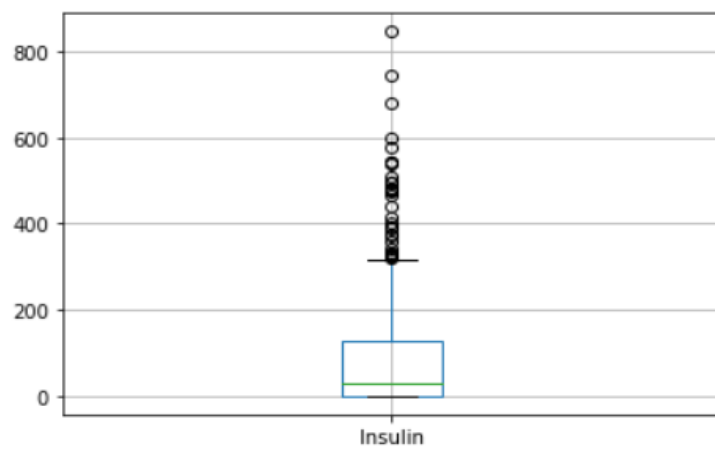


Figure 4.6: Box plot for Insulin

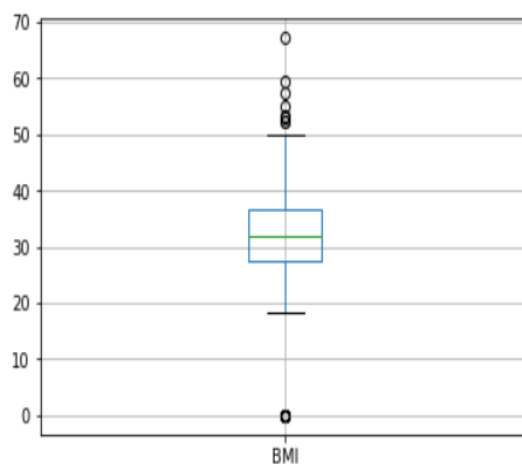


Figure 4.7: Box plot for BMI

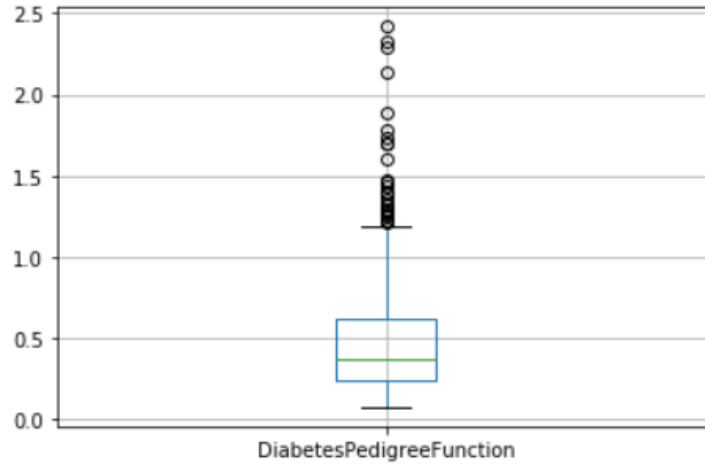


Figure 4.8: Box plot for Diabetes Pedigree Function

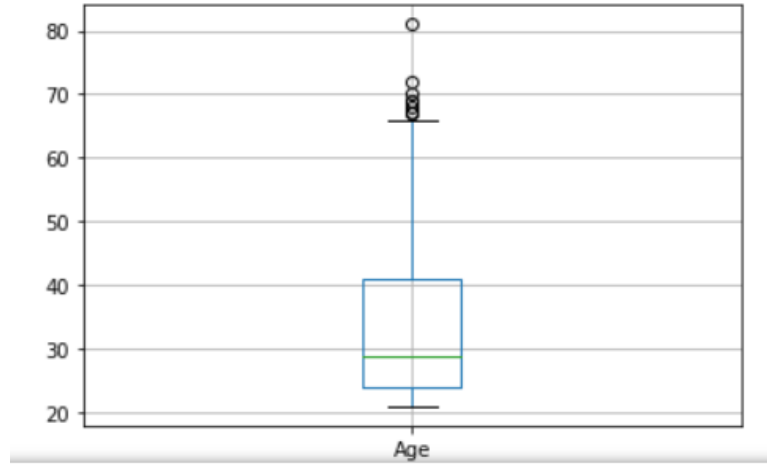


Figure 4.9: Box plot for Age

4.2 Feature Selection

To answer RQ1 Univariate Feature selection has been performed by Selecting k best features. The features which show an impact on the target variable are selected. The SelectKBest method uses the chi-square test for selecting optimal features. The attributes with higher chi-square scores were considered to be the features showing the maximum impact on the target variable. The top five features were taken as optimal features based on the scores which are provided in decreasing order and then used for training the models. The scores for top five features were shown in Table 4.1

After selecting the top five features, the dataset will be reduced to six columns (including the Outcome label). The updated dataset after removing features is shown in the below Figure 4.10.

