NEURAL NETWORKS AND MACHINE LEARNING
ASSIGNMENT - 2

Group 6 Members:

AMISHA TRIPATHY ROLL NO. - 1804426

MOHIT KESHWANI ROLL NO. - 1804450

RIYA PATEL ROLL NO. - 1804464

TWISHA MUKHERJEE ROLL NO. - 1804558

Objective:

- **PREDICTION OF DIABETES USING SVM AND MLP ALGORITHM.**

Problem Statement:

- India is the diabetes capital of the world with as many as 50 million people suffering from diabetes, which causes increase in blood sugar.It is considered as one of the deadliest and chronic diseases and many complications can occur if diabetes remains untreated and unidentified. However, medical experts feel that timely detection and right management can go a long way in helping patients lead a normal life.
- Therefore, the rise in machine learning approaches can effectively solve this critical problem. The motive of this project is to design a model which can prognosticate the likelihood of diabetes in patients with maximum accuracy. Perhaps, two Machine learning classification algorithms namely **Support Vector Machine(SVM)** and **Multi Linear Perceptron(MLP)** are used to detect diabetes at an early stage

Technology And Libraries Used:

- Machine Learning
- Scikit-Learn
- Python
- Numpy
- Pandas
- Matplotlib

- Seaborn
- Jupyter Notebook

PROJECT WORKFLOW:

- Data Acquisition
- Data Exploration
- Missing Values
    - Replace all 0s using Simple Imputer
- Visualization
- Splitting The Dataset into train and test
- Data Normalization
- Feature Selection
- Check Class Imbalance
    - SMOTE
- Model Building
    - Model Hyperparameter Optimization
    - Model Selection
- Final Result

DATASET INFORMATION *:*

- The Pima Indian Diabetes Dataset consists of information on 768 patients (268 tested positive and 500 tested negative). Tested positive and tested negative indicates whether the patient is diabetic or not. The datasets consist of several medical predictor (independent) variables and one target (dependent) variable, Outcome. Independent variables include the number of pregnancies the patient has had, their Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function and Age. In particular, all patients here are females at least 21 years old of Pima Indian heritage.

The Dataset has been taken from
kaggle : https://www.kaggle.com/uciml/pima-indians-diabetes-database

Research paper Referred:
https://www.sciencedirect.com/science/article/pii/S1877050918308548

Our Code: https://github.com/Amishatripathy22/Diabetes-Prediction

1) Data Acquisition :

The Dataset has been taken from
kaggle : https://www.kaggle.com/uciml/pima-indians-diabetes-database

The dataset contains 768 rows and 9 features.

Original features in the dataset:

1. **Pregnancies** - Number of times pregnant
2. **Glucose**- Plasma glucose concentration(2 hours in an oral glucose tolerance test).
3. **Blood Pressure-** Diastolic blood pressure (mm Hg)
4. **Skin Thickness**- Triceps skin fold thickness (mm)
5. **Insulin**- 2-Hour serum insulin (mu U/ml)
6. **BMI**- Body mass index (weight in kg/(height in m)^2)
7. **Diabetes Pedigree Function**- Diabetes pedigree function
8. **Age**- Age in years
9. **Outcome**- Class variable (0 or 1) i.e if a person is diabetic(1) or not(0).

2) Data Exploration:

Having observed the basic characteristics of the dataset, we now move on to observe characteristics of the features.
The average age is is 33 whereas the median age is 29. This further confirms our analysis (as in case of normal distribution, the mean should be approximately equal to the median). None of the values seem abnormal, that is, the minimum age of 21 and the maximum age of 81 are possible.

Overview :

Overview | Warnings 5 | Reproduction

Dataset statistics

Number of variables	9
Number of observations	768
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	54.1 KiB
Average record size in memory	72.2 B

