Task 2

DIABETES CASE STUDY

INTRODUCTION

Diabetes is a major health issue affecting millions of people worldwide. Diabetes can lead to nerve damage, visual disturbance, risk of health disease, etc. Early detection and intervention can easily prevent the severity of these outcomes. In this project, we were assigned a task to create a classification model from the set of machine learning algorithms we study during the course, that can predict whether someone will develop diabetes using the given health parameters.

DETAILS ABOUT DATASET

The dataset contains 768 individuals' data on various attributes such as pregnancies, glucose, blood pressure, skin thickness, insulin, BMI, diabetes pedigree function, and age. These are the independent variables, and we use these independent variables to predict the dependent variable, which is the outcome of whether a person has diabetes.

- Pregnancies: - Number of times someone e has been pregnant
- Glucose: - Glucose concentration in blood
- Blood Pressure: - Diastolic Blood Pressure (mm hg)
- Skin Thickness: - Triceps skin fold thickness(mm)
- Insulin: - 2-hour serum insulin (mu U/ml)
- BMI: - Body Mass Index ((weight in kg/height in m) ^2)
- Age: - Age(years)
- Diabetes Pedigree Function: -scores likelihood of diabetes based on family history)
- Outcome: - 0(does not have diabetes) or 1 (has diabetes)

EXPLORATORY DATA ANALYSIS

Before doing the exploratory data analysis, we must import all the necessary libraries like pandas, NumPy, matplotlib, seaborn, etc., and load the dataset. In exploratory data analysis, we try to understand the data and clean the data.

UNDERSTANDING THE DATASET

Head of the dataset: It helps to get familiar with the dataset by displaying the top 5 data records.

The shape of the dataset: Getting to know about the rows and columns of the dataset.

Type of columns: Knowledge of data types

Summary of data: to understand how data has been spread across the table.

From all this we can see that some columns have a minimum value of 0, which is impossible in the medical scenario, hence data cleaning becomes necessary. Also, there are some outliers.

DATA CLEANING

Data cleaning includes 2 steps. In the first step, we check the dataset for null values and count them. In the given dataset we do not have any null values. In the second step, we check for zero in 5 columns, count them, and replace it with mean or median. Glucose and blood pressure have normal distribution so in these columns 0 is replaced by mean and the rest 3 columns have skewed distribution, so the median replace zero.

DATA VISUALIZATION

In data visualization, we are plotting:

COUNT PLOT: It is plotted to see if the data is balanced or not. From the count plot, we can easily conclude that the number of people having diabetes is far less than the number of people who do not.

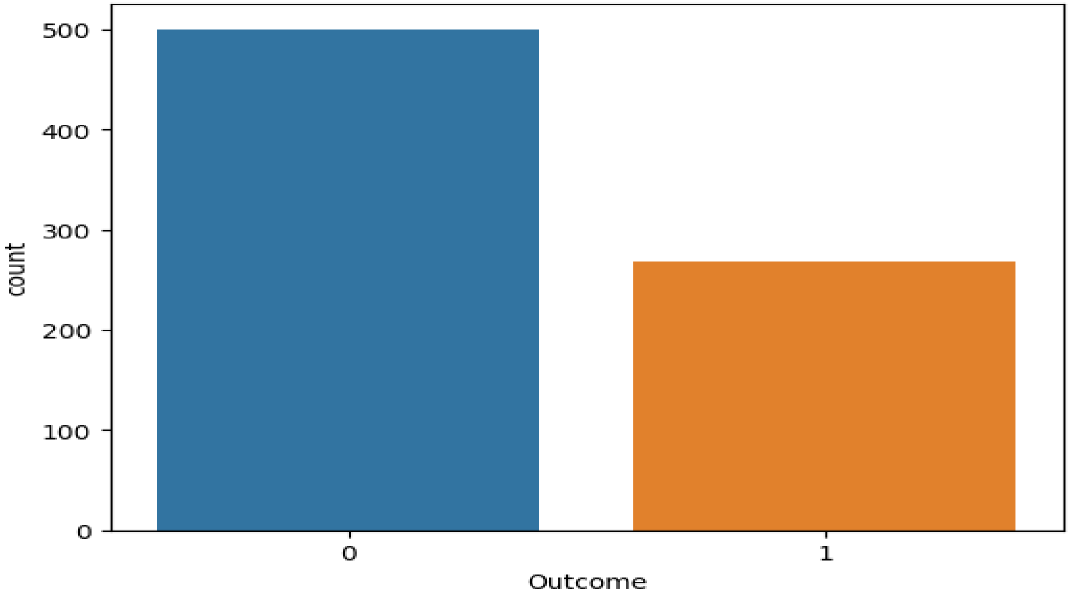


Fig 1.1 Count plot

HISTOGRAM: It is plotted to see the data distribution. It leads us to the conclusion of which feature has a normal distribution and which has a skewed distribution, which further helps in data cleansing. We concluded Glucose and blood pressure have a normal distribution and the rest have skewed distribution.