Feature	Score
Insulin	2175.565273
Glucose	1411.887041
Age	181.303689
BMI	127.669343
Pregnancies	111.519691

Table 4.1: Top five Features

	Pregnancies	Glucose	Insulin	BMI	Age	Outcome
0	6	148	0	33.6	50	1
1	1	85	0	26.6	31	0
2	8	183	0	23.3	32	1
3	1	89	94	28.1	21	0
4	0	137	168	43.1	33	1
...
763	10	101	180	32.9	63	0
764	2	122	0	36.8	27	0
765	5	121	112	26.2	30	0
766	1	126	0	30.1	47	1
767	1	93	0	30.4	23	0

Figure 4.10: The data after removing non-essential features

The method which was discussed until now is the same for both datasets. Now the experiment has been performed in two parts. One experiment by performing binning on the dataset and the other without binning.

4.3 Binning

From the original dataset, the Age and BMI features were categorized. The dataset consists of women aged 21 years and above. Therefore, the Age [19] was divided into three categories namely Young Adult ($21 < \text{Age} < 39$), Middle-aged Adult ($40 < \text{Age} < 59$) and Old ($\text{Age} > 60$).

	Pregnancies	Glucose	Insulin	BMI	Age	Outcome	bmi_categories	age_categories
0	6	148	0	33.6	50	1	obese	Middle_Aged_Adult
1	1	85	0	26.6	31	0	overweight	Young_Adult
2	8	183	0	23.3	32	1	normal	Young_Adult
3	1	89	94	28.1	21	0	overweight	Young_Adult
4	0	137	168	43.1	33	1	obese	Young_Adult
...
763	10	101	180	32.9	63	0	obese	Old
764	2	122	0	36.8	27	0	obese	Young_Adult
765	5	121	112	26.2	30	0	overweight	Young_Adult
766	1	126	0	30.1	47	1	obese	Middle_Aged_Adult
767	1	93	0	30.4	23	0	obese	Young_Adult

Figure 4.11: Binning Age

BMI [17] was divided into four categories namely Underweight ($0 < \text{BMI} < 18.5$), Normal ($18.6 < \text{BMI} < 25$), Overweight ($25 < \text{BMI} < 30$) and Obese ($\text{BMI} \geq 30$).

	Pregnancies	Glucose	Insulin	BMI	Age	Outcome	bmi_categories
0	6	148	0	33.6	50	1	obese
1	1	85	0	26.6	31	0	overweight
2	8	183	0	23.3	32	1	normal
3	1	89	94	28.1	21	0	overweight
4	0	137	168	43.1	33	1	obese
...
763	10	101	180	32.9	63	0	obese
764	2	122	0	36.8	27	0	obese
765	5	121	112	26.2	30	0	overweight
766	1	126	0	30.1	47	1	obese
767	1	93	0	30.4	23	0	obese

Figure 4.12: Binning BMI

This process was done using the Pandas cut method. The parameters in the cut method include the dataframe along with the column on which binning to be

performed, range of values, and labels to each range. Therefore, new columns will be created after performing binning namely `age_categories` and `bmi_categories`. The number of labels should be one less than the number of bins.

After performing binning on Age and BMI, the columns `age_categories` and `bmi_categories` were dropped from the dataframe using pandas drop method.

In the next step, the columns were sorted in order such that the outcome column will be the last column reducing burden while splitting independent and dependent variables and easier to understand.

4.4 One-Hot Encoding

One-Hot Encoding is one of the categorical encoding techniques widely used in almost all researches where categorical variables are present in the data. In this technique, new columns will be added for each category. One-hot encoding was used to represent categorical data expressively. Mostly ML algorithms cannot work directly with categorical data. The categories have to be changed to numbers or binary vectors. This applies to both input and output variables.

Since there were three categories in age, therefore three additional columns will be added to the data frame. Each column describes each category. Each category was associated with a binary value (0 or 1). Each row represents a binary vector. For example, if a person belongs to the Old age category then the binary vector will be [0,0,1]. Similarly the same applies to BMI which has four categories. Therefore, four additional columns was added to the data frame.

This reduces the burden in classifying multi class attributes. After performing one-hot encoding, the features `age_categories` and `bmi_categories` were dropped from the data frame using the pandas drop method. Therefore, the number of features increased from nine to eleven.

4.5 Train-Test Split and Handling Imbalance Data

In the dataset, the outcome label was the dependent variable (denoted by y) and the rest of them were independent (denoted by x). Using train test split from the scikit-learn library, the independent and dependent variables, a test data size of 0.2, and random state was set to an integer value were passed as parameters. The training data comprises 80% and testing data of 20%. The count of the outcome variable was checked in both train and test data using `pandas.value_counts` method.

The number of non-diabetic patients were 500 (indicated by outcome 0) and 268 patients were diabetic (indicated by outcome 1) which shows the imbalance nature of the dataset. Therefore, Oversampling [46] [34] technique was used to mitigate the imbalance nature. Oversampling was performed on both the binned and non-binned training datasets separately. The number of samples in both training sets was equally distributed. The data has increased with an equal number of the target value. Now the model cannot be completely biased. After performing oversampling, the number of samples in the training data increased.

4.6 Training and Testing The Models

The training and testing of the model had been performed separately for each dataset. The DT, FC, SVM, BNB, and GNB algorithms were imported initially from the sklearn library. The training dataset is used to build the models by calling the fit method and providing `x_train` and `y_train` as parameters. Then the models were ready for the prediction. The testing data was provided for the prediction. The accuracy, precision, recall, and f1-score were calculated for each model on both datasets. The accuracy, precision, recall, and f1-score scores were discussed in the Results section.