Variable types

NUM	8
BOOL	1

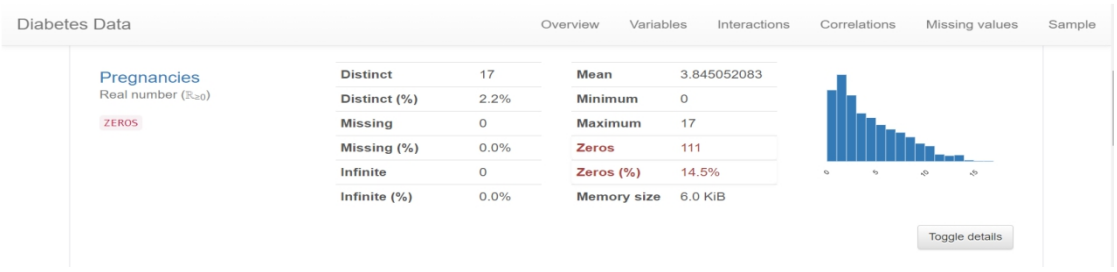
Warnings

`Pregnancies` has 111 (14.5%) zeros	Zeros
`BloodPressure` has 35 (4.6%) zeros	Zeros
`SkinThickness` has 227 (29.6%) zeros	Zeros
`Insulin` has 374 (48.7%) zeros	Zeros
`BMI` has 11 (1.4%) zeros	Zeros

Overview of the Dataset

The basic information about the dataset such as the size, missing values, etc.Here's 8 of numerical columns and 1 Boolean column . In the lower panel, (%) of zeros are given in every column.

FEATURES :



Statistics | Histogram | Common values | Extreme values

Quantile statistics

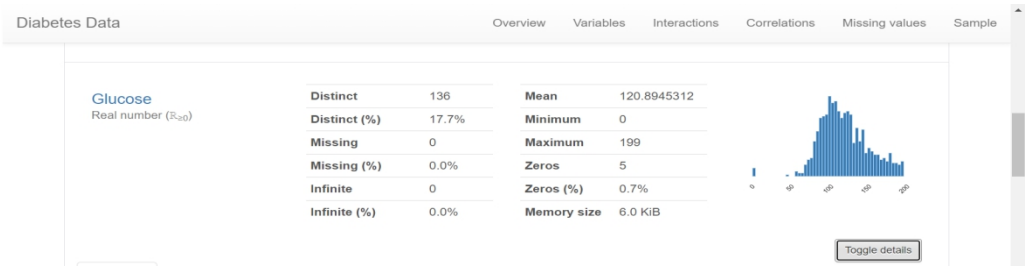
Minimum	0
5-th percentile	0
Q1	1
median	3
Q3	6
95-th percentile	10
Maximum	17
Range	17
Interquartile range (IQR)	5

Descriptive statistics

Standard deviation	3.369578063
Coefficient of variation (CV)	0.8763413316
Kurtosis	0.1592197775
Mean	3.845052083
Median Absolute Deviation (MAD)	2
Skewness	0.9016739792
Sum	2953
Variance	11.35405632
Monotocity	Not monotonic

Independent Variable : Pregnancy

The variable is positively skewed with 14.5% zero values. We need not treat zero values as its not an abnormal occurrence. 85.5% of the values is non-zero.



Statistics | Histogram | Common values | Extreme values

Quantile statistics

Minimum	0
5-th percentile	79
Q1	99
median	117
Q3	140.25
95-th percentile	181
Maximum	199
Range	199
Interquartile range (IQR)	41.25

Descriptive statistics

Standard deviation	31.9726182
Coefficient of variation (CV)	0.2644670347
Kurtosis	0.6407798204
Mean	120.8945312
Median Absolute Deviation (MAD)	20
Skewness	0.1737535018
Sum	92847
Variance	1022.248314
Monotocity	Not monotonic

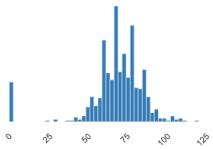
Independent Variable : Glucose

- Analyzing Glucose, we observe the variable not following the normal distribution. We encounter zero-values in this instance as well. There are 5 such values for which treatment is required. We perform the same treatment as before, replacing by median (class-wise).

