|  |
| --- |
|  |
|  |
| |  |  |  | | --- | --- | --- | |  |  |  | |

**Group 10**

**Refining Prediction of Disease using machine learning**

Contents

[1 Introduction 2](#_Toc121049670)

[2 What’s the Data? 2](#_Toc121049671)

[3 Methods 5](#_Toc121049672)

[3.1 Importing Libraries, Packages, and Dataset: 5](#_Toc121049673)

[3.2 Exploratory Analysis of the Data: 5](#_Toc121049674)

[3.3 Checking for Outliers: 5](#_Toc121049675)

[3.4 Data Preprocessing: 6](#_Toc121049676)

[3.5 Evaluation Metrics: 6](#_Toc121049677)

[4 Results 8](#_Toc121049678)

[4.1 Logistic Regression 8](#_Toc121049679)

[4.2 Random Forest 9](#_Toc121049680)

[4.3 XGBoost 9](#_Toc121049681)

[4.4 SVM 10](#_Toc121049682)

[4.5 KNN 10](#_Toc121049683)

[4.6 Comparing accuracies of all the models 11](#_Toc121049684)

[4.7 Comparing the F1, precision, accuracy and recall scores of the best two performing models 11](#_Toc121049685)

[4.8 Comparing the auc scores of all the models 12](#_Toc121049686)

[5 Discussion 12](#_Toc121049687)

[5.1 Logistic Regression 12](#_Toc121049688)

[5.2 Random Forest 13](#_Toc121049689)

[5.3 XGBoost 13](#_Toc121049690)

[5.4 SVM 14](#_Toc121049691)

[5.5 KNN 14](#_Toc121049692)

[6 Conclusion 15](#_Toc121049693)

[7 Contributions 15](#_Toc121049694)

[8 References 17](#_Toc121049695)

[9 Appendices 18](#_Toc121049696)

# Introduction

Nowadays, People are waiting an hour or more in some cases just to be triaged, assessed by a nurse, and assigned a department based on the form they fill. The triage time is high because they follow a traditional process which have not been updated since long time. The traditional process is as follows:

* A patient enters a hospital
* Then goes to front desk and fill on an average 4-5 forms when combined contains 100-120 Question. (This process on an average will take about 20 – 25 minutes)

**Available Solution in market:** Automation solution are already available in market which works same as traditional model. (i.e They also asks 100-120 Question on computer and answering them requires time (15-20 minutes) and effort which is the last things any patient wants they can provide solution faster than traditional pen and paper method)

**Our Solution:** We have built multiple algorithm and tune accuracy and selected feature based on their importance**.** Our model is built in such a way that it requires patient to answer only 20 – 25 questions and gives same accuracy.

# What’s the Data?

We took data from the hospital located in the USA from Kaggle data. In the dataset we have 40 odd prognosis different feature with around 100 to 120 different samples of diseases with 5000 rows.

**Data engineering:**

In data Engineering we import seasons by dates categorization from the data sets for batter visualizations and it helps us to visualize our data in a proper way.

**Data cleaning:**

Data cleaning is a process which is used to clean the data into usable and understandable vision.

Only pertinent information is kept from the cleaned data, which may then be used to feed machine learning models for additional processing.

**Dropping Null Values:**

The first step in data cleaning was to count the null values and missing values for each column in our dataset. according to the results of applying the method. When we used isna().sum(), we discovered that no column contained any null values.

There were no anomalies found in the preprocessed dataset, and everything appeared to be in order. This indicates that our dataset is ready to be sent into the machine learning models for additional processing.

**Machine learning models:**

The study of algorithms that get better over time is called machine learning. Building a model that can process sample data, sometimes referred to as train data, is utilized in the artificial intelligence area to train the model to make more precise predictions for test data in the future. For this dataset, we utilized 5 machine learning models. The models are,

**1. Decision Tree.**

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving ***regression and classification problems*** too.

The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by***learning simple decision rules*** inferred from prior data (training data).

In Decision Trees, for predicting a class label for a record we start from the **root** of the tree. We compare the values of the root attribute with the record’s attribute. Based on comparison, we follow the branch corresponding to that value and jump to the next node.