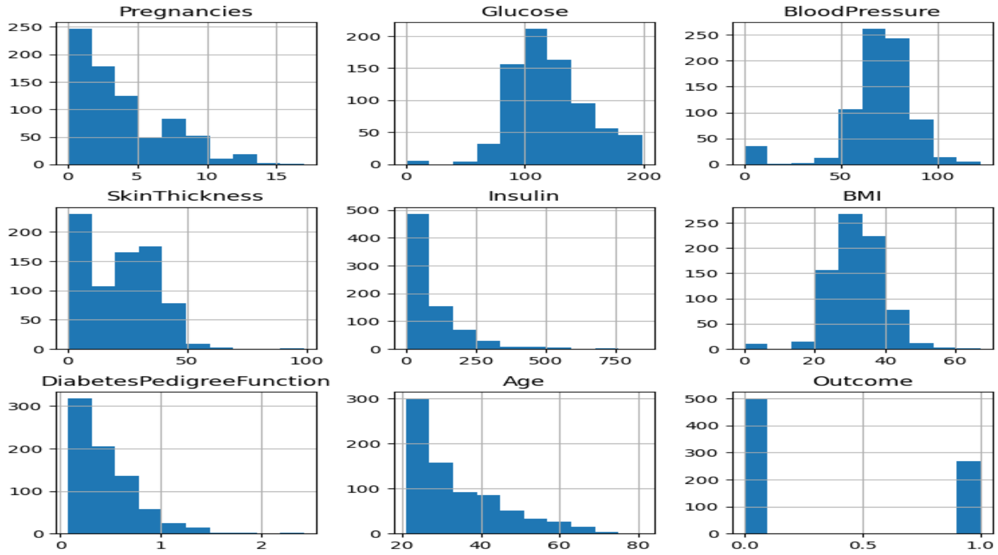


Fig 1.2 histogram of the dataset

BOXPLOT: It is plotted to see if there are any outliers in our data. It gives us the conclusion that our provided data has outliers.