BloodPressure
Real number (ℝ≥0)

ZEROS

Distinct	47	Mean	69.10546875
Distinct (%)	6.1%	Minimum	0
Missing	0	Maximum	122
Missing (%)	0.0%	Zeros	35
Infinite	0	Zeros (%)	4.6%
Infinite (%)	0.0%	Memory size	6.0 KiB



Toggle details

Statistics | Histogram | Common values | Extreme values

Quantile statistics

Minimum	0
5-th percentile	38.7
Q1	62
median	72
Q3	80
95-th percentile	90
Maximum	122
Range	122
Interquartile range (IQR)	18

Descriptive statistics

Standard deviation	19.35580717
Coefficient of variation (CV)	0.2800908166
Kurtosis	5.18015656
Mean	69.10546875
Median Absolute Deviation (MAD)	8
Skewness	-1.843607983
Sum	53073
Variance	374.6472712
Monotocity	Not monotonic

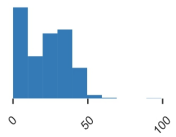
Independent Variable : Blood Pressure

- The variable can approximate to a normal distribution. However, we can not confirm that visually. The null hypothesis (*Ho)* is that the data is normal.

SkinThickness
Real number (ℝ≥0)

ZEROS

Distinct count	51	Mean	20.53645833
Unique (%)	6.6%	Minimum	0
Missing	0	Maximum	99
Missing (%)	0.0%	Zeros	227
Infinite	0	Zeros (%)	29.6%
Infinite (%)	0.0%	Memory size	6.1 KiB



Toggle details

Statistics | Histogram(s) | Common values | Extreme values

Quantile statistics

Minimum	0
5-th percentile	0
Q1	0
median	23
Q3	32
95-th percentile	44
Maximum	99
Range	99
Interquartile range (IQR)	32

Descriptive statistics

Standard deviation	15.95221757
Coefficient of variation (CV)	0.776775494
Kurtosis	-0.5200718662
Mean	20.53645833
Median Absolute Deviation (MAD)	13.65962728
Skewness	0.1093724965
Sum	15772
Variance	254.4732453

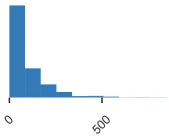
Independent Variable : Skin Thickness

- The data is positively skewed with 29.6% of zero values.

Insulin
Real number ($\mathbb{R}_{\geq 0}$)

ZEROS

Distinct count	186	Mean	79.79947917
Unique (%)	24.2%	Minimum	0
Missing	0	Maximum	846
Missing (%)	0.0%	Zeros	374
Infinite	0	Zeros (%)	48.7%
Infinite (%)	0.0%	Memory size	6.1 KiB



Toggle details

Statistics | Histogram(s) | Common values | Extreme values

Quantile statistics

Minimum	0
5-th percentile	0
Q1	0
median	30.5
Q3	127.25
95-th percentile	293
Maximum	846
Range	846
Interquartile range (IQR)	127.25

Descriptive statistics

Standard deviation	115.2440024
Coefficient of variation (CV)	1.444169856
Kurtosis	7.214259554
Mean	79.79947917
Median Absolute Deviation (MAD)	84.50507948
Skewness	2.272250858
Sum	61286
Variance	13281.18008

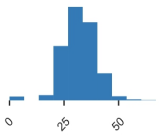
Independent Variable : Insulin

- The variable is positively skewed. However, the occurrence of zero-values is high in this case, making up 48.7% of the data.

BMI
Real number ($\mathbb{R}_{\geq 0}$)

ZEROS

Distinct count	248	Mean	31.99257812
Unique (%)	32.3%	Minimum	0.0
Missing	0	Maximum	67.1
Missing (%)	0.0%	Zeros	11
Infinite	0	Zeros (%)	1.4%
Infinite (%)	0.0%	Memory size	6.1 KiB



Toggle details

Statistics | Histogram(s) | Common values | Extreme values

Quantile statistics

Minimum	0
5-th percentile	21.8
Q1	27.3
median	32
Q3	36.6
95-th percentile	44.395
Maximum	67.1
Range	67.1
Interquartile range (IQR)	9.3

Descriptive statistics

Standard deviation	7.88416032
Coefficient of variation (CV)	0.2464371671
Kurtosis	3.290442901
Mean	31.99257812
Median Absolute Deviation (MAD)	5.842269897
Skewness	-0.4289815885
Sum	24570.3
Variance	62.15998396

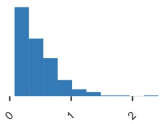
Independent Variable : BMI

- The variable seems to be closely following the normal distribution as the mean and median are approximately equal.