**2. Random forest.**

Random forest is a ***Supervised Machine Learning Algorithm*** that is ***used widely in Classification and Regression problems***. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

One of the most important features of the Random Forest Algorithm is that it can handle the data set containing ***continuous variables*** as in the case of regression and ***categorical variables*** as in the case of classification. It performs better results for classification problems.

**3. Gradient boosting.**

Gradient boosting is a type of machine learning boosting. It relies on the intuition that the best possible next model, when combined with previous models, minimizes the overall prediction error. The key idea is to set the target outcomes for this next model in order to minimize the error. How are the targets calculated? The target outcome for each case in the data depends on how much changing that case's prediction impacts the overall prediction error:

* If a small change in the prediction for a case causes a large drop in error, then next target outcome of the case is a high value. Predictions from the new model that are close to its targets will reduce the error.
* If a small change in the prediction for a case causes no change in error, then next target outcome of the case is zero. Changing this prediction does not decrease the error.

The name *gradient boosting*arises because target outcomes for each case are set based on the gradient of the error with respect to the prediction. Each new model takes a step in the direction that minimizes prediction error, in the space of possible predictions for each training case.

**4. Logistic Regression (LR)**

**Logistic regression** is a statistical model that Is used to determine the probability that an event will happen. It shows the relationship between features, and then calculates the probability of a certain outcome.

Logistic regression is used in ML to help create accurate predictions. It is like linear regression, except rather than a graphical outcome, the target variable is binary; the value is either 1, or 0.

There are two types of measurables, the explanatory variables/ features (item being measured) and the response variable/ target binary variable, which is the outcome.

For example, when trying to predict whether a student will pass or fail a test, the hours studied are the feature, and the response variable will have two values - pass or fail.

There are three basic kinds of logistic regression:

1. **Binary logistic regression:** Here there are only two possible outcomes for the categorical response. As in the example above – a student passes or fails.
2. **Multinomial logistic regression:** This is where the response variables can include three or more variables, which will not be in any order. An example is predicting whether diners at a restaurant prefer a certain kind of food – vegetarian, meat or vegan.
3. **Ordinal logistic regression:** Like multinomial regression, there can be three or more variables. However, there is an order the measurements follow. An example is rating a hotel on a scale of 1 to 5.

**5. Fully connected Deep learning**

* These are the models that we used in our datasets. We used these models for our dataset. Some of our models differ from the models that were used in the work of the other group, as was previously indicated in the section on related work.

**6. PyCaret**

* PyCaret is an open-source, low-code machine learning library in Python that automates machine learning workflows. It is an end-to-end machine learning and model management tool that exponentially speeds up the experiment cycle and makes you more productive.

# Methods

In this section we will provide a brief information about the steps or approaches taken, exploratory data analysis, the preprocessing techniques, the machine learning models, and evaluation metrics we are intending to use for our dataset.

## Importing Libraries, Packages, and Dataset:

We imported the required libraries and packages which is essential to run a specific section or all the sections of the code in the python notebook (ipynb file). Then we had imported our dataset and stored it in a data frame for further processing.

## Exploratory Analysis of the Data:

Exploratory Data Analysis refers to the critical process of performing initial investigations on data to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations. It is a good practice to understand the data first and try to gather as many insights from it. EDA is all about making sense of data in hand, before getting them dirty with it.

We performed Exploratory data analysis using various technologies like Excel, Tableau and Python. Use of various technologies made it easier for our team to better get the better understanding of the data.

Using python we analyze the data using various functions like “ .head()”function of pandas library which returns first five observations of the data set. Similarly “.tail()” returns last five observations of the data set. We found the total number of rows and columns in the data set using “. shape”. The describe () function in pandas is very handy in getting various summary statistics. This function returns the count, mean, standard deviation, minimum and maximum values and the quantiles of the data.

Furthermore, we analyzed the distribution of Numerical and Categorical Features within our dataset to understand the features visually. Then we created a heat map for correlation matrix to summarize the data and check which features were strongly correlated and weakly correlated.

