ABSTRACT

The study aimed to address the challenges of obtaining high accuracy in detecting diabetes while keeping the complexity of the data mining model low. This study also defined a solution to the challenge i.e., is to utilize five feature selection algorithms and six classification algorithms to detect Type 2 diabetes. Support Vector Machine with features obtained from Forward Stagewise Selection achieved the peak accuracy of 91.5582% in 10-fold cross-validation This study contributes to the development of efficient tools for detection of diabetes, focusing on simplicity for broader accessibility.

Keywords: ***SVM, Forward Stagewise Selection, Data Mining***

LIST OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

When a body does not produce enough insulin or the insulin produced by our body is not sufficient then a condition arises in our body known as Diabetes mellitus[1]. Individuals diagnosed with this disease are unable to regulate their blood sugar levels in their bodies. Diabetes could lead to other life-endangering diseases such as diabetic neuropathy, retinopathy, kidney failure, and other heart-related diseases if it is not diagnosed and treated effectively at its early stage[2].

Type 1 diabetes is known for a deficit in the production of insulin and requires daily dosages of insulin. According to the reports it was found that in the year 2017, 9 million individuals were impacted by type 1.

Type 2 diabetes is the majority prevalent type of diabetes, according to the International Diabetes Federation (IDF), 10.5% of adults aged 20 to 79 have diabetes. Type 2 diabetes affects more than 90% of diabetics[4]. Type 2 diabetes alters how the body uses sugar as fuel. This prevents the body from utilizing insulin as it should, which, if addressed, can result in elevated blood sugar levels. [3].

Gestational diabetes is caused by hyperglycemia, which is defined as blood glucose levels above normal but below those of diagnosed diabetes. This happens when a woman is pregnant. Diabetes increases the risk of difficulties during pregnancy and childbirth for women who have it. There is a higher likelihood that these mothers and their children may develop Type 2 diabetes in the future.[3].

It is projected that 537 million adults globally will have diabetes in 2021; by 2030, the number will rise to 643 million, and by 2045, it is expected to reach 783 million. The diabetic population is set on increasing as the years go by, the many factors that contribute to the rise of diabetes are urbanization, the aging population, decreasing levels of physical activity, increase in weight, and obesity prevalence [4].

Data mining is the technique used to sort through large sets of data to identify patterns and relationships which will help solve business and even medical problems through data analysis. Numerous techniques have been developed to mine data in recent years, they include classification, clustering, association, evolution, pattern matching, characterization, data visualization, and meta-rule-guided mining[5].

The purpose of this study is to examine the behaviour of several boosting algorithms on a training data set. This paper will use weighted k-NN to preprocess the data and make it readable and noise-free, and it will also use a variety of feature selection techniques to take out the most relevant subset of features from a given dataset and use several powerful classification algorithms to detect Type 2 diabetes from a given set of labeled input data and find the most accurate method for detecting diabetes in its early stage. Classification, a supervised learning method, that matches data into one of the several predefined classes[6].

There have been several efforts for detecting diabetes early in the past and they have given varying results. The study conducted by S.Saru et al (2019) gave a high accuracy of 90.36% using an ensemble method[7], In the study done by L.Chaves et al (2021) they normalized the data set that they had, containing information from 520 patients between the ages (i.e 16 to 90), and they correctly predicted 510 records out of 520 instances using Neural Network, observing the highest accuracy of 98.1%[1].

1.1 General overview of the problem

In the current times, many medical diagnoses are performed learning machine learning, as they are accurate, and fast. There are numerous ways of detecting diabetes using data mining in the present, but most of these algorithms are highly optimized and complex for a regular person with no coding knowledge to understand. This research aims to search for an efficient way of detecting diabetes i.e., getting good accuracy while keeping the complexity of the used algorithms as low as possible, so that the medical professionals may be able to change the parameters themselves to fit their use case.

1.2 Feasibility Study

1.2.1 Economic Feasibility

- two datasets are being used in this research one is a free dataset provided by Kaggle.com and the other one is the dataset given to the study team by their mentor. The tools used in this study will be open source and the algorithms used will be written by the study team.
- So, there will be no need for any external funding.

1.2.2 Technical Feasibility

- Procured two datasets, the Pima Indians diabetes dataset, with 768 records, including information on glucose, and insulin levels, blood pressure, and such. The other dataset contains 2535 records, with information such as family history, abortions, and so on.
- This study will be conducted using a modern laptop with a reasonable processing capability.
- This study will be using Python libraries (sci-kit-learn, pandas, numpy) and IDE (Visual Studio Code).

1.2.3 Operational Feasibility

- The team conducting this study consists of MCA students and is equipped with the knowledge and skills required to complete this study.

1.2.4 Schedule Feasibility

- Data collection: One Week.
- Data preprocessing and Feature selection: One month
- Algorithm development and training: 3 months
- Model Evaluation: One week
- Report writing: whole duration of the project

By conducting this feasibility study, we assessed that this study is economically, technically, operationally, and schedule-wise feasible.

1.3 Literature Survey

Many studies, like the one by S. Saru et al. [7], have looked into finding and predicting diabetes using data-driven methods. In one instance, they suggested a machine learning model to predict diabetes. In this research, they have used techniques like Principal Component Analysis to take out important features and various classification methods such as k-nearest Neighbour, Random Forest, Support Vector Machine, Decision tree, and Logistic Regression. However, this research pointed out a research gap of not having enough sample data and suggested exploring different datasets.

Another study, done by Luis Chaves et al. [1], focused on bringing Data Mining into systems that might help doctors make decisions. So they focused on making improvements in disease detection and public health management, using methods like Neural Network, Ada Boost,

KNN, Random Forest, Naive Bayes, and SVM. In these studies, they found that the Neural Network has the highest prediction accuracy of 98.1%. So, this research showed that using a mix of techniques and preparing the data well, along with reducing its complexity, is very important.

Another study, done by F. A. Khan et al. [8] first sorted out methods for finding diabetes based on the models they used, and then they also showed how important it is to prepare the data well and use a mix of different methods. This research also highlighted the importance of selecting the right features, cleaning up the data, and making it very simple to get accurate results.

Another research done by Rufo DD et al. [9], worked on finding diabetes in places where resources are limited. They used a method called Light Gradient Boosting Machine (LightGBM) and got really good results with 98.1% accuracy and 98.1% AUC rates. Also, this research showed us the importance of using current circumstances and identifying alternative methods to determine whether a person has diabetes or not.

Similarly, a study by Birjais and team [10]looked into Gradient Boosting, Logistic Regression, and Naive Bayes for finding diabetes. They found accuracies of 86%, 79%, and 77%, respectively. These studies have mainly focused on preparing the data in advance and understanding the co-relation between body mass index (BMI) and blood glucose levels.

Meanwhile Kazerouni et al. [11], compared different ways to find diabetes and found that Support Vector Machine (SVM) and Logistic Regression worked well, achieving 95% accuracy. In this study, Kazerouni et al. [6] highlighted the importance of selecting the right method and diagnosis.

A study conducted by Wee et al. [2] With diabetes becoming more common, focused on the need for models that use data to detect it. They discussed using simple techniques, the importance of selecting specific features, and the challenges of using deep learning models.