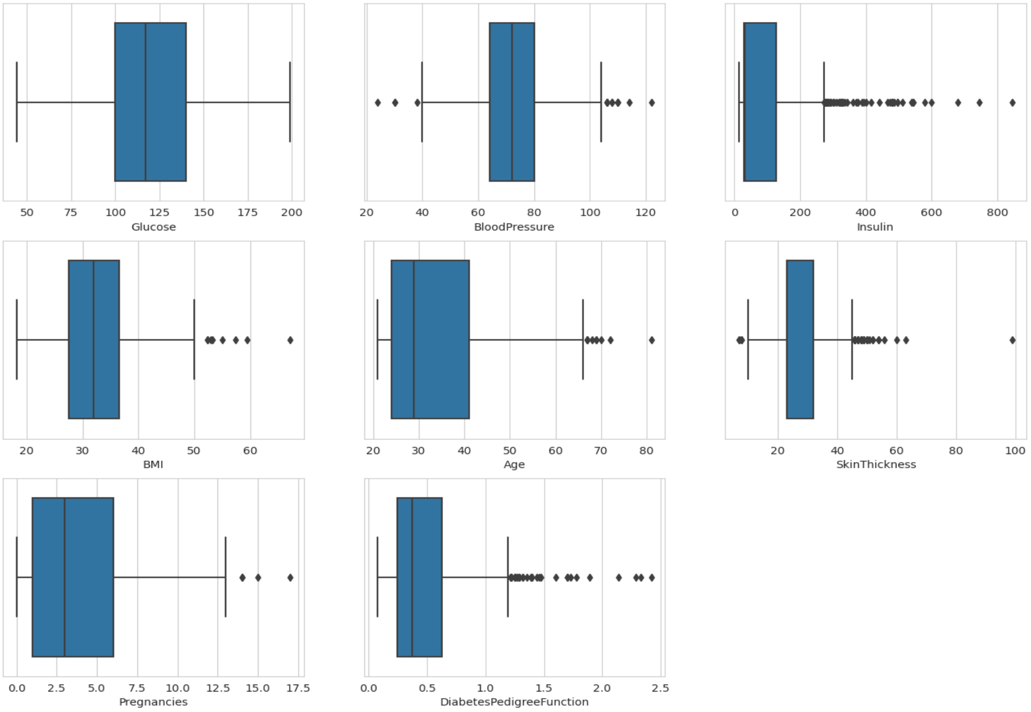


Fig 1.3 Boxplot of dataset

HEATMAPS: It helps to understand the correlation between all the given features in the data set. We can see that glucose, BMI, and age are the most correlated features with the outcome. Blood pressure, skin thickness, insulin, and diabetes pedigree function are the least correlated, so it does not contribute much to the outcome.

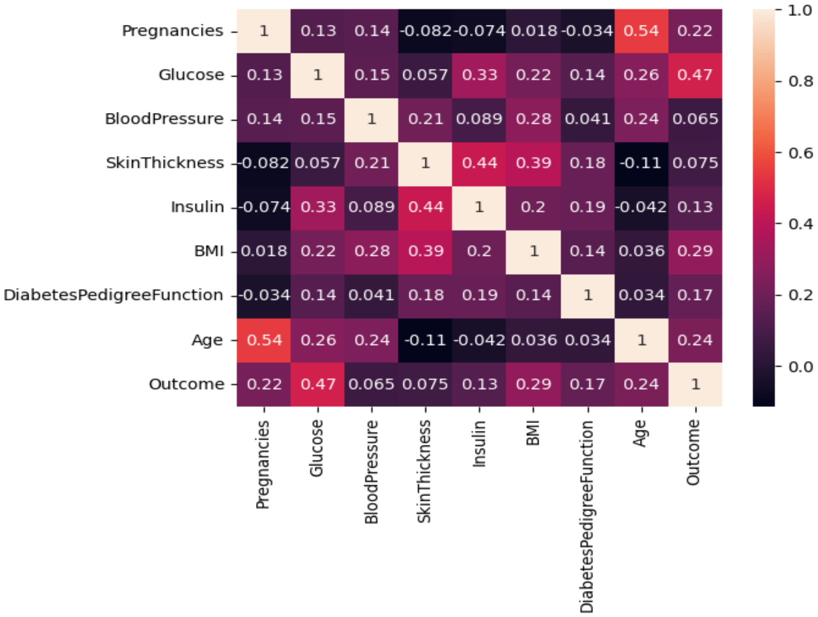


Fig 1.2 Heat map of the correlation matrix

TRAIN TEST SPLIT

The given data is split into a training set and a testing split to evaluate the algorithm’s performance. We use the train data to fit the algorithm and a test set is used to evaluate. I have split 70% of the data into the training set and the rest 30 % into the testing set.

MODEL BUILDING AND PERFORMANCE EVALUATION

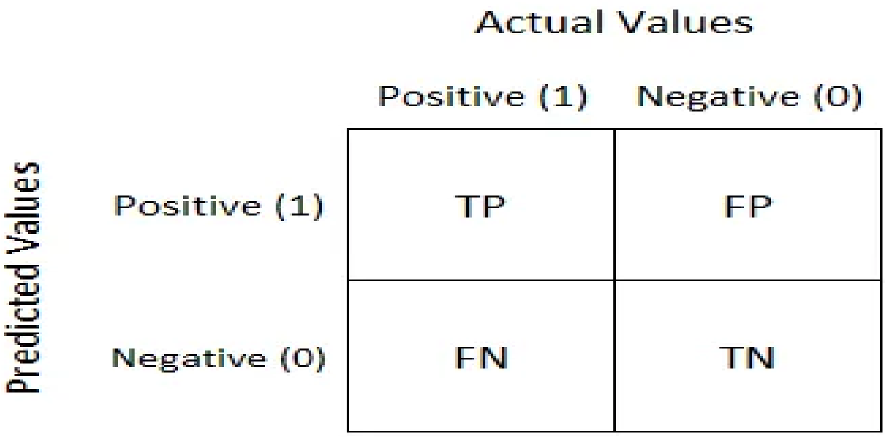
There are several algorithms we can use for classification. In this dataset, I have used 7 different classification algorithms.