## Checking for Outliers:

An *outlier* is an observation that lies an abnormal distance from other values in a random sample from a population. In a sense, this definition leaves it up to the analyst (or a consensus process) to decide what will be considered abnormal. Before abnormal observations can be singled out, it is necessary to characterize normal observations. So we checked the dataset for outlier and if found decided how to handle it.

## Data Preprocessing:

Data preprocessing, a component of data Preparation, describes any type of processing performed on Raw Data to prepare it for another data processing procedure. It has traditionally been an important preliminary step for the Data Mining process. More recently, data preprocessing techniques have been adapted for training machine learning models and AI models and for running inferences against them.

Data preprocessing transforms the data into a format that is more easily and effectively processed in data mining, machine learning and other data science tasks. The techniques are generally used at the earliest stages of Machine Learning model.

The raw data might be incomplete and might cause errors later, which is why it is always a good preprocess the data. Data is cleaned so that only important information is retained and the same can be fed into the machine learning models for further processing

Data Cleaning:

1. **Dropping Null Values:**

We started preparing the data by checking the number of null values or missing values for each column in our dataset. Based on the output from using the method. isna **(). sum()**, we found that there was no null values in any of the column.

Data Transformation:

Data transformation is the process of converting data from one format to another, typically from the format of a source system into the required format of a destination system.

In our case, one thing which we did was that our dataset contained the data for one year so for Machine Learning algorithm to perform better we broke the one-year data into brackets of seasons which will make for ml to perform better because instead of working with more than 4000 odd date it must predict only 4 Season i.e. (Winter, Autumn, Spring & Summer).

## Evaluation Metrics:

Evaluation metrics are **used to measure the quality of the statistical or machine learning model**. Evaluating machine learning models or algorithms is essential for any project. It is essential to use these metrics as they offer a better understanding of how every model is performing to its peak capabilities and what can be improved to increase the scores later. There are many different types of evaluation metrics available to test a model. The different types of metrics are as follows:

* **F1 Score**
* **Precision**
* **Recall**
* **Accuracy**
* **Confusion Matrix**

**F1 Score:**

The F1 score is a performance metric for classification and is calculated as the harmonic mean of precision and recall: The F1 score is commonly used to measure performance of binary classification, but extensions to multi-class classifications exist.

**Precision:**

The precision is the ratio tp / (tp + fp) where tp is the number of true positives and fp the number of false positives. The precision is intuitively the ability of the classifier not to label as positive a sample that is negative. The best value is 1 and the worst value is 0

**Recall:**

The recall is calculated as the ratio between the numbers of Positive samples correctly classified as Positive to the total number of Positive samples. The recall measures the model's ability to detect positive samples. The higher the recall, the more positive samples detected.

**Accuracy:**

Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right. Formally, accuracy has the following definition: Accuracy = Number of correct predictions Total number of predictions.

**Confusion Matrix:**

A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix.

(After all the work done, we printed the accuracy scores of every model for train and test data. Then, we made plots to compare the accuracies of different models and we also printed the evaluation metrics scores, graphs, and compared them together for further analysis. All the results are mentioned in the Results section.)--Remove

# Results

|  |  |
| --- | --- |
| **Machine Learning Model** | **Accuracy Achieved** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

From the above table, we can see that XGBoost has achieved the highest accuracy on train and test data. Random Forest classifier is the second-best machine learning model with 79.17% accuracy on test data.

Following are the charts, of train and test accuracies of the all the machine learning models:

## Logistic Regression



## Random Forest



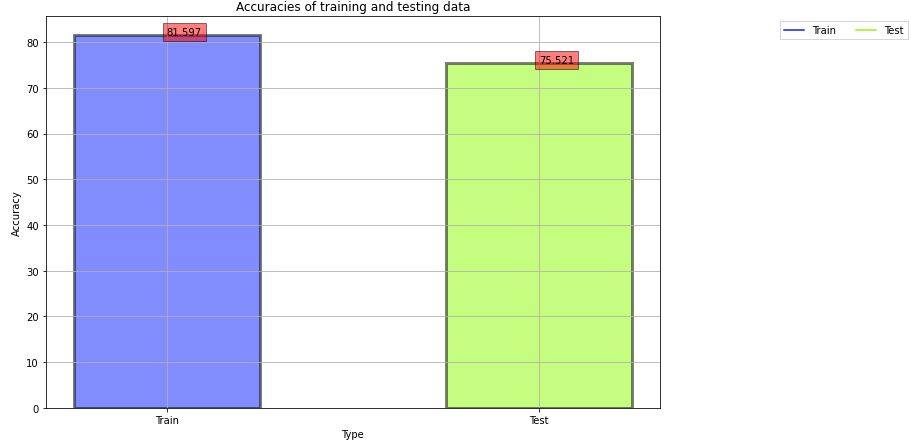
## XGBoost



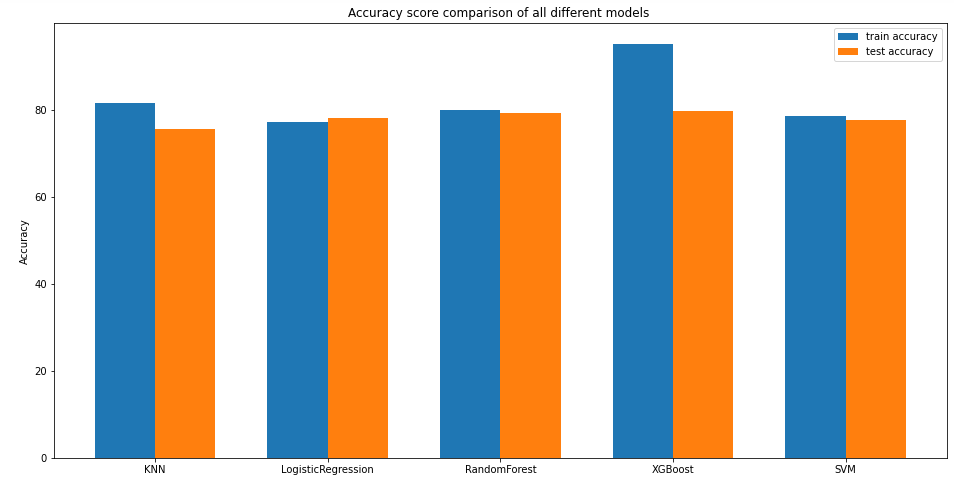
## SVM



## KNN

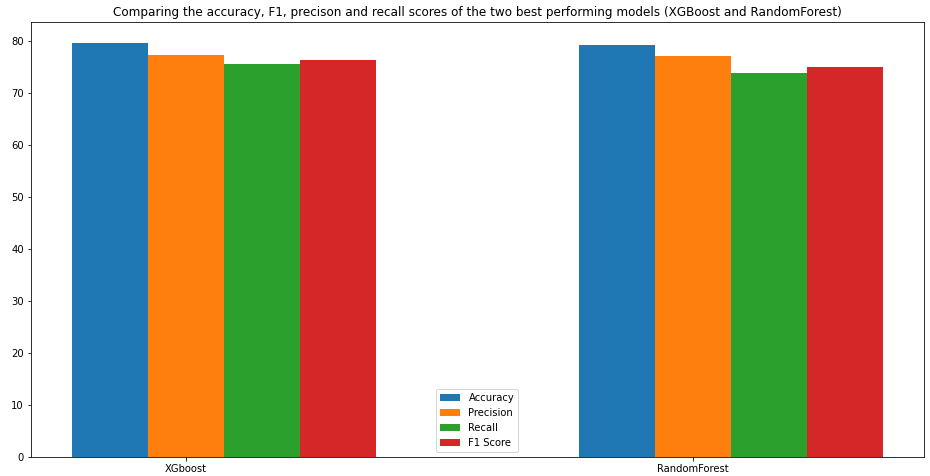


## Comparing accuracies of all the models



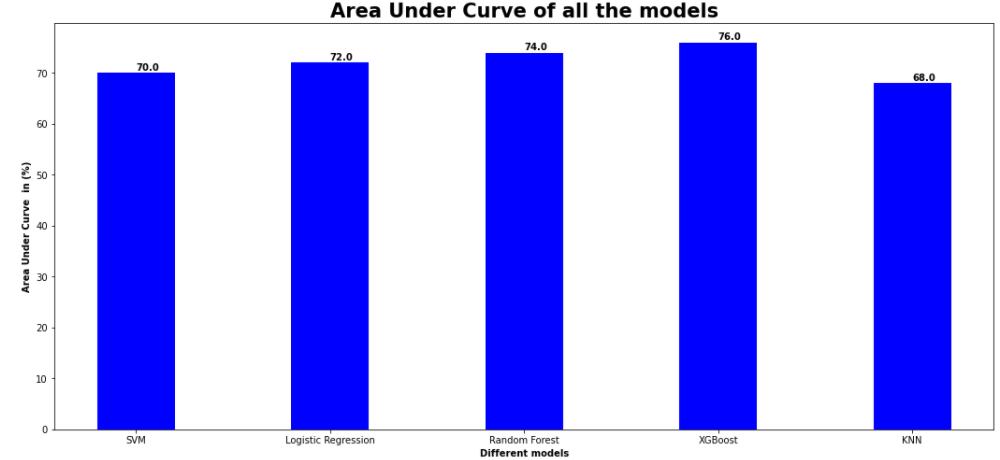
**Interpretation:** From the above chart, we can see that XGBoost has yielded the highest train and test accuracies followed by random forest which has the second highest test accuracy.

## Comparing the F1, precision, accuracy and recall scores of the best two performing models



**Interpretation:** From the above chart, we can see the comparison of precision, recall, accuracy and F1 scores Of XGBoost and Random Forest. We can see that the scores are similar.

## Comparing the auc scores of all the models



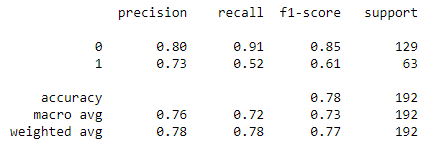
**Interpretation:** Higher the AUC score better is the model at predicting class 0 as 0 and class 1 as 1. From the chart, we can see that XGBoost has the highest auc score of 76%. This means that XGBoost is good at predicting classes accurately.

# Discussion

We performed hyperparameter tuning for all the machine learning models to find the best parameters and achieve the highest accuracies using gridsearch.

Following are the precision and recall scores from classification report and its interpretation of each of the machine learning models:

## Logistic Regression



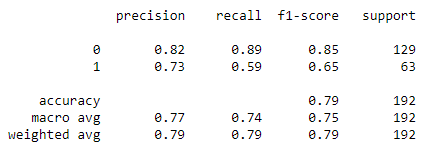
The accuracies yielded by **logistic regression** are train accuracy of 77.08% and