A study by Uddin et al. [12] used a mix of methods like Logistic Regression, Linear Regression, k-nearest neighbor, Naive Bayes, Random Forest, Support Vector Machine, and Decision Tree to find diabetes. They got high accuracy rates of 97% and 80% on diabetes

datasets. This research study suggested that Random Forest is really good for prediction and also can be used for diabetes detection.

The research by Sneha et al. [13] aimed to find diabetes early using important features and machine learning algorithms. They designed a predictive algorithm that closely matches clinical outcomes. The highest accuracy was given by decision tree and Random Forest algorithms, while Naive Bayes gave the best accuracy.To improve accuracy, it focuses on choosing the right features.

Another study by Rian Budi Lukmanto et al. [14] addresses the increasing number of diabetes cases by suggesting a framework that combines F-Score Feature Selection and Fuzzy Support Vector Machine. They used a lot of patient records and got a promising 89.02% accuracy in predicting diabetes cases. This research suggests that we can further improve through techniques like clustering or genetic algorithms.

Another study by Kopitar et al. [15] compared machine learning models with traditional models and found that adding new data, especially in LightGBM-based models, improves accuracy and stability. While machine learning models didn't significantly outperform traditional models, they were better at visualization. This research suggests exploring ensemble methods for healthcare decisions.

The study by Chang V et al. [16] introduced an e-diagnosis system for the Internet of Medical Things (IoMT) to diagnose type 2 diabetes. In this research, they used models like Naive Bayes, J48 decision tree, and random forest while the random forest classifier gave the highest accuracy. Also, this research suggests to use of ML within IoMT for diabetes diagnosis and remote monitoring.

The study by Ravinder Ahuja et al. [17] used four different methods, and they discovered that the C4.5 Decision Tree gave the highest accuracy. Also, it gave useful information for healthcare professionals that might be helpful for the prediction of early diabetes.

Ravinder Ahuja and team [17] did a study using machine learning to make diabetes predictions better. They used a dataset from Pima Indians with different ways of organizing the data and pointed out that the Multilayer Perceptron classifier worked well. Using this he thought about getting the data ready, choosing important features, and comparing different

methods, giving helpful ideas for predicting diabetes well. They also suggested looking into other methods and ways to choose features in the future.

Lastly, a paper by Aishwarya Iyer et al. [18] addressed the increasing diabetes prevalence, especially in women. She introduced a new way to diagnose diabetes using Decision Tree and Naive Bayes algorithms and information technology. Also, this research identifies possible predictive factors and anticipates global data integration to increase diagnostic accuracy it makes a significant addition to diabetes diagnosis driven by technology.

Collectively, these studies contribute to the evolving landscape of diabetes detection and prediction methodologies, offering diverse techniques to address this global health challenge.

1.4 Problem Definition

The majority of the machine learning models currently available are complex, as to get a high success rate in detecting diabetes the developers of these models optimize the algorithms as much as possible. The challenge of this research lies in the fact that to get higher accuracy a more complex model is required.

1.5 Solution Strategy

To get high accuracy while keeping the complexity of the machine learning model low, the team will use five feature selection algorithms, to select features from a dataset, and then use three ensemble(boosting) algorithms to detect diabetes, from these newly acquired datasets of selected features. The plan is to keep the parameters of these algorithms as simple as possible.

- Data Acquisition and Preprocessing: Acquire medical records of numerous individuals and then do any preprocessing step required.
- Feature Selection: Select the features from the datasets, using the five algorithms.
- Split dataset: Datasets are split into training and testing datasets.
- Model Development along with Training: To train the model using the training datasets, while setting the parameters to default.
- Evaluate: evaluate the model using metrics such as accuracy.

2. PROJECT PLAN

2.1 Hardware and Software Requirements

- Hardware:
    - Processor: Intel core i3 7th Gen+ or AMD Ryzen 3 3rd Gen+
    - Memory: 4GB DDR3 RAM
    - Storage: 50GB
    - 64-bit operating system, x64-based processor or 32-bit operating system

- Software:
    - Windows 7 and above
    - Python 3.11.3
    - Visual Studio Code, Weka