The training and testing for the LapSVM model were different from all the other models. The dataset was split into 80% training data and 20% testing data. Then the LapSVM class was executed. The LapSVM code [5] in Python was found in GitHub. The class was instantiated. All the methods in the LapSVM class can be accessed by its object. The training dataset was divided into two equal parts then one part was considered as label dataset and the other was unlabeled.

The target attribute was dropped from one trained dataset, which refers unlabeled data. Therefore, the class has the fit method, where the model will be trained by using a labeled dataset. In the fitting process, the LapSVM class has done the computing adjacent matrix, computing laplacian graph, computing kernel metrics then the metrics was inverted and computing the alpha, beta finally. Then the test dataset was given for the method accuracy in the LapSVM class. The accuracy method will predict the values and measure the accuracy then return the accuracy score. Similarly, precision, recall, and f1-score were calculated for the prediction. Those scores were discussed in the Results section.

4.7 Hyper Parameter Tuning

The hyperparameters for every classifier were tuned by using the GSCV method. Initially, the GSCV was imported and the required parameters for the GSCV were estimator, `parameter_grid`, scoring, cross-validation, `n_jobs`. The estimator is the ML classifier and `parameter_grid` are the parameters of the classifier. All available parameters were given to the GSCV, and the training dataset will be used to fit the GSCV in order to give the correct parameters, which is suitable for efficient prediction. Then `best_score` will return the accuracy on the training dataset.

Each model's multiple parameters and stratified-10-fold cross validation are passed to the GSCV and then the training data was given to the GSCV. GSCV performs internal cross validation on 80% of the train data to set the hyperparameters. The GSCV returned the accuracy score on train data. Then `best_params_` command was used to get the best parameters for the models. The LapSVM model was a bit different from the other models. Therefore, manual hyperparameter tuning was performed for LapSVM to obtain high performance.

After building the models, evaluation metrics were applied to know how well the predictions were made on the models.

The models were trained and tested 10 times to obtain the average accuracies for both datasets. The accuracies were summarized in the below Table 5.1. The default parameters for some algorithms lead to better results. For the non-binned dataset, SVM and BNB obtained better accuracy with default hyperparameters. However, DT, RF, GNB, and LapSVM obtained a higher accuracy with hyperparameter tuning. Likewise for binned dataset DT, RF, SVM scored higher accuracy with the default parameters. However, for LapSVM and GNB obtained higher accuracy after hyperparameter tuning. BNB scored the same accuracy with the default parameters and in hyperparameter tuning.

Accuracy is considered when TPs and TNs are crucial. Accuracy measures only the predictions which are correct. But the disadvantage is, it is not considered when the classes are imbalanced. Thus, to overcome such problems precision, recall, and f1-score are used to handle when classes are misclassified and imbalanced.

When FPs and FNs are important, f1-score is used. It is the harmonic mean of precision and recall which shows the misclassification of classes. Using harmonic mean, extreme values will be penalized. It measures trade-off between precision and recall.

The precision scores of all the algorithms were described in Table 5.2. The LapSVM algorithms were scored high precision among other algorithms. For the non-binned dataset, the precision was 75.13% and 78.33% for the binned dataset. Higher precision means that more relevant results than irrelevant results are provided by an algorithm. Likewise, recall was also calculated for the algorithms and shown in Table 5.3. For the non-binned dataset, the recall score of LapSVM was 100%, which means LapSVM has no FNs. For the binned dataset, the high recall score was 92.45% for the BNB algorithm and LapSVM obtained 87.03%, which was the second highest recall score. When an algorithm has a high recall, it returns most of the relevant results whether or not irrelevant ones are also returned. The f1-scores were measured for selected algorithms and shown in Table 5.4. The LapSVM scored a high f1-score on both binned and non-binned datasets. High f1-score indicates that the model detects a smaller number of FPs and FNs. Note that in tables 5.1-5.4 the best performed algorithms are shown in bold.

Table 5.1: Accuracies on test datasets

Algorithm	Non-Binned Dataset	Binned Dataset
DT	73.13%	68.18%
RF	74.50%	72.72%
SVM	72.54%	67.5%
GNB	69.93%	68.83%
BNB	51.63%	58.44%
LapSVM	89.61%	86.93%

Table 5.2: Precision scores on test datasets

Algorithm	Non-Binned Dataset	Binned Dataset
DT	62.75%	54.27%
RF	63.65%	63.06%
SVM	60.86%	50.7%
GNB	60.6%	54.41%
BNB	41.8%	44.95%
LapSVM	75.13%	78.33%

Table 5.3: Recall scores on test datasets

Algorithm	Non-Binned Dataset	Binned Dataset
DT	48.33%	54.71%
RF	75%	66.03%
SVM	70%	60.37%
GNB	63.33%	69.81%
BNB	68.33%	92.45%
LapSVM	100%	87.03%

Table 5.4: F1-scores on test datasets

Algorithm	Non-Binned Dataset	Binned Dataset
DT	60.60%	54.71%
RF	68.85%	63.06%
SVM	64.40%	55.17%
GNB	58.18%	61.15%
BNB	51.57%	60.49%
LapSVM	87.34%	82.45%

Finally, the LapSVM, which is a semi-supervised learning technique has obtained the highest performance among the DT, RF, SVM, BNB, and GNB on both binned and non-binned datasets. The LapSVM was predicted with 87.34% accuracy on the non-binned dataset and 82.45% accuracy on the binned dataset. The reason was that the unsupervised learning methods learns new patterns which were undiscoverable

in supervised learning methods. RF algorithm was the second best among other ML algorithms on both datasets. Comparing performances between binned and non-binned datasets, non-binned dataset produced better results. Hence the non-binned dataset is suitable for Type-3 diabetes prediction.