test accuracy of 78.12%. To evaluate the performance of the model we created confusion matrix and from it we can see that the model is classifying TP and TN with high accuracy.

**From the above classification report for logistic regression we can infer that** the model’s precision is high because the model is predicting 80% of 0 class correctly and 73% of class 1 accurately.

The recall score us 91% for class 0 and 52% for class 1 which means that the model has correctly labeled 91% of class 0 out of all the class 0 observations. Similarly, the model has only correctly labeled 52% of class 1 labels out of all the class 1 labels. Generally, a recall score of above 50% is considered good, which means that the model is good at correctly labeling classes.

## Random Forest



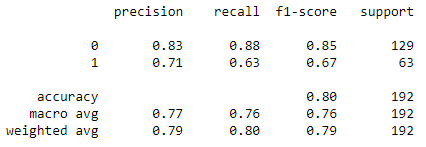
The accuracies yielded by **Random Forest** are train accuracy of 79.86% and

test accuracy of 79.17%. To evaluate the performance of the model we created confusion matrix and from it we can see that the model is classifying TP and TN with high accuracy.

**From the above classification report for Random Forest we can infer that** the model’s precision is high because the model is predicting 82% of 0 class correctly and 73% of class 1 accurately.

The recall score us 89% for class 0 and 59% for class 1 which means that the model has correctly labeled 89% of class 0 out of all the class 0 observations. Similarly, the model has only correctly labeled 59% of class 1 labels out of all the class 1 labels. Generally, a recall score of above 50% is considered good, which means that the model is good at correctly labeling classes.