DiabetesPedigreeFunction

Real number (ℝ≥0)

Distinct count	517	**Mean**	0.4718763021
Unique (%)	67.3%	**Minimum**	0.078
Missing	0	**Maximum**	2.42
Missing (%)	0.0%	**Zeros**	0
Infinite	0	**Zeros (%)**	0.0%
Infinite (%)	0.0%	**Memory size**	6.1 KiB



Toggle details

Statistics | Histogram(s) | Common values | Extreme values

Quantile statistics

Minimum	0.078
5-th percentile	0.14035
Q1	0.24375
median	0.3725
Q3	0.62625
95-th percentile	1.13285
Maximum	2.42
Range	2.342
Interquartile range (IQR)	0.3825

Descriptive statistics

Standard deviation	0.331328595
Coefficient of variation (CV)	0.7021513764
Kurtosis	5.594953528
Mean	0.4718763021
Median Absolute Deviation (MAD)	0.24730857
Skewness	1.919911066
Sum	362.401
Variance	0.1097786379

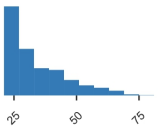
Independent Variable : Diabetes Pedigreefunction

- Diabetes Pedigree Function is a positively skewed variable with no zero values.

Age

Real number (ℝ≥0)

Distinct count	52	**Mean**	33.24088542
Unique (%)	6.8%	**Minimum**	21
Missing	0	**Maximum**	81
Missing (%)	0.0%	**Zeros**	0
Infinite	0	**Zeros (%)**	0.0%
Infinite (%)	0.0%	**Memory size**	6.1 KiB



Toggle details

Statistics | Histogram(s) | Common values | Extreme values

Quantile statistics

Minimum	21
5-th percentile	21
Q1	24
median	29
Q3	41
95-th percentile	58
Maximum	81
Range	60
Interquartile range (IQR)	17

Descriptive statistics

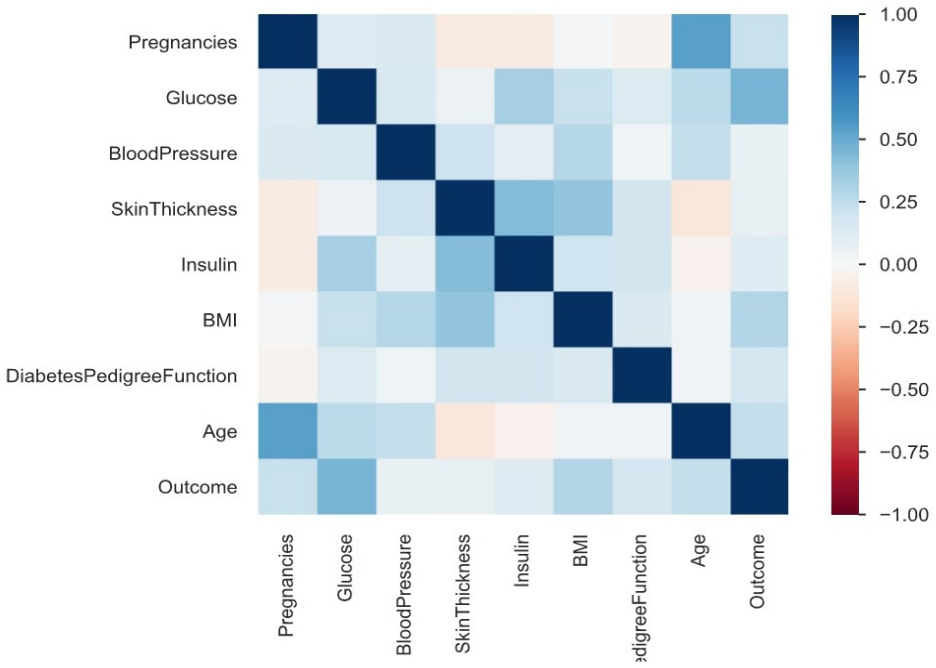
Standard deviation	11.76023154
Coefficient of variation (CV)	0.3537881556
Kurtosis	0.6431588885
Mean	33.24088542
Median Absolute Deviation (MAD)	9.586405436
Skewness	1.129596701
Sum	25529
Variance	138.3030459

Independent Variable : Age

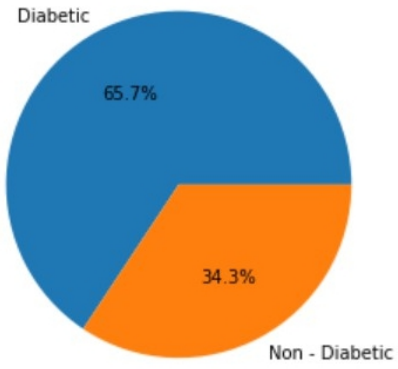
- The minimum age of 21 and the maximum age of 81 are possible.