- Logistic regression
- KNN Algorithm
- Decision tree algorithm
- Random forest algorithm
- Support vector machine algorithm
- XGBoost algorithm
- Gradient boost algorithm

And compared the performance of each algorithm using different matrices like confusion matrix, accuracy, specificity, sensitivity, and f1 score.

CONFUSION MATRIX

It is an N-squared matrix used for evaluating a machine-learning model.



		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Fig 1.4 Confusion matrix

The elements of the confusion matrix are used to find 3 important parameters.

1)ACCURACY SCORE

accuracy score = (number of correctly classified instances) / (total number of instances)

2)SENSITIVITY

The sensitivity of an algorithm refers to its ability to detect true positive cases, or in other words, its ability to correctly identify instances that belong to a certain category or class.

3)SPECIFICITY

Specificity is the number of correctly predicted negatives to the total number of negatives.

F1 SCORE

The harmonic mean of precision and recall is called F1 SCORE.

$$F1=2*(Precision*Recall)/(Precision+Recall)$$

Compared to the precision score and recall score, I preferred the F1 score since the given data is unbalanced and it is a better measure when we need a harmonic balance between precision and recall.

	ACCURACY	SENSITIVITY	SPECIFICITY	F1 SCORE
Logistic regression	*0.7857*	*0.8989*	*0.5818*	*0.7857*
KNN algorithm	*0.6948*	*0.7878*	*0.5273*	*0.6948*
Decision tree algorithm	*0.6688*	*0.7071*	*0.6688*	*0.6688*
Random forest algorithm	*0.6688*	*0.7071*	*0.6*	*0.6688*
XGBoost algorithm	*0.7338*	*0.8182*	*0.5818*	*0.7337*
Gradient boost algorithm	*0.7338*	*0.8182*	*0.5818*	*0.7337*
Support vector machine algorithm	*0.7922*	*0.9495*	*0.5091*	*0.792*

Table 1: Comparison Report of Accuracy, Sensitivity, Specificity, and F1_score concerning Different model

CONCLUSION

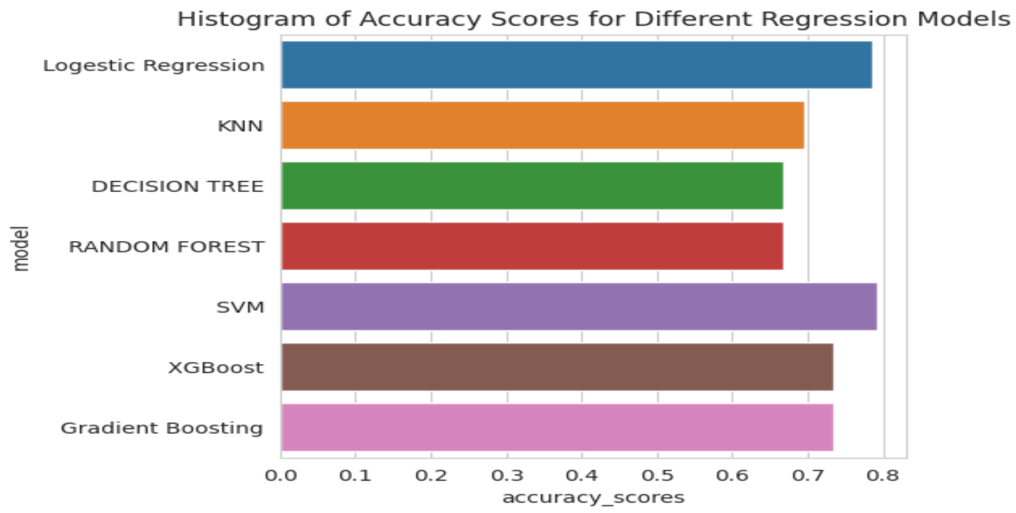


Figure 1. Bar plot of different algorithms concerning Accuracy

After comparing all these evaluation matrices, I found that the SVM algorithm performs the best, with an accuracy of 0.79, sensitivity of 0.95, specificity of 0.51, and an F1 score of 0.64. Even KNN gives the same accuracy when we change the value of N_neighbours to 22 or above but the value of sensitivity, specificity, and f1_score is not comparable to SVM. Logistic regression is the next best model compared followed by XGBoost and Gradient boost.

This is the link for the GitHub repository "https://github.com/AmmuVA/Diabetes"