2.2 Team Structure

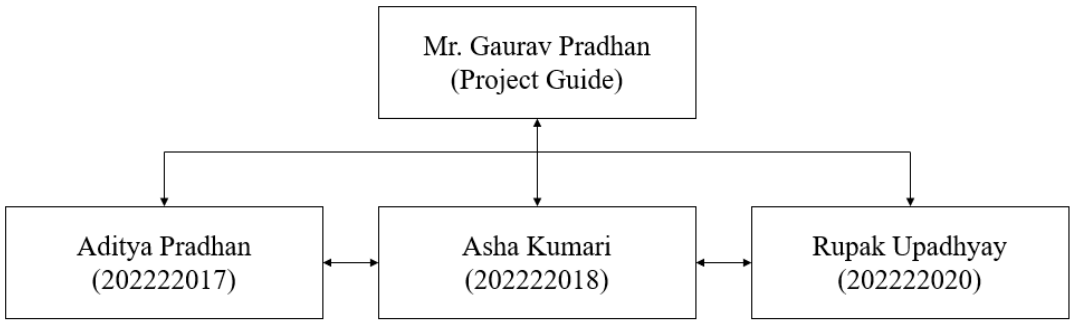

Fig 2.1 Team Structure

2.3 Gantt Chart

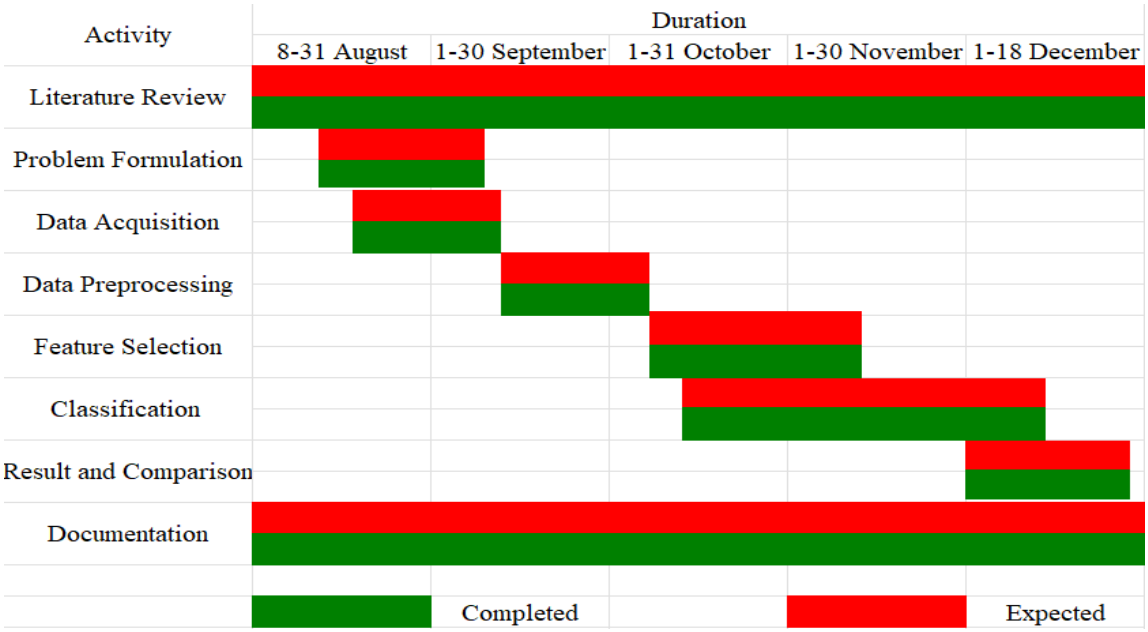

Fig 2.2 Gantt Chart

3. DESIGN STRATEGY FOR THE SOLUTION

3.1 Workflow-chart

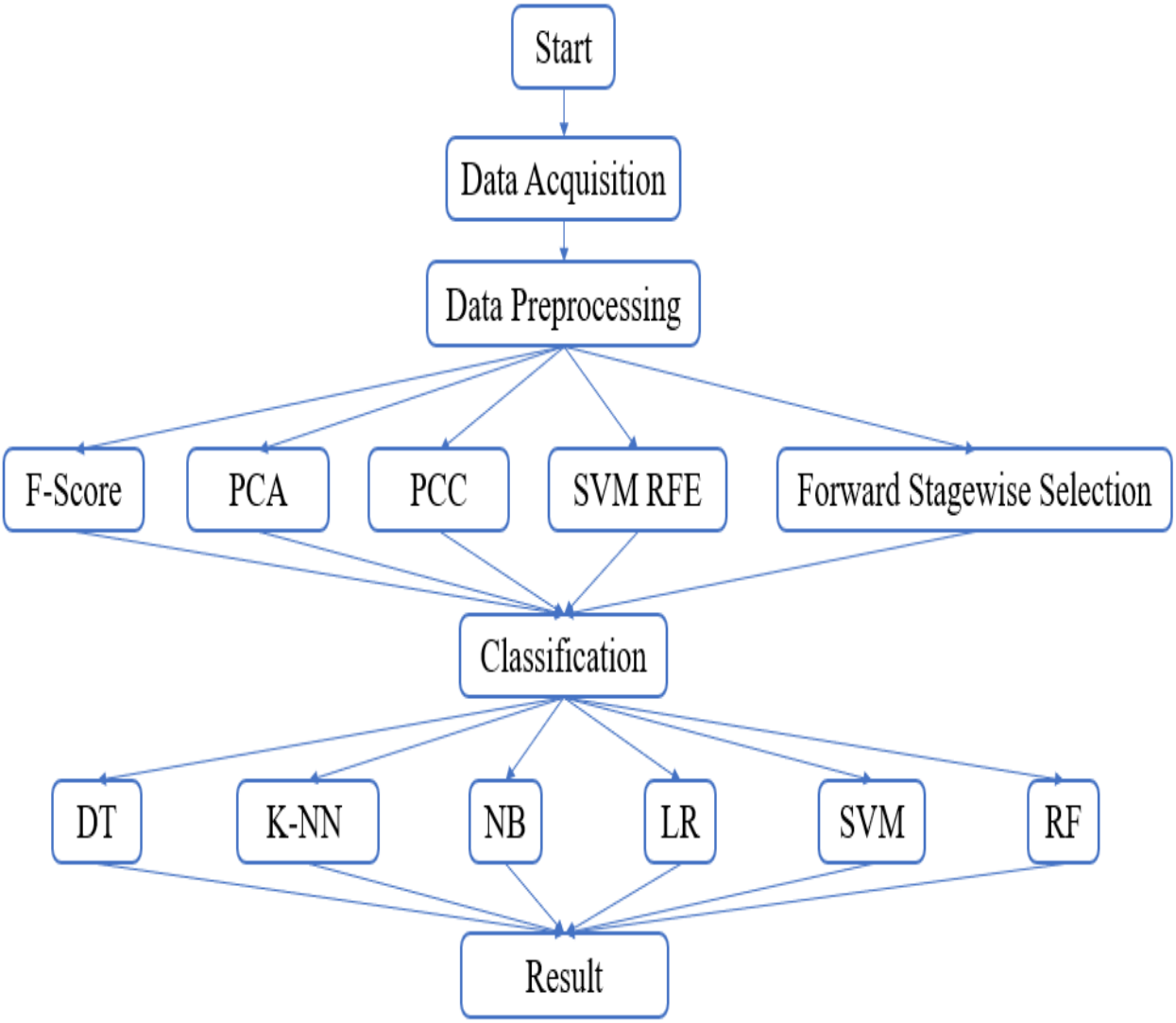

Fig 3.1 Workflow chart

4. IMPLEMENTATION

4.1 Data Acquisition

This study uses two datasets. One is the Pima Indians Diabetes Database[19], this dataset has medical records of 768 individuals. The other one was collected by requesting various researchers through emails and it has 2535 individual records, called Dataset2. The Pima Indians dataset contains eight features, and the other one has 19 features.

4.2 Data Preprocessing

The datasets were pre-processed using the weighted k-NN approach. It is a kNN algorithm that assigns different weights to the neighbors based on certain criteria. In the dataset we filled in the null values with the weighted averages of their neighboring values, here the weights are determined by the inverse of the distances between the data points.

The parameters of the KNNImputer to calculate the replacement for the null value:

KNNImputer(n_neighbors = 10, weights = ‘distance’)

This was done for both datasets.

The dataset2 contained many features with categorical data (‘yes’ or ‘no’, ‘true’ or ‘false’), these were replaced with numerical values 0 and 1. Like in the Sex feature, it had ‘male’ or ‘female’ as its value, while preprocessing the male was replaced with 0 and the female was replaced with 1.

4.3 Feature Selection

This study uses five different feature selection algorithms, namely F-score, Pearson Correlation Coefficient (PCC), Principal Component Analysis (PCA), Support Vector Machine Recursive Feature Elimination (SVM_RFE), and Forward Stagewise selection or boosting. The goal is to minimize the number of features using each algorithm and, in the end, choose the most recurring set of features for the classification.

4.3.1 F-Score:

F-score classifies data items into ‘positive’ or ‘negative’, ‘yes’ or ‘no’, a binary classification system. It is calculated by using the harmonic mean of the model’s precision and also to

$$F_1 = \frac{2}{\frac{1}{\text{recall}} \times \frac{1}{\text{precision}}} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$
$$= \frac{\text{tp}}{\text{tp} + \frac{1}{2}(\text{fp} + \text{fn})}$$

recall, and then it combines the precision and recall of the model. The chi-squared test returns 2 values: F-score and p-value. We will check the accuracy while considering different numbers of features for training at a time, based on the F-score for each feature. The features having higher F-scores are of more importance[20].

In this feature selection method, the dataset will be split into training and testing sets, and the ‘SelectKBest’ method is called from the script-learn library:

“Selector = SelectKBest(score_func = f_classif, k = 5)”

- score_func = f_classif: calculates the ANOVA (Analysis of Variance) F-value i.e. the F-score for each features available in the dataset.
- k = 5: Select top five features with the highest F-score calculated from the statistical test.

4.3.2 Principal Component Analysis:

It is a technique that is used to reduce the dimensions based on statistics, In order to produce a dataset with the necessary number of dimensions, the algorithm seeks to minimize the number of features in the dataset. It confirms that the maximal information from the original datasets is retained in the datasets with less dimensions by matching a higher-dimensional feature space to a lower-dimensional feature space[21]. The new features i.e., Principal Components are denoted as pci (where i = 1,2…n). The amount of information gathered from the original dataset is maximum in pc1 followed by pc2, then pc3, and so on.