## XGBoost



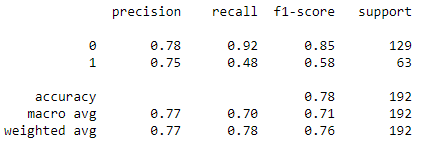
The accuracies yielded by **XGBoost** are train accuracy of 91.54% and

test accuracy of 79.69%. To evaluate the performance of the model we created confusion matrix and from it we can see that the model is classifying TP and TN with high accuracy.

**From the above classification report for XGBoost** **we can infer that** the model’s precision is high because the model is predicting 83% of 0 class correctly and 71% of class 1 accurately.

The recall score us 88% for class 0 and 63% for class 1 which means that the model has correctly labeled 88% of class 0 out of all the class 0 observations. Similarly, the model has only correctly labeled 63% of class 1 labels out of all the class 1 labels. Generally, a recall score of above 50% is considered good, which means that the model is good at correctly labeling classes. **The recall and precision scores are similar to random forest but slightly better which makes XGBoost our top choice for PIMA Indians Diabetes data.**

## SVM



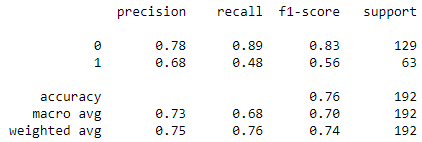
The accuracies yielded by **SVM** are train accuracy of 78.47% and

test accuracy of 77.60%. To evaluate the performance of the model we created confusion matrix and from it we can see that the model is classifying TP and TN with high accuracy.

**From the above classification report for SVM** **we can infer that** the model’s precision is high because the model is predicting 78% of 0 class correctly and 75% of class 1 accurately.

The recall score us 92% for class 0 and 48% for class 1 which means that the model has correctly labeled 92% of class 0 out of all the class 0 observations. Similarly, the model has only correctly labeled 48% of class 1 labels out of all the class 1 labels. Generally, a recall score of above 50% is considered good, which means that **SVM is not very good at correctly labeling classes as it on has 48% accuracy at correctly labeling class 1 labels.**

## KNN



The accuracies yielded by **KNN** are train accuracy of 81.60% and

test accuracy of 75.52%. To evaluate the performance of the model we created confusion matrix and from it we can see that the model is classifying TP and TN with high accuracy.

**From the above classification report for KNN** **we can infer that** the model’s precision is high because the model is predicting 78% of 0 class correctly and 68% of class 1 accurately.

The recall score us 89% for class 0 and 48% for class 1 which means that the model has correctly labeled 89% of class 0 out of all the class 0 observations. Similarly, the model has only correctly labeled 48% of class 1 labels out of all the class 1 labels. Generally, a recall score of above 50% is considered good, which means that **KNN is not very good at correctly labeling classes as it on has 48% accuracy at correctly labeling class 1 labels.**

**In conclusion, when choosing a model precision, recall and F1 scores give a better picture of model’s actual performance over train and test accuracies.**

**From the above discussion, we can see that XGBoost and Random Forest have the best precision, recall and F1 scores but XGBoost has slightly better scores which makes it our top pick and a good fit for predicting if a person has diabetes or not based on several different factors.**

# Conclusion

Close to 400 million people worldwide have diabetes. It is one of the leading causes of death is type 2 diabetes. Our analysis attempts to accurately predict if a patient has diabetes based on several different features. Our analysis can be used to accurately predict if a patient has diabetes or not. Accuracy is the highest priority in healthcare.

Based on the outcome of the implementation of various Machine Learning models namely XGBoost, Random Forest, KNN, Logistic Regression and Support Vector Machine and from their individual F1 scores, it is very evident that the two best performing algorithms turn out to be XGBoost and Random Forest.

F1 score is calculated using the weighted average of Precision and Recall where Precision is the ratio correctly predicting positive observations of all the positive predicted positive observations and Recall is the ratio of correctly predicted positive observations of all the observations of the true class.