Supervised and semi-supervised algorithms were used for the Type-3 diabetes prediction on the PIDDD dataset. From the original dataset, a new dataset has been created by binning Age and BMI features. The selected supervised and semi-supervised algorithms were built and tested using both datasets. Then the efficient algorithm was identified by comparing the evaluation metrics score of selected algorithms on both datasets. Our project intended to help doctors to identify diabetes. Doctors are important for curing and treating diabetes patients. In this section, the answers for RQ's were briefly explained.

RQ1:What are the important features which influence in prediction of gestational (Type-3) diabetes?

Answer:

The feature selection is the crucial method, which will be used for selecting the important features, that had more correlation with the target label. Having irrelevant features leads to a negative impact on the performance of the model. Feature selection should be performed before designing a model. The advantages of performing feature selection are that it reduces overfitting, improves accuracy, and reduces training time.

In this thesis, the SelectKBest feature selection was performed to find the important features by using the feature's chi-square score. Insulin, Glucose, Age, BMI, and Pregnancies were the five best features in increasing order as shown in Table 4.1. When features increased or decreased, it showed an impact on the performance of the model. When five features are selected, the models produced accurate results. Therefore, these were the important features, which influence in prediction of Type-3 diabetes.

RQ2:Which is the efficient ML algorithm among SVM, NB, DT, RF, and LapSVM for each dataset to predict gestational (Type-3) diabetes?

Answer:

The efficient algorithm for Type-3 diabetes prediction among DT, RF, SVM, GNB, BNB, and LapSVM was identified by comparing accuracies, precision, recall and f1-score. The comparison of the accuracy, precision, recall, and f1-scores of selected algorithms were discussed as follows.

Comparison Based on Accuracy

The final accuracy was taken as the average of the 10 accuracies of each classifier. The accuracy obtained by the DT was 73.13%, RF with 74.50%, SVM with 72.54%, GNB with 69.93%, BNB with 51.63%, and LapSVM with 89.61% for the non-binned dataset. The accuracy obtained by the DT was 68.18%, RF with 72.72%, SVM with 67.50%, GNB with 68.86%, BNB with 58.44%, and LapSVM with 86.93% for the binned dataset. These accuracies were shown in the bar graph as Figure 6.1.

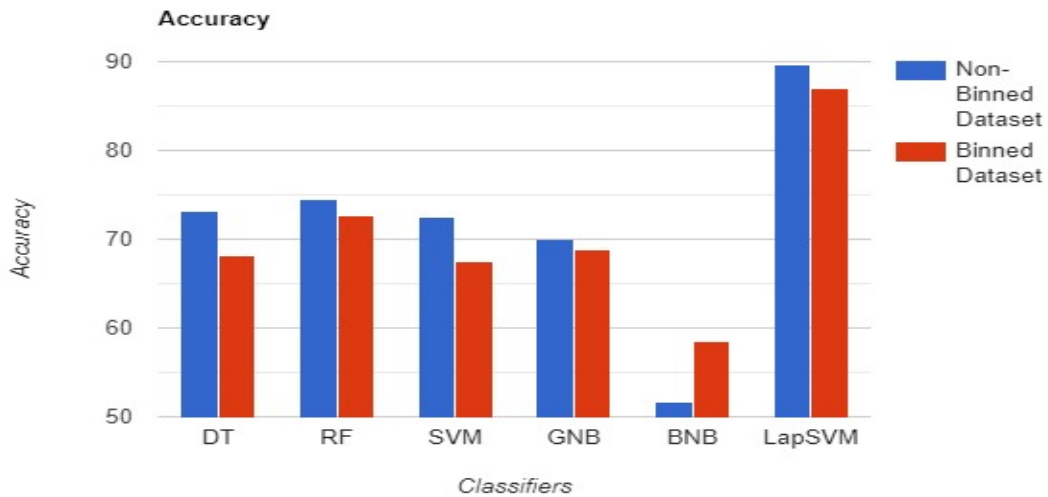


Figure 6.1: Accuracies for non-binned and binned datasets

From Figure 6.1, LapSVM which is a semi-supervised learning method obtained the highest accuracy among all algorithms for both datasets. Based on the accuracy RF, which is a supervised learning method obtained the second best accuracy. The BNB algorithm scored the least accuracy for both datasets. From the dataset point of view for all selected algorithms except BNB non-binned dataset was efficient for prediction than binned dataset.

Comparison Based on Precision

The precision obtained by DT was 62.75%, RF with 63.65%, SVM with 60.86%, GNB with 60.6%, BNB with 41.8%, and LapSVM with 75.13% which is the highest of all for non-binned dataset. For binned dataset, the precision obtained by DT was 56.36%, RF with 60.71%, SVM with 50%, GNB with 54.41%, BNB with 44.95% and for LapSVM with 78.33% which is the highest in this dataset also. When FPs are less, it results in high precision.

The highest precision among all ML algorithms is LapSVM for both datasets as shown in Figure 6.2. For both datasets, the BNB classifier attained the least precision. The second best precision score was obtained by RF for both datasets. For the non-binned dataset, DT, RF, SVM, GNB, and LapSVM algorithms scored high precision than the binned dataset.