CORRELATION :

The correlation matrix below uses Pearson's correlation coefficient to illustrate the relationship between variables. From the figure, a significant correlation can be observe between Pregnancies and Age.

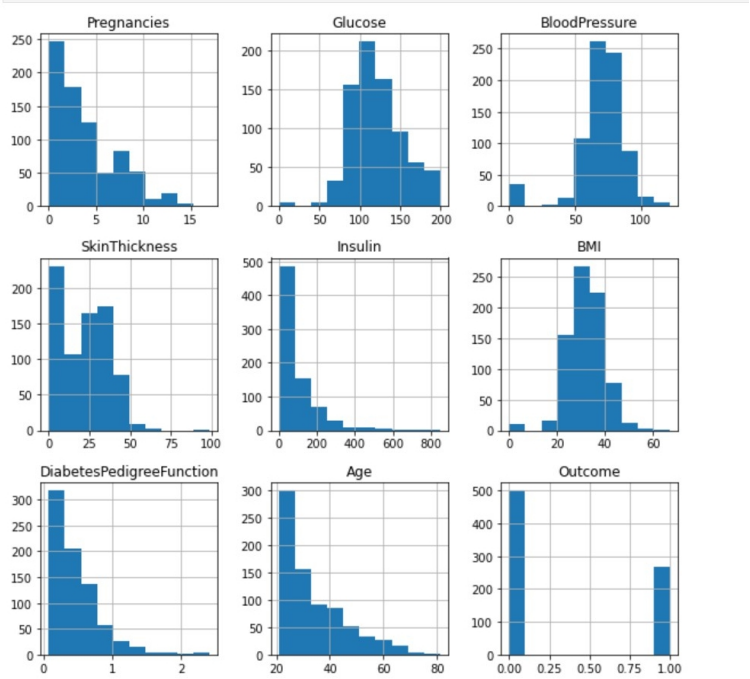


Distribution of target class(Outcome) which states if a person is diabetic or not:



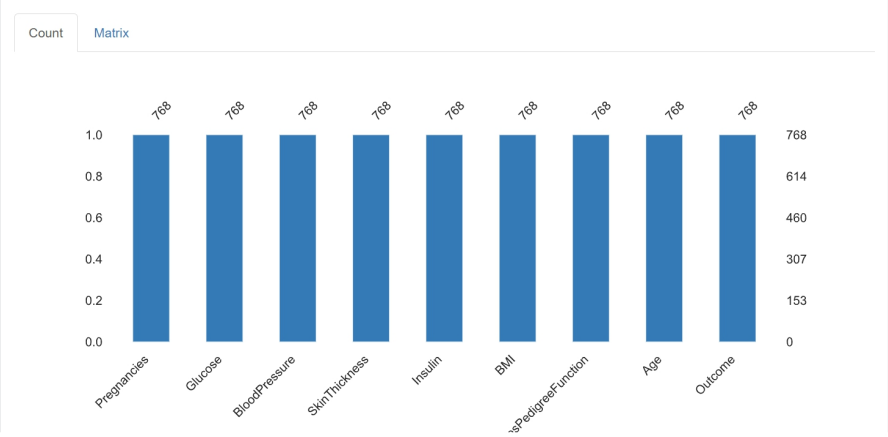
- Percentage of people having Diabetes 65%
- Percentage of people doesn't having Diabetes 34%

Distribution of features in the dataset:



- **Pregnancies:** Positively Skewed.
- **Glucose:** Not normally distributed, 0.7% of values are zero.
- **Blood Pressure**: Not normally distributed, 4.6% of values are zero.
- **Skin Thickness:** Not normally distributed, 29.6% of values are zero.
- **Insulin:** Positively Skewed, 48.7% of values are zero.
- **BMI:** Normal distribution, 1.4% of values are zero.
- **Diabetes Pedigree Function:** Positively Skewed.
- **Age:** Positively Skewed.