The dataset is imported along with the required libraries, and then the dataset is standardized using ‘StandardScalar’ to ensure that they have zero mean and unit variance. Then the PCA method is called to get the required number of Principal components:

"PCA (n_components = 5)"

"X_pca = pca.fit_transform(X_scaled)"

- PCA: initializes a Principal Component Analysis(PCA) object from the scikit-learn library
- n_components: specifies the number of PCi to remain after reduction.
- pca.fit_tansform(X_scaled): computes the PCi and converts the original data into reduced-dimensional space.

4.3.3 Pearson Correlation Coefficient:

PCC helps to measures the statistical co-relations between two continuous variables. This gives the amount of correlation between the features present in the dataset. If a feature is highly correlated to another feature, then one of them can be considered a duplicate feature and can be dropped[22].

The architecture of the PCC follows:

correlation_matrix = df.corr()

corelation_with_target = correlation_matrix['tar_var'].abs()

sorted_features = correlation_with_target.sort_value(ascending = False)

sorted_features[1:6]

- corr(): calculates the correlation matrix for all the features.
- correlation_matrix['tar_var'].abs(): extracts the absolute values of the coefficients between each feature and the target variable ('tar_var').
- correlation_with_target.sort_value(ascending = False): sorting the features based on their absolute correlation with the target variable in descending order.
- sorted_features[1:6]: Select the top five features based on their correlation with their target variable.

4.3.4 Support Vector Machine Recursive Feature Elimination:

In order to accept sparse data and classify groups (make predictive rules) for data that cannot be classified by linear decision functions, SVM models are a potent tool for identifying predictive models or classifiers.[23].

SVM-RFE is a technique that uses a backward elimination process that selects the most relevant features in a dataset by iteratively training an SVM on a subset of the features and eliminating the least relevant feature in each iteration. This algorithm helps to measure the smallest change in the cost function by assuming that there is no change in the value of the discarded parameters in the optimization problem, which allows us to select the most relevant features where there is no need to retrain the classifier for every feature to be deleted[23].

For reducing features with this method, in this research, the datasets is split into two types i.e training and testing datasets, into the ratio of 4:1, and the following code is used:

"svm = SVC (kernel = 'linear')"

"rfe = RFE (estimator = svm, n_features_to_select = 5)"

"rfe.fit(X_train, y_train)"

- "SVC (kernel = 'linear')": is used to initialize an SVM classifier with a linear kernel.
- "RFE (estimator = svm, n_features_to_select = 5)": initializes the RFE model with SVM classifier and specifies the number of features to select.
- "rfe.fit (X_train, y_train)": fits the model to the training data to perform feature selection

4.3.5 Forward Stagewise Selection:

Also known as boosting is a bottom-up approach which helps in selection of features where the algorithm gradually builds up the model by selecting one feature at a time and training a supervised learner, sequentially one weight at a time [24].

This study uses boosting with XGBoost Classifier, the following is the architecture for this feature selection method:

"xgb_classifier = xgb.XGBClassifier(objective='binary:logistic', eval_metric='logloss', n_estimators=100, random_state=42)"

"X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)"

"sfs = SFS(xgb_classifier, k_features=5, forward=True, scoring='accuracy', cv=5, n_jobs=-1)"

"sfs.fit(X_train, y_train)"

- "xgb.XGBClassifier(objective='binary:logistic', eval_metric='logloss', n_estimators=100, random_state=42)": Initializes an XGBoost classifier with 100 rounds, the total number of decision trees in the ensemble.
- "train_test_split(X, y, test_size=0.2, random_state=42)": splits the dataset in the ratio of 4:1.
- "SFS(xgb_classifier, k_features=5, forward=True, scoring='accuracy', cv=5, n_jobs=-1)": initializes Sequential Feature Selector with XGBoost as the classifier, it selects five features in a forward manner, using accuracy as the scoring metric and five-fold cross-validation.
- "sfs.fit(X_train, y_train)": fits the sequential feature selector to train data and select the features.

4.4 Classification Algorithm

Classification is a supervised learning algorithm that involves putting things into a class/category according to particular characteristics so it's easier to make sense of them. This study uses six classification algorithms namely, k-nearest neighbor, Decision Tree, Logistic Regression, Naïve Bayes, Random Forest, and Support Vector Machine. This study uses Weka version 3.8.6 to run classification as it provides a GUI for ease of use.

4.4.1 Decision Tree:

The structure resembles a tree, with each core node representing a feature, branches signifying the rules, and leaf nodes representing the algorithm's outcome. It starts at the root node, from which the classification rules are produced by the path to its child nodes.[25]. This study made use of the well-known decision tree-based J48 algorithm. The study's J48 has the following parameters:

"weka.classifiers.trees.J48 -C 0.25 -M 2"

- ‘-C 0.25’: it the pruning confidence
- ‘-M 2’: is the minimum number of instance per leaf

4.4.2 K-Nearest Neighbour:

It is a supervised learning algorithm that is used for resolving the problem of classification as well as regression. This algorithm assumes that similar things remain close to each other i.e.,closer two items are more similar to each other. Initially, the k parameter is determined, as the number of neighbors for a given point. Used to calculates the distance of the new datasets that will be added further in the sample set, using the distance function. The class of the k neighbors is assigned according to the attribute value. Finally, the data is labeled[1]. This study uses the IBk an Instance-Based k-Nearest Neighbor from Weka. The algorithm used in this study has the following parameters:

“weka.classifiers.lazy.IBk -K 1 -W 0 -A”

- ‘-K 1’: set the number of neighbor
- ‘-W 0’: specifies unweighted voting
- ‘-A
- -R first-last \””: is used to configure the nearest neighbor search with a linear search algorithm and Euclidean distance as the distance metric. The ‘-R first-last’ option indicates that Euclidean distance should be applied to all attributes.

4.4.3 Logistic Regression:

It is a commonly used model when it comes to predicting health or illness[25]. It uses logistic sigmoid activation function to make predictions on the probability of the target category dependent variable. The resulted probability lies between 0 to 1. This study uses this algorithm with parameters:

“weka.classifiers.functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4”

- ‘-R 1.0E-8’: is used to specify the ridge parameter (regularization strength) for logistic regression.
- ‘-M -1’: is used to set the minimum number of instances per leaf. -1 indicates that no minimum is specified.

- ‘-num-decimal-places 4’: setting the number of decimal places for the output.

4.4.4 Naive Bayes:

For classification issues, this probabilistic classifier is employed. The Bayes Theorem serves as its foundation. [1]. This algorithm assumes that particular features don't affect the other features which means that the features are not dependent. This study the Weka classifier “weka.classifiers.bayes.NaiveBayes” with the default parameters.

4.4.5 Random Forest:

Random Forest is a collection of Decision Trees. While training, it builds several trees to randomly partition the datas. In this technique tree takes a classification as a vote for other trees, and the classification with the highest number of votes is chosen. [25]. This study uses the Forest classifier found with parameters set as:

> “weka.classifiers.trees.RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1”

- ‘-P 100’: the bag size percentage, percentage of the training instances to be used for building each tree in the forest.
- ‘-I 100: number of iterations.
- ‘-num-slots 1’: number of execution slots. Relevant when running in parallel but here, it's set to 1.
- ‘-K 0’: the number of attributes to randomly investigate. Setting it to 0 means that all attributes will be considered.
- ‘-M 1.0’: minimum number of instances per leaf.
- ‘-V 0.001’: variance reduction threshold for candidate splits.
- ‘-S 1’: the random seed.