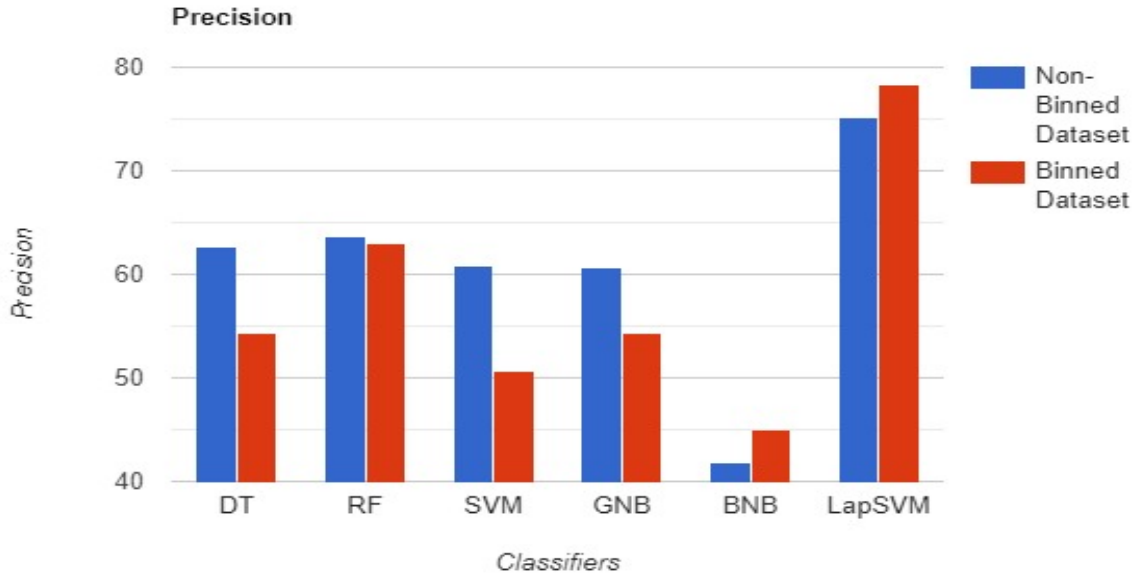


Figure 6.2: Precision for non-binned and binned datasets

Comparison Based on Recall

The recall obtained by DT was 48.33%, RF with 75%, SVM with 70%, GNB with 63.33%, BNB with 68.33% and for LapSVM with 100% which is the highest of all for non-binned dataset. For binned dataset, the precision obtained by DT was 58.49%, RF with 66.15%, SVM with 60.37%, GNB with 69.81%, BNB with 92.45% and for Lap SVM with 78.33%. BNB produces highest recall score of all. When FNs are less, it results in high recall.

From Figure 6.3 the LapSVM obtained the highest recall for the non-binned dataset and BNB scored the highest recall for the binned dataset. The DT classifier scored the least recall for both datasets. The second best recall score was scored by BNB for binned dataset and RF for the non-binned dataset. DT, GNB, and BNB algorithms obtained high recall on the binned dataset. However, RF, SVM, and LapSVM scored high recall on the non-binned dataset.

By considering the above comparison results the accuracy and precision were more to the same algorithms and non-binned dataset. However, the recall score was slightly opposite in the case of datasets i.e., some algorithms were performed well on binned dataset and other on non-binned dataset. In addition, LapSVM was scored high recall for non-binned and BNB was scored high for the binned dataset. The results cannot be concluded by the above comparison. Therefore, f1-score comparison was conducted as follows.

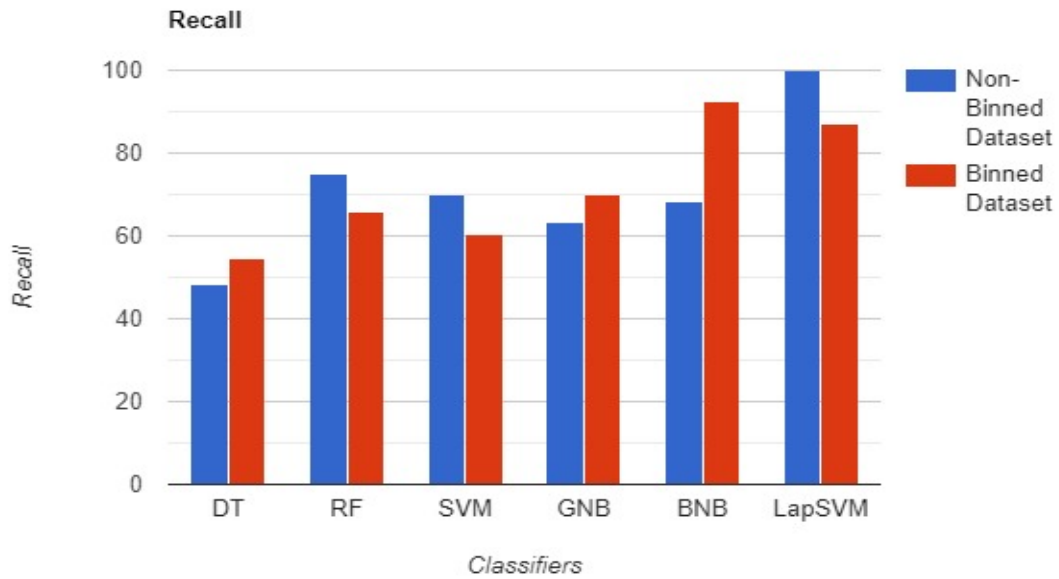


Figure 6.3: Recall for non-binned and binned datasets

Comparison Based on f1-Score

The f1-score obtained by the DT was 60.60%, RF with 68.85%, SVM with 64.40%, GNB with 58.18%, BNB with 51.57%, and LapSVM with 87.34% for the non-binned dataset. The f1-score obtained by the DT is 55.31%, RF with 60.97%, SVM with 58.68%, GNB with 59.9%, BNB with 60.0%, and LapSVM with 82.45% for the binned dataset. These scores are shown in the Figure 6.4.