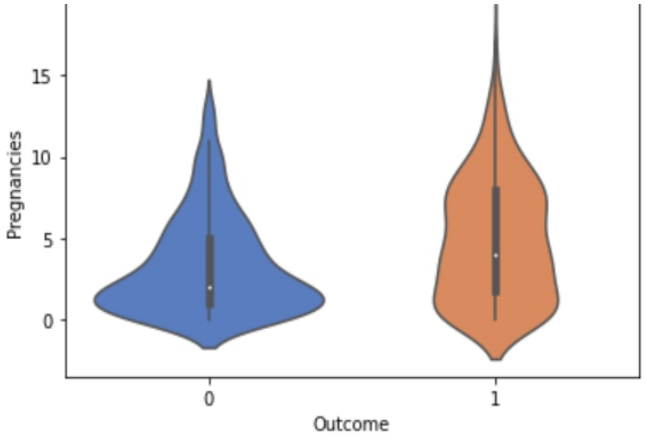
3) MISSING VALUES:



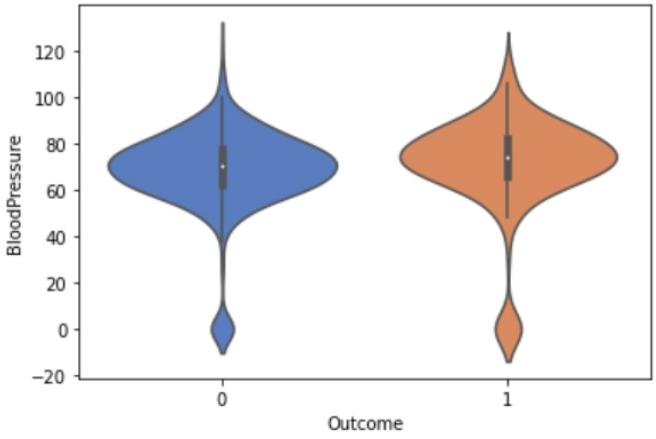
- So from the above histogram its clear that there are no missing values in the dataset. But, as we move forward we will see that are lots of 0 values in some columns which practically can't be 0.

Analysing the distribution of the features using violin plot to find which statistical method(mean, median, mode, constant) will be helpful in imputing the zero values.

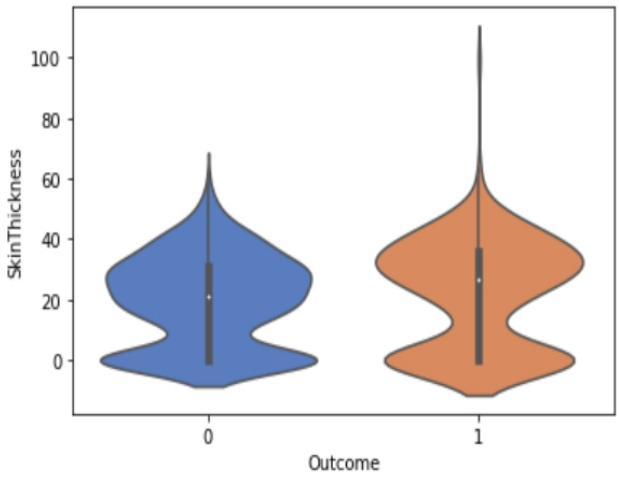
- Using violin plots we observe diabetic women had more pregnancies than non-diabetic.



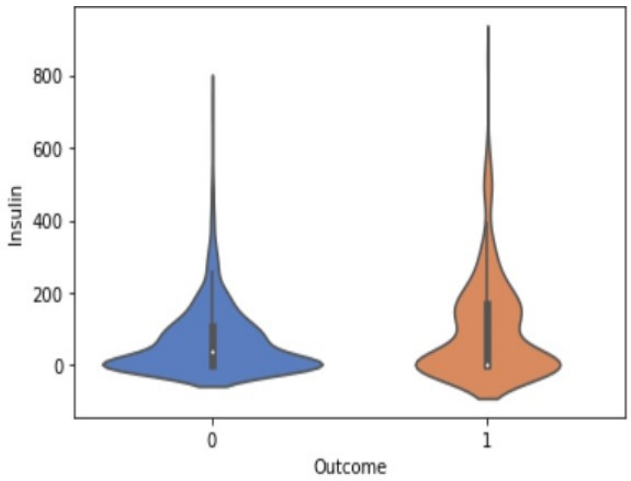
- The box plot for *1 (Diabetic)* inside the Violin is a little more away from the horizontal axis than the box plot for *0 (Non Diabetic)*. It can be implied that diabetics seem to have a higher blood pressure than the non-diabetics.