4.4.6 Support Vector Machine:

It is suitable for small data collections with minimal outliers. Its main objective is to identify a hyperplane that can be used to connect data points[25] , also find the number of features that differently classify the data points. After that it finds a space with the maximum margin, which signifies the maximum distance between all classes of data points. The result of the

classification can be improved by using maximum margin distance[1]. This study uses the Sequential Minimal Optimization (SMO) implementation of SVM. The parameters are:

> “weka.classifiers.functions.SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007" -calibrator "weka.classifiers.functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4"”

- ‘-C 1.0’: the cost.
- ‘-L 0.001’: tolerance parameter for termination.
- ‘-P 1.0E-12’: the epsilon for round-off errors. Sets the precision for floating-point comparisions.
- ‘-N 0’: number of folds for reduced error pruning. 0 means that pruning is disabled.
- ‘-V -1’: verbosity level. -1 disables it.
- ‘-W 1’: use the transductive approach
- ‘-K "weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007"’: kernel to be used.
- ‘-calibrator "weka.classifiers.functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4"’: the calibrator used.

5. RESULT AND DISCUSSION

5.1 Result

If we don't select features, we may end up with a huge number of features that may be noisy, redundant, or irrelevant which leads to overfitting and poor performance of the classification model. Feature selection addresses this problem by selecting a subset of relevant features that are most informative for the learning task at hand[26]. The study team obtained the datasets and pre-processed them using weighted kNN from which the team obtained 10 datasets (5 for each raw dataset).

Serial No.	Features	Description
1	Pregnancies	Number of pregnancies
2	Glucose	Glucose tolerance test using plasma glucose concentration
3	BloodPressure	Diastolic blood pressure (mm Hg)
4	SkinThickness	Triceps skin fold thickness (mm)
5	Insulin	2-h serum insulin (mu U.ml)
6	BMI	Body mass index
7	DiabetesPedigreeFunction	Diabetes pedigree Function
8	Age	Age in years
9	Outcome	Class label (0 or 1)

Table 5.1 Description of PIMA Indian Diabetes Database[19]

Serial No.	Features	Description
1	Sex	The gender of the patient
2	Age	Age of the patient in years
3	History of high blood pressure	High blood pressure history of the patient (Yes or No)

4	History of use of drugs for high blood pressure	Use of drugs for high blood pressure (Yes or No)
5	Systolic blood pressure	Systolic blood pressure in mm of Hg
6	Diastolic blood pressure	Diastolic blood pressure in mm Hg
7	Height	Height of the patient in cm
8	Weight	Weight of the patient in kg
9	BMI	Body mass index
10	history of diabetes	History of diabetes
11	Family history of diabetes	Diabetes in the patient's family
12	History of aborted baby	Abortion performed on the patient
Serial No.	Features	Description
13	History of gestational diabetes	Gestational diabetes in the patient (Yes or No)
14	History of pregnancy	Pregnancy history of the patient (Yes or No)
15	FBS	Fasting Blood Sugar test (mg/dL)
16	Cholesterol	Cholesterol level measured in mg/dL
17	HDL	High-density lipoprotein measured in mg/dL
18	Triglyceride	Triglycerides level measured in mmol/L
19	result of high blood pressure screening	Class label (Positive or Negative or Old patient)
20	result of diabetes screening	Class label (Positive or Negative or Old patient)

Table 5.2 Description of Dataset2

5.1.1 Feature Selection:

5.1.1.1 F-score:

Pima Indians Dataset	Dataset2
• Glucose • SkinThickness • Insulin • BMI • Age	• Systolic blood pressure • Diastolic blood pressure • BMI • Family history of diabetes • History of aborted baby

Table 5.3 Features selected using F-score

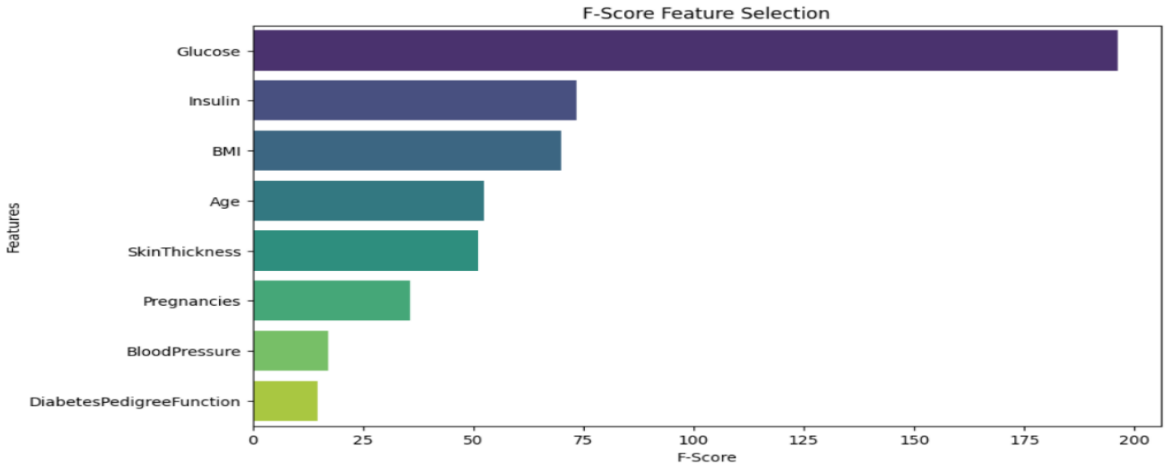

Fig 5.1 Bar Graph showing the f-score for the features in Pima Indians Dataset

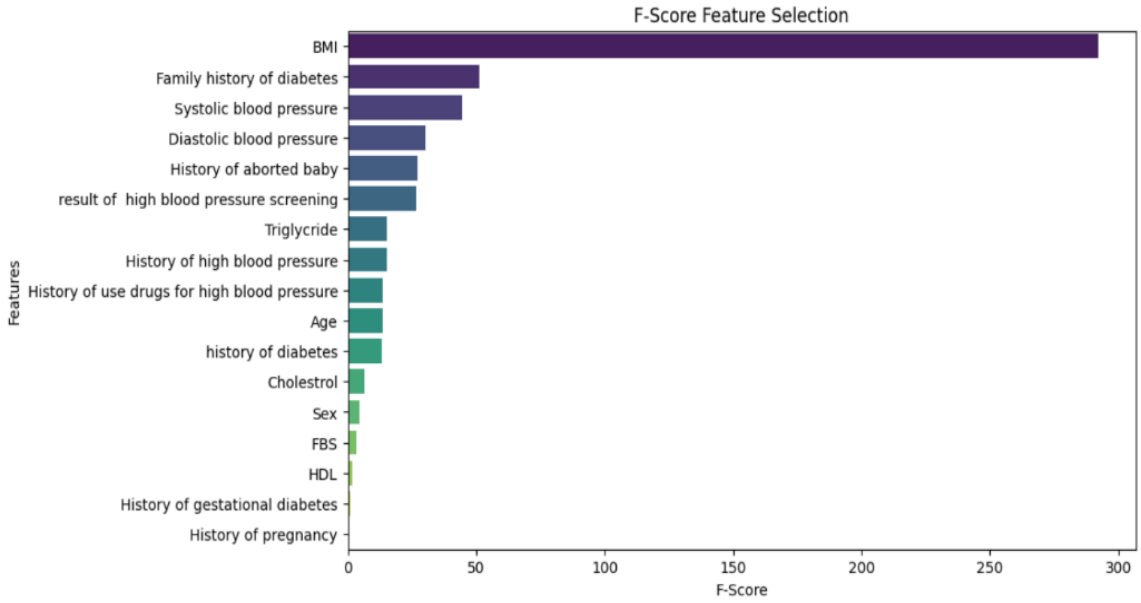

Fig 5.2 Bar Graph showing the f-score for the features in Dataset2

5.1.1.2 Principal Component Analysis:

This dimensional reduction technique maps the features into Principal Components namely PC1, PC2, PC3, PC4, and PC5 for each of the datasets.