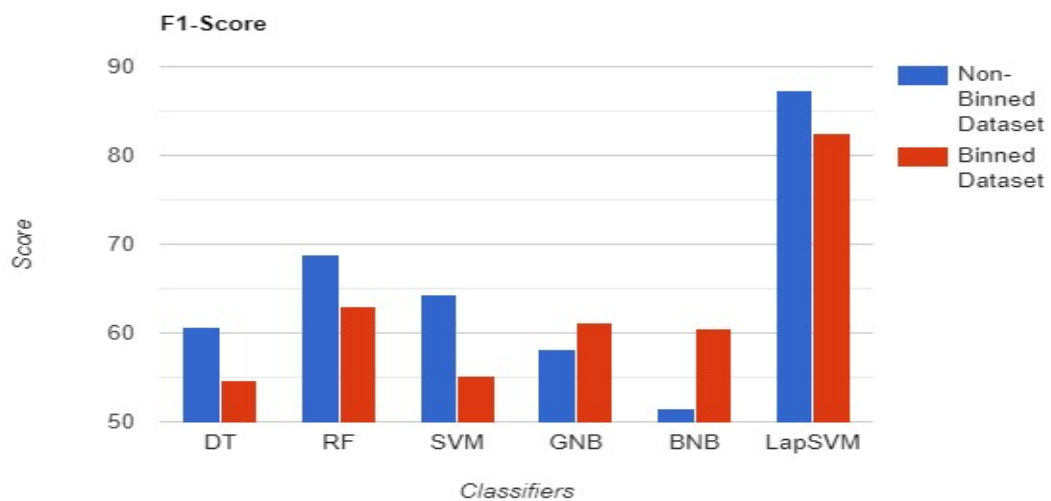


Figure 6.4: F1-score for binned and non-binned datasets

From Figure 6.4, LapSVM attained the highest f1-score among other ML techniques on both datasets. The lowest f1-score was obtained by BNB for the non-binned dataset and DT for the binned dataset. RF obtained the second best f1-score for both datasets. All selected algorithms were performed well except GNB and BNB on non-binned dataset than binned dataset.

The accuracy, precision, and f1-score were high for the selected algorithms on non-binned dataset. Whereas the recall was above 70% for SVM, RF, and LapSVM algorithms on non-binned dataset and DT, GNB, and BNB algorithms were obtained above 55% on the binned dataset. By this the non-binned dataset was preferred for diabetes prediction is concluded.

Based on the accuracies obtained, LapSVM outperformed among other ML classifiers in both binned and non-binned datasets. The second best algorithm was RF, which is a supervised learning algorithm. LapSVM was 89.61% accurate on non-binned dataset and 86.93% accurate on binned dataset. Whereas RF was 74.50% accurate on the non-binned dataset and 72.72% accurate on the binned dataset. Since the dataset is imbalanced, accuracy is not the only metric to be considered. Other metrics like precision, recall, and f1-score are considered. LapSVM scores high precision and f1-score on both datasets. However, LapSVM was scored high recall for the binned dataset as well. The LapSVM model was trained by both labeled and unlabeled data. The half dataset was given as labeled and the other half was given as unlabelled by removing the target variable to the model for training. Then the LapSVM performed pattern recognition on unlabelled data and predicted the outcome values.

Thus, the LapSVM, which was hybrid from both supervised and unsupervised ML techniques scored the best performance. By considering all these comparisons the LapSVM is the efficient ML algorithm among SVM, NB, DT, and RF for both datasets to predict Type-3 diabetes.

7.1 Conclusion

Diabetes is a disease that should not be neglected. It must be brought under control as fast as possible in the next few years. In this thesis, Type-3 diabetes prediction was performed using the PIDD dataset. Important features were selected on the dataset and a new dataset was created by binning Age and BMI features. Selected ML algorithms were built using each dataset and evaluated. Hyperparameter tuning was performed on each model to improve the performance. LapSVM was 89.61% accurate on non-binned dataset and 86.93% accurate on binned dataset. Whereas RF was 74.50% accurate on the non-binned dataset and 72.72% accurate on the binned dataset. The best algorithm was LapSVM and second best algorithm was RF. The non-binned dataset predicted accurate results for most of the algorithms. Hence, non-binned dataset was the suitable dataset for Type-3 diabetes prediction.

7.2 Future Work

In the future, diabetes can be performed with a precise dataset with all types of diabetes and consisting of both genders. Also, performance can be further improved by using Deep Learning techniques like Multi-Layer Perceptron (MLP), Artificial Neural Networks (ANN). ML algorithms worked well on training data but failed to work the same way on the new data due to overfitting. This can be solved by using the drop-out method. Dropout is only used during the training of a model and is not used when evaluating the skill of the model.