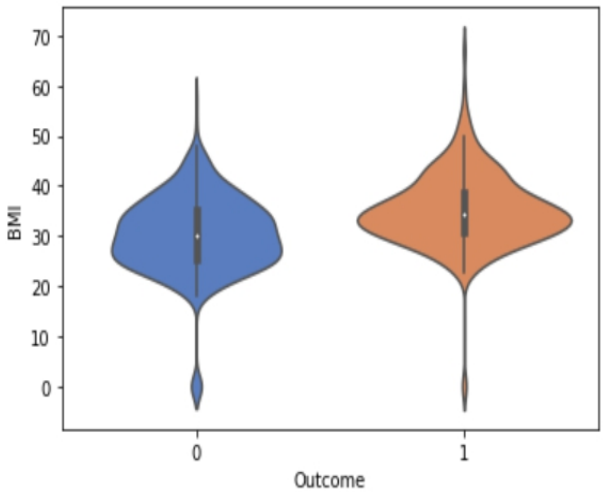
- Skin Thickness for Diabetics is more than that of Non-Diabetics



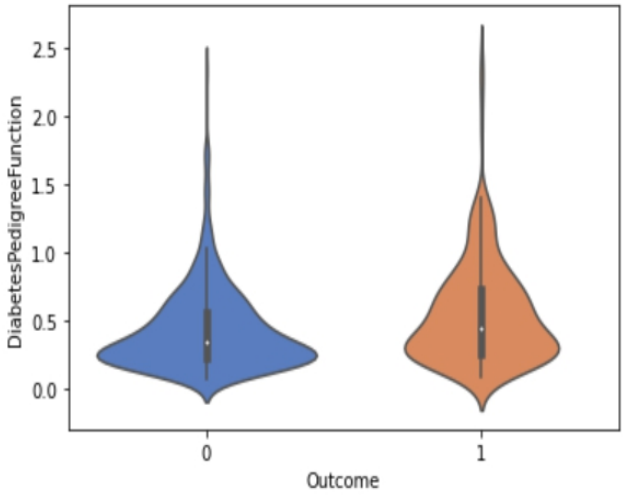
- We can still see 0 as the median for Insulin for Diabetics. However, for Non-Diabetics, Insulin is a little higher.



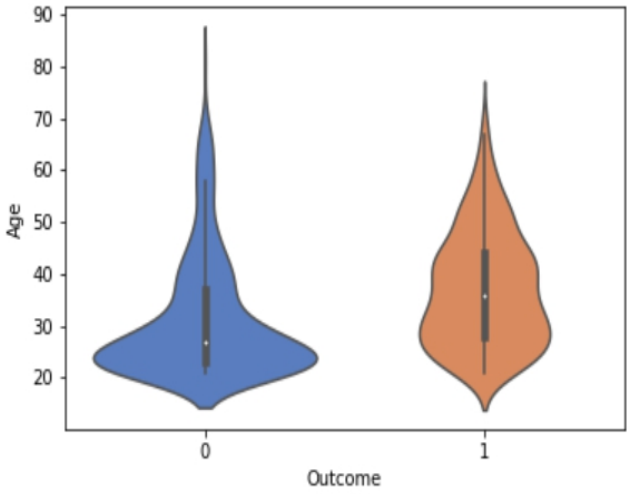
- 1 follows normal distribution, while 0 doesn't. Also, BMI for diabetics is more than BMI for non-diabetics.



- Diabetes Pedigree Function is a positively skewed variable with no zero values.



- Diabetics seem to have a higher pedigree function than the non-diabetics.



3.1) Filling Missing Values:

We have used **Simple Imputer** to replace zero values of different features with the **mean** of that particular feature.
Though there are other statistical methods as well (median, mode and constant), but mean best suits our dataset.