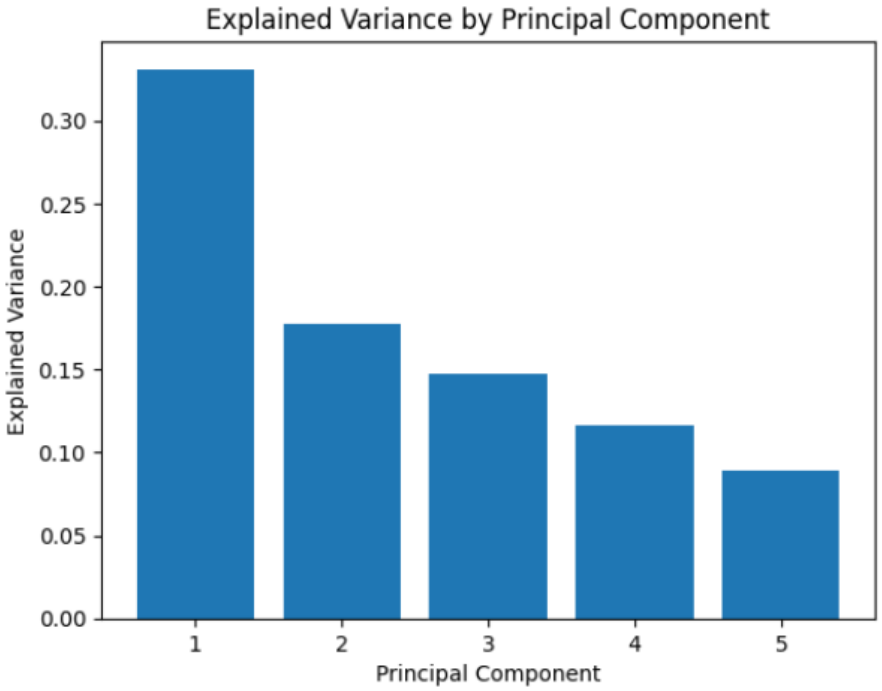

Fig 5.3 Bar graph depicting the amount of Explained Variance captured by various PCi from the Pima Indian Dataset

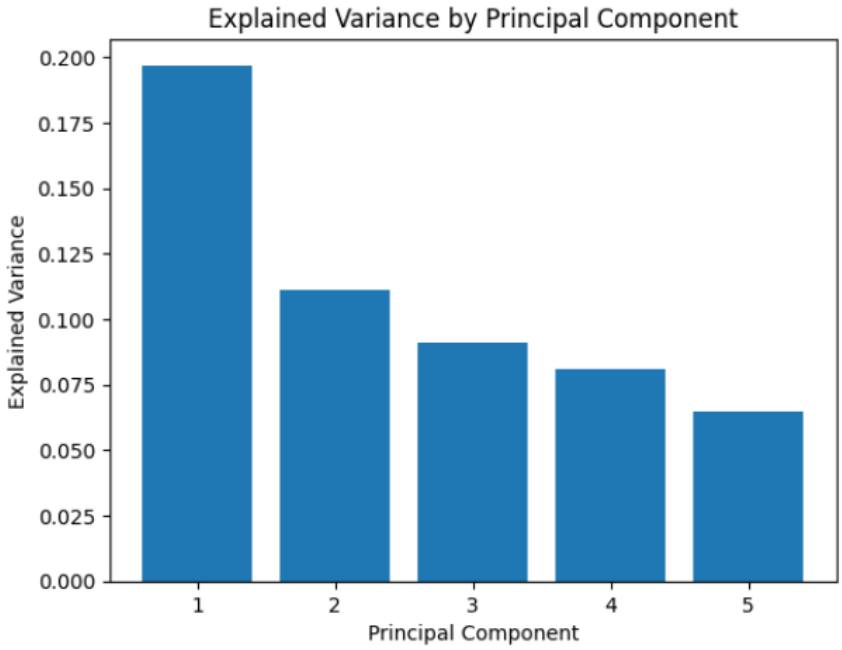

Fig 5.4 Bar graph depicting the amount of Explained Variance captured by various PCi from Dataset2.

5.1.1.3 Pearson Correlation Coefficient:

Pima Indians Dataset	Dataset2
• Glucose • SkinThickness • Insulin • BMI • Pregnancies	• Systolic blood pressure • Diastolic blood pressure • BMI • Family history of diabetes • result of high blood pressure screening

Table 5.4 Features selected using PCC

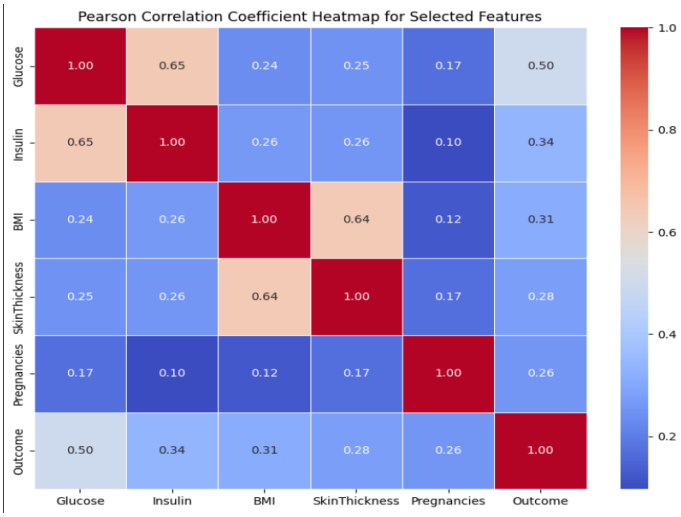

Fig 5.5 Heatmap of the correlation matrix of features from the Pima Indians Dataset