F1 score is more useful than the accuracy score, especially when we the distribution of the class to be uneven, and hence it is better if there is a requirement to balance between Precision and Recall. We will now be able to perform accurate prediction if a person has diabetes or not by using the 2 machine learning models name XGBoost and Random Forest.

**We concluded that XGBoost, is the best model because of it has the highest test accuracy, model, precision and recall scores.**

# Contributions

|  |  |
| --- | --- |
| **Name** | **Contribution** |
| **Aditya Danturthi (0756019)** | **Project work**   * Performed data exploration. * Used gridsearch to find the best parameters for XGBoost and use them to improve the overall accuracy of the model. * Applied **XGBoost** algorithm which yielded 95.14% train accuracy and 79.69% test accuracy. * Performed evaluation metrics on XGBoost by creating a confusion matrix and a classification report and then calculated the AUC value and made the ROC curve. * Created a chart for comparing train, test accuracies and AUC scores of all models.   **Final Report**   * Results |
| **Anirudh Rajaram (0754920)** | **Project work**   * Performed data exploration * Applied **Logistic Regression** algorithm which yielded 77.08% train accuracy and 78.12% test accuracy. * Used gridsearch to find the best parameters for Logistic Regression and use them to improve the overall accuracy of the model. * Performed evaluation metrics on Logistic Regression by creating a confusion matrix and a classification report and then calculated the AUC value and made the ROC curve.   **Final Report**   * Methods |
| **Meenakshi Jayakeerthi (0754457)** | **Project work**   * Performed data exploration * Applied **Random Forest** algorithm which yielded 79.86% train accuracy and 79.17% test accuracy. * Used gridsearch to find the best parameters for Random Forest and use them to improve the overall accuracy of the model. * Performed evaluation metrics on Random Forest by creating a confusion matrix and a classification report and then calculated the AUC value and made the ROC curve.   **Final Report**   * Discussion |
| **Sharanjit kaur (0755577)** | **Project work**   * Performed data exploration * Applied **SVM** algorithm which yielded 78.47% train accuracy and 77.60% test accuracy. * Used gridsearch to find the best parameters for SVM and use them to improve the overall accuracy of the model. * Performed evaluation metrics on SVM by creating a confusion matrix and a classification report and then calculated the AUC value and made the ROC curve.   **Final Report**   * Related work |
| **Tejal Dalvi (0756090)** | **Project work**   * Performed data exploration * Used gridsearch to find the best parameters for **KNN** and use them to improve the overall accuracy of the model. * Applied KNN algorithm which yielded 81.60% train accuracy and 75.52% test accuracy. * Performed evaluation metrics on KNN by creating a confusion matrix and a classification report and then calculated the AUC value and made the ROC curve.   **Final Report**   * Introduction * Conclusion |

# References

* **Dataset (Kaggle):** <https://www.kaggle.com/uciml/pima-indians-diabetes-database>
* **Diabetes:**
  + <https://www.who.int/features/factfiles/diabetes/en/>
  + <https://www.healthline.com/health/diabetes#:~:text=Diabetes%20mellitus%2C%20commonly%20known%20as,the%20insulin%20it%20does%20make>.
  + <https://www.diabetes.ca/about-diabetes/what-is-diabetes>
  + <https://www.webmd.com/diabetes/how-sugar-affects-diabetes>
  + <https://www.webmd.com/diabetes/high-blood-pressure#:~:text=Diabetes%20damages%20arteries%20and%20makes,heart%20attack%2C%20and%20kidney%20failure>.
* **Machine learning:**
  + <https://en.wikipedia.org/wiki/Hyperparameter_optimization>
  + <https://www.jeremyjordan.me/hyperparameter-tuning/>
* **Cover image:**
* <https://www.freepik.com/free-photo/hand-holding-blood-glucose-meter-measuring-blood-sugar-background-is-stethoscope-chart-file_1193187.htm#page=1&query=diabetes&position=2>

# Appendices

* **Group24FinalReportHA.docx:** Contains the final report
* **Group42Projectha.ipynb**: Contains all the code for our project including importing, preprocessing, exploratory data analysis, machine learning algorithms and evaluation metrics.