Bibliography

- [1] “1.10. Decision Trees — scikit-learn 0.24.2 documentation.” [Online]. Available: <https://scikit-learn.org/stable/modules/tree.html>
- [2] “1.4. Support Vector Machines — scikit-learn 0.24.2 documentation.” [Online]. Available: <https://scikit-learn.org/stable/modules/svm.html#support-vector-machines>,
- [3] “1.9. Naive Bayes — scikit-learn 0.24.2 documentation.” [Online]. Available: https://scikit-learn.org/stable/modules/naive_bayes.html#naive-bayes,
- [4] “Diabetes.” [Online]. Available: <https://www.who.int/westernpacific/health-topics/diabetes>
- [5] “HugoooPerrin/semi-supervised-learning.” [Online]. Available: <https://github.com/HugoooPerrin/semi-supervised-learning>
- [6] “An introduction to seaborn — seaborn 0.11.1 documentation.” [Online]. Available: <https://seaborn.pydata.org/introduction.html>
- [7] “Matplotlib: Python plotting — Matplotlib 3.4.2 documentation.” [Online]. Available: <https://matplotlib.org/>
- [8] “Pima Indians Diabetes Database.” [Online]. Available: <https://kaggle.com/uciml/pima-indians-diabetes-database>
- [9] “scikit-learn: machine learning in Python — scikit-learn 0.24.2 documentation.” [Online]. Available: <https://scikit-learn.org/stable/>
- [10] “sklearn.ensemble.RandomForestClassifier — scikit-learn 0.24.2 documentation.” [Online]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html?highlight=random%20forest#sklearn.ensemble.RandomForestClassifier>
- [11] “sklearn.metrics.accuracy_score — scikit-learn 0.24.2 documentation.” [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html
- [12] “sklearn.metrics.confusion_matrix — scikit-learn 0.24.2 documentation.” [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html?highlight=confusion%20matrix#sklearn.metrics.confusion_matrix
- [13] “sklearn.metrics.f1_score — scikit-learn 0.24.2 documentation.” [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html?highlight=f1%20score#sklearn.metrics.f1_score
- [14] “sklearn.metrics.precision_score — scikit-learn 0.24.2 documentation.” [Online].

- Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_score.html?highlight=precision#sklearn.metrics.precision_score
- [15] “sklearn.metrics.recall_score — scikit-learn 0.24.2 documentation.” [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall_score.html?highlight=recall#sklearn.metrics.recall_score
 - [16] “Supervised Learning | SpringerLink.” [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-540-75171-7_2
 - [17] “Table 1 . Age intervals and age groups.” [Online]. Available: https://www.researchgate.net/figure/Age-intervals-and-age-groups_tbl1_228404297
 - [18] “Semi-supervised Learning,” in *Encyclopedia of the Sciences of Learning*, N. M. Seel, Ed. Boston, MA: Springer US, 2012, pp. 3036–3036. [Online]. Available: https://doi.org/10.1007/978-1-4419-1428-6_2402
 - [19] “Defining Adult Overweight & Obesity | Overweight & Obesity | CDC,” Apr. 2021. [Online]. Available: <https://www.cdc.gov/obesity/adult/defining.html>
 - [20] “Manifold regularization,” Jan. 2021, page Version ID: 1000158791. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Manifold_regularization&oldid=1000158791
 - [21] A. S. Alanazi and M. A. Mezher, “Using Machine Learning Algorithms For Prediction Of Diabetes Mellitus,” in *2020 International Conference on Computing and Information Technology (ICCIT-1441)*, Sep. 2020, pp. 1–3.
 - [22] A. Bastaki *et al.*, “Diabetes mellitus and its treatment,” *International journal of Diabetes and Metabolism*, vol. 13, no. 3, p. 111, 2005.
 - [23] J. Brownlee, “Why One-Hot Encode Data in Machine Learning?” Jul. 2017. [Online]. Available: <https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/>
 - [24] —, “How to Choose a Feature Selection Method For Machine Learning,” Nov. 2019. [Online]. Available: <https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/>
 - [25] —, “Random Oversampling and Undersampling for Imbalanced Classification,” Jan. 2020. [Online]. Available: <https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/>
 - [26] N. Cho, J. Shaw, S. Karuranga, Y. Huang, J. da Rocha Fernandes, A. Ohlrogge, and B. Malanda, “Idf diabetes atlas: Global estimates of diabetes prevalence for 2017 and projections for 2045,” *Diabetes research and clinical practice*, vol. 138, pp. 271–281, 2018.
 - [27] E. R. F. Collaboration *et al.*, “Diabetes mellitus, fasting blood glucose concentration, and risk of vascular disease: a collaborative meta-analysis of 102 prospective studies,” *The Lancet*, vol. 375, no. 9733, pp. 2215–2222, 2010.
 - [28] Z. Ghahramani, “Unsupervised Learning,” in *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2 - 14, 2003, Tübingen, Germany, August 4 - 16, 2003, Revised Lectures*, ser. Lecture Notes in Computer Science, O. Bousquet, U. von Luxburg, and G. Rätsch, Eds. Berlin, Heidelberg: Springer, 2004, pp. 72–112. [Online]. Available:

- https://doi.org/10.1007/978-3-540-28650-9_5
- [29] M. K. Hasan, M. A. Alam, D. Das, E. Hossain, and M. Hasan, "Diabetes Prediction Using Ensembling of Different Machine Learning Classifiers," *IEEE Access*, vol. 8, pp. 76 516–76 531, 2020, conference Name: IEEE Access.
 - [30] J. D. Hunter, "Matplotlib: A 2d graphics environment," *IEEE Annals of the History of Computing*, vol. 9, no. 03, pp. 90–95, 2007.
 - [31] K. Imam, "Gestational Diabetes Mellitus," in *Diabetes: An Old Disease, a New Insight*, ser. Advances in Experimental Medicine and Biology, S. I. Ahmad, Ed. New York, NY: Springer, 2013, pp. 24–34. [Online]. Available: https://doi.org/10.1007/978-1-4614-5441-0_4
 - [32] A. Jović, K. Brkić, and N. Bogunović, "A review of feature selection methods with applications," in *2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO)*. Ieee, 2015, pp. 1200–1205.
 - [33] J. J. Khanam and S. Y. Foo, "A comparison of machine learning algorithms for diabetes prediction," *ICT Express*, Feb. 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405959521000205>
 - [34] A. Liu, J. Ghosh, and C. E. Martin, "Generative oversampling for mining imbalanced datasets." in *DMIN*, 2007, pp. 66–72.
 - [35] C. D. Mathers and D. Loncar, "Projections of global mortality and burden of disease from 2002 to 2030," *PLoS medicine*, vol. 3, no. 11, p. e442, 2006.
 - [36] R. G. McClarren, "Chapter 4 - NumPy and Matplotlib," in *Computational Nuclear Engineering and Radiological Science Using Python*, R. G. McClarren, Ed. Academic Press, Jan. 2018, pp. 53–74. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780128122532000054>
 - [37] W. McKinney, *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*. " O'Reilly Media, Inc.", 2012.
 - [38] T. M. Mitchell, *Machine Learning*, ser. McGraw-Hill series in computer science. New York: McGraw-Hill, 1997.
 - [39] N. Nai-arun and R. Moungrmai, "Comparison of Classifiers for the Risk of Diabetes Prediction," *Procedia Computer Science*, vol. 69, pp. 132–142, Jan. 2015. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050915031786>
 - [40] D. M. Nathan, "Long-term complications of diabetes mellitus," *New England Journal of Medicine*, vol. 328, no. 23, pp. 1676–1685, 1993.
 - [41] J. M. Olefsky, "Prospects for research in diabetes mellitus," *Jama*, vol. 285, no. 5, pp. 628–632, 2001.
 - [42] P. Sonar and K. JayaMalini, "Diabetes Prediction Using Different Machine Learning Approaches," in *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, Mar. 2019, pp. 367–371.
 - [43] S. Tosi, *Matplotlib for Python developers*. Packt Publishing Ltd, 2009.
 - [44] J. Wu, Y.-B. Diao, M.-L. Li, Y.-P. Fang, and D.-C. Ma, "A semi-

supervised learning based method: Laplacian support vector machine used in diabetes disease diagnosis,” *Interdisciplinary Sciences: Computational Life Sciences*, vol. 1, no. 2, pp. 151–155, Jun. 2009. [Online]. Available: <https://doi.org/10.1007/s12539-009-0016-2>

- [45] H. Yang, “Data preprocessing,” 2018.
- [46] B. W. Yap, K. Abd Rani, H. A. Abd Rahman, S. Fong, Z. Khairudin, and N. N. Abdullah, “An application of oversampling, undersampling, bagging and boosting in handling imbalanced datasets,” in *Proceedings of the first international conference on advanced data and information engineering (DaEng-2013)*. Springer, 2014, pp. 13–22.

Appendix A

Best parameters for the selected algorithms

A.1 Best parameters for DT

Non-binned dataset

```
{'criterion':  
'entropy',  
'max_depth': None}
```

Binned dataset

```
{max_depth=50}.
```

A.2 Best parameters for RF

Non-binned dataset

```
{'bootstrap': True, 'max_depth': 4,  
'max_features': 'auto',  
'min_samples_leaf': 1,  
'min_samples_split': 5,  
'n_estimators': 56}
```

Binned dataset

```
{'bootstrap': True,  
'max_depth': 4,  
'max_features': 'auto',  
'min_samples_leaf': 1,  
'min_samples_split': 2,  
'n_estimators': 64}
```

A.3 Best parameters for SVM

Non-binned dataset

```
{'C': 1,  
'gamma': 0.3,  
'kernel': 'rbf'}
```

Binned dataset

```
{'C': 1,  
'gamma': 0.7,  
'kernel': 'rbf'}
```

A.4 Best parameters for BNB

Non-binned dataset

```
{'alpha': 1.0,  
'binarize': 1.0,  
'class_prior': None,  
'fit_prior': True}
```

Binned dataset

```
{'alpha': 1.0,  
'binarize': 0.0,  
'class_prior': None,  
'fit_prior': True} for the binned dataset.
```

A.5 Best parameters for GNB

Non-binned dataset

```
{'var_smoothing': 0.0002848035868435802} for the normal dataset
```

Binned dataset

```
{'var_smoothing': 1.519911082952933e-06} for the binned dataset.
```

A.6 Best parameters for LapSVM

Non-binned dataset

```
{'C': [20],  
'kernel': ['rbf'],  
'gamma': [6,100000]}
```

Binned dataset

```
{'C': [5],  
'kernel': ['rbf'],  
'gamma': [0.1,10000]}
```