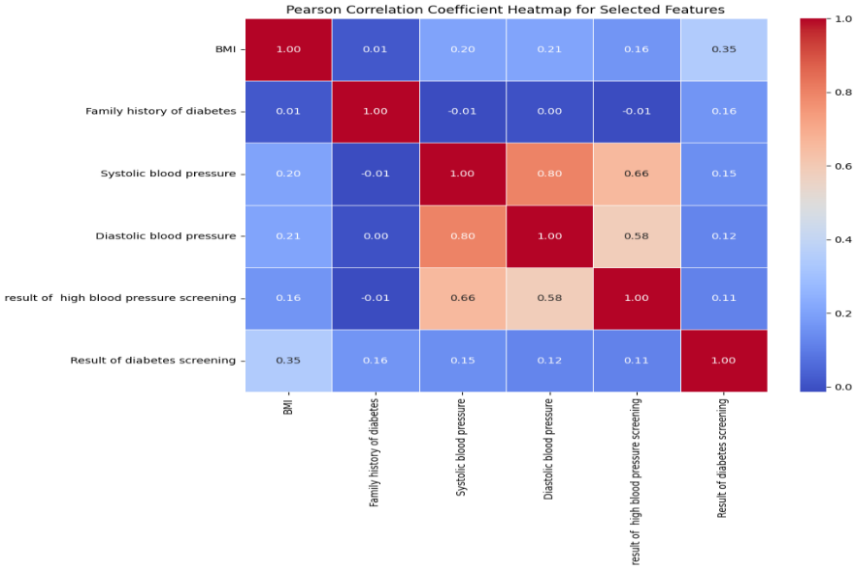

Fig 5.6 Heatmap of the correlation matrix of features from the Pima Indians Dataset

5.1.1.4 Support Vector Machine Recursive Feature Elimination:

Pima Indians Dataset	Dataset2
• Glucose • DiabetesPedigreeFunction • Age • BMI • Pregnancies	• History of diabetes • History of pregnancy • result of high blood pressure screening • Family history of diabetes • History of aborted baby

Table 5.5 Features selected using SVM-RFE

5.1.1.5 Forward Stagewise Selection:

Pima Indians Dataset	Dataset2
• Glucose • Insulin • Age • BMI • DiabetesPedigreeFunction	• Age • BMI • HDL • Family history of diabetes • History of aborted baby

Table 5.6 Features selected using Forward Stagewise Selection

5.1.2 Classification:

After the preprocessing and the selection of features, the classification of the data was performed, here every dataset the team got from the feature selection was classified. The PIMA Indians Dataset produced 5 datasets and it was classified using 6 algorithms namely, Decision Tree (DT), k-nearest neighbour (kNN), Logistic Regression (LR), Naïve Bayes (NB), Random Forest (RF), and Support Vector Machine (SVM), and the same goes for Dataset2.

Feature Selection / Classification	Forward Stagewise Selection	F-score	PCA	PCC	SVM_RFE
DT	65.1042	65.1042	65.1042	65.1042	65.1042
kNN	67.0573	64.4531	65.1042	65.8854	63.4115

LR	57.1615	64.9740	65.1042	65.4948	59.7656
NB	67.3177	69.5313	35.4167	69.2708	68.6198
RF	64.7135	65.8854	65.1042	66.6667	66.4063
SVM	65.2344	65.3646	65.1042	68.6198	64.8438

Table 5.7 Result of Classification algorithms for every Feature Selection used on PIMA Indians Dataset (Accuracy)

Feature Selection / Classification	Forward Stagewise Selection	F-score	PCA	PCC	SVM_RFE
DT	91.4004	91.4004	91.4004	91.4004	91.4004
kNN	91.4398	91.0848	91.3609	90.6114	91.4004
LR	88.9152	88.7968	91.4004	87.9684	91.4004
NB	91.5187	91.3215	8.6391	91.1637	91.4004
RF	90.6509	91.4793	91.4004	91.5976	91.4004
SVM	91.5582	91.5187	91.4004	91.4398	91.4004

Table 5.8 Result of Classification algorithms for every Feature Selection used on Dataset2 (Accuracy)

Support Vector Machine defined with cost 1 and using the PolyKernel gave the highest accuracy while classifying Dataset2 with features obtained from Forward Stagewise Selection feature selection method.

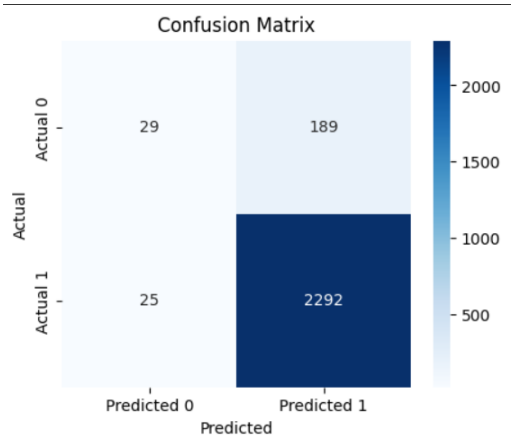

Fig 5.8 Confusion matrix of SVM

5.2 Discussion and Comparison

There are several similar research done for detecting diabetes early, the results of this study are compared with some of the recent literature. Table 8 depicts the comparative results between the studies.

Algorithm / Study	DT	kNN	LR	NB	RF	SVM
[25]	81.3000	-	83.8600	-	87.4000	86.0200
[27]	-	83.3300	-	-	-	-
[1]	-	97.3100	-	86.9200	96.9200	97.1200
[11]	94.4400	93.7900	-	79.8400	-	-
[9]	-	78.4000	-	92.7000	96.9000	90.8000
This study	91.4004	91.4398	91.4004	91.5187	91.4793	91.5582

Table 5.9 Comparison of Accuracy (%)

This study shows that the decision tree gives the most consistent result for all combinations of features 65.1042 and 91.4004 for PIMA Indians Dataset and Dataset2 respectively. The highest accuracy was obtained from SVM with features selected with Forward Stagewise Selection 91.5582.

6. SUMMARY AND CONCLUSION

6.1 Summary of Achievement

To date, this study has effectively outlined the challenges associated with existing machine learning models for detecting diabetes, wherein there is a trade-off between accuracy and complexity. This study also proposes a well-structured solution strategy that aligns with the identified challenges. While this study was in progress the study team reviewed many research papers to keep themselves up to date with the current trends and best practices in the field. A comprehensive methodology has been used that incorporates weighted kNN for data preprocessing, and five feature selection algorithms that select features based on their ranking. The datasets then underwent classification using six algorithms namely Decision Tree, k-NN, Logistic Regression, Naïve Bayes, Random Forest, and Support Vector Machine.

6.2 Difficulties encountered during the project

The collection of data is difficult as the patients might lie during their medical examination.

6.3 Limitation of the project

This study doesn't consider the case where a patient has lied about his medical record.

6.3 Future Scope

This study only considers a handful of feature selection methods and classification algorithms, further going on more classification algorithms can be used with their default parameter to see if they give more accurate result or not.

6.4 Conclusion

This study aimed to address the challenges of obtaining high accuracy in detecting diabetes while keeping the complexity of the machine learning model low. The study also proposed a solution to the challenge i.e., is to utilize five feature selection algorithms and six classification algorithms to detect Type 2 diabetes. The feasibility study showed that this study was economically, technically, operationally, and schedule-wise feasible. Two datasets were collected, namely Pima Indians Diabetes Database[19] and Datase2 which were pre-processed using a weighted kNN approach. The five feature selection algorithms used were F-score, Principal Component Analysis (PCA), Pearson Correlation Coefficient (PCC), Support Vector Machine Recursive Feature Elimination (SVM_RFE), and Forward Stagewise selection. The features that were obtained showed that different algorithms selected different

sets of features. The diverse feature selection algorithms contribute to the overall robustness of the study, offering insights into the relevance of different features in diabetes detection. This paper was able to achieve a high accuracy of 91.5582 using the Support Vector Machine with features obtained from the forward stagewise selection. Therefore, the developed Support Vector Machine gets the highest accuracy using 10-fold cross-validation, whilst keeping the model as simple as possible, as all the models in this study were used with their default parameters, to keep the complexity of the models low.

REFERENCES

[1] L. Chaves and G. Marques, “Data mining techniques for early diagnosis of diabetes: A comparative study,” *Appl. Sci.*, vol. 11, no. 5, pp. 1–12, Mar. 2021, doi: 10.3390/app11052218.

[2] B. F. Wee, S. Sivakumar, K. H. Lim, W. K. Wong, and F. H. Juwono, “Diabetes detection based on machine learning and deep learning approaches,” *Multimed. Tools Appl.*, 2023, doi: 10.1007/s11042-023-16407-5.

[3] “Diabetes.” Accessed: Nov. 17, 2023. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/diabetes

[4] “Diabetes Facets and Figures | International Diabetes Federation.” Accessed: Nov. 17, 2023. [Online]. Available: https://idf.org/about-diabetes/diabetes-facts-figures/

[5] N. Jothi, N. A. Rashid, and W. Husain, “Data Mining in Healthcare - A Review,” *Procedia Comput. Sci.*, vol. 72, pp. 306–313, 2015, doi: 10.1016/j.procs.2015.12.145.

[6] T. Hendrickx, B. Cule, P. Meysman, S. Naulaerts, K. Laukens, and B. Goethals, “Mining association rules in graphs based on frequent cohesive itemsets,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9078, no. 3, pp. 637–648, 2015, doi: 10.1007/978-3-319-18032-8_50.

[7] S. Saru, “ANALYSIS AND PREDICTION OF DIABETES USING MACHINE LEARNING,” 2019. [Online]. Available: https://ssrn.com/abstract=3368308

[8] U. Ahmed *et al.*, “Prediction of Diabetes Empowered With Fused Machine Learning,” *IEEE Access*, vol. 10, pp. 8529–8538, 2022, doi: 10.1109/ACCESS.2022.3142097.

[9] D. D. Rufo, T. G. Debelee, A. Ibenthal, and W. G. Negera, “Diagnosis of diabetes mellitus using gradient boosting machine (Lightgbm),” *Diagnostics*, vol. 11, no. 9, Sep. 2021, doi: 10.3390/diagnostics11091714.

[10] R. Birjais, A. K. Mourya, R. Chauhan, and H. Kaur, “Prediction and diagnosis of future diabetes risk: a machine learning approach,” *SN Appl. Sci.*, vol. 1, no. 9, Sep. 2019, doi: 10.1007/s42452-019-1117-9.

[11] F. Kazerouni, A. Bayani, F. Asadi, L. Saeidi, N. Parvizi, and Z. Mansoori, “Type2 diabetes mellitus prediction using data mining algorithms based on the long-noncoding

RNAs expression: A comparison of four data mining approaches," *BMC Bioinformatics*, vol. 21, no. 1, Aug. 2020, doi: 10.1186/s12859-020-03719-8.

[12] H. Lu, S. Uddin, F. Hajati, M. A. Moni, and M. Khushi, "A patient network-based machine learning model for disease prediction: The case of type 2 diabetes mellitus," *Appl. Intell.*, vol. 52, no. 3, pp. 2411–2422, Feb. 2022, doi: 10.1007/s10489-021-02533-w.

[13] N. Sneha and T. Gangil, "Analysis of diabetes mellitus for early prediction using optimal features selection," *J. Big Data*, vol. 6, no. 1, Dec. 2019, doi: 10.1186/s40537-019-0175-6.

[14] R. B. Lukmanto, Suharjito, A. Nugroho, and H. Akbar, "Early detection of diabetes mellitus using feature selection and fuzzy support vector machine," in *Procedia Computer Science*, Elsevier B.V., 2019, pp. 46–54. doi: 10.1016/j.procs.2019.08.140.

[15] L. Kopitar, P. Kocbek, L. Cilar, A. Sheikh, and G. Stiglic, "Early detection of type 2 diabetes mellitus using machine learning-based prediction models," *Sci. Rep.*, vol. 10, no. 1, Dec. 2020, doi: 10.1038/s41598-020-68771-z.

[16] V. Chang, J. Bailey, Q. A. Xu, and Z. Sun, "Pima Indians diabetes mellitus classification based on machine learning (ML) algorithms," *Neural Comput. Appl.*, vol. 35, no. 22, pp. 16157–16173, Aug. 2023, doi: 10.1007/s00521-022-07049-z.

[17] R. Ahuja, S. C. Sharma, and M. Ali, "A diabetic disease prediction model based on classification algorithms," *Ann. Emerg. Technol. Comput.*, vol. 3, no. 3, pp. 44–52, Jul. 2019, doi: 10.33166/AETiC.2019.03.005.

[18] A. Iyer, J. S, and R. Sumbaly, "Diagnosis of Diabetes Using Classification Mining Techniques," *Int. J. Data Min. Knowl. Manag. Process*, vol. 5, no. 1, pp. 01–14, Jan. 2015, doi: 10.5121/ijdkp.2015.5101.

[19] "Pima Indians Diabetes Database." Accessed: Nov. 18, 2023. [Online]. Available: https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

[20] "Feature Selection - F-score | Kaggle." Accessed: Nov. 18, 2023. [Online]. Available: https://www.kaggle.com/code/tanmayunhale/feature-selection-f-score

[21] "Reduce Data Dimensionality using PCA - Python - GeeksforGeeks." Accessed: Nov. 19, 2023. [Online]. Available: https://www.geeksforgeeks.org/reduce-data-

dimentionality-using-pca-python/

[22] “Feature Selection - Pearson Correlation | Kaggle.” Accessed: Nov. 19, 2023. [Online]. Available: https://www.kaggle.com/code/tanmayunhale/feature-selection-pearson-correlation

[23] H. Sanz, C. Valim, E. Vegas, J. M. Oller, and F. Reverter, “SVM-RFE: Selection and visualization of the most relevant features through non-linear kernels,” *BMC Bioinformatics*, vol. 19, no. 1, pp. 1–18, 2018, doi: 10.1186/s12859-018-2451-4.

[24] “9.6 Feature Selection via Boosting.” Accessed: Nov. 19, 2023. [Online]. Available: https://jermwatt.github.io/machine_learning_refined/notes/9_Feature_engineer_select/9_6_Boosting.html

[25] M. J. Uddin *et al.*, “A Comparison of Machine Learning Techniques for the Detection of Type-2 Diabetes Mellitus: Experiences from Bangladesh,” *Inf.*, vol. 14, no. 7, Jul. 2023, doi: 10.3390/info14070376.

[26] J. Li *et al.*, “Feature selection: A data perspective,” *ACM Comput. Surv.*, vol. 50, no. 6, 2017, doi: 10.1145/3136625.

[27] A. Anggrawan and M. Mayadi, “Application of KNN Machine Learning and Fuzzy C-Means to Diagnose Diabetes,” *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 22, no. 2, pp. 405–418, 2023, doi: 10.30812/matrik.v22i2.2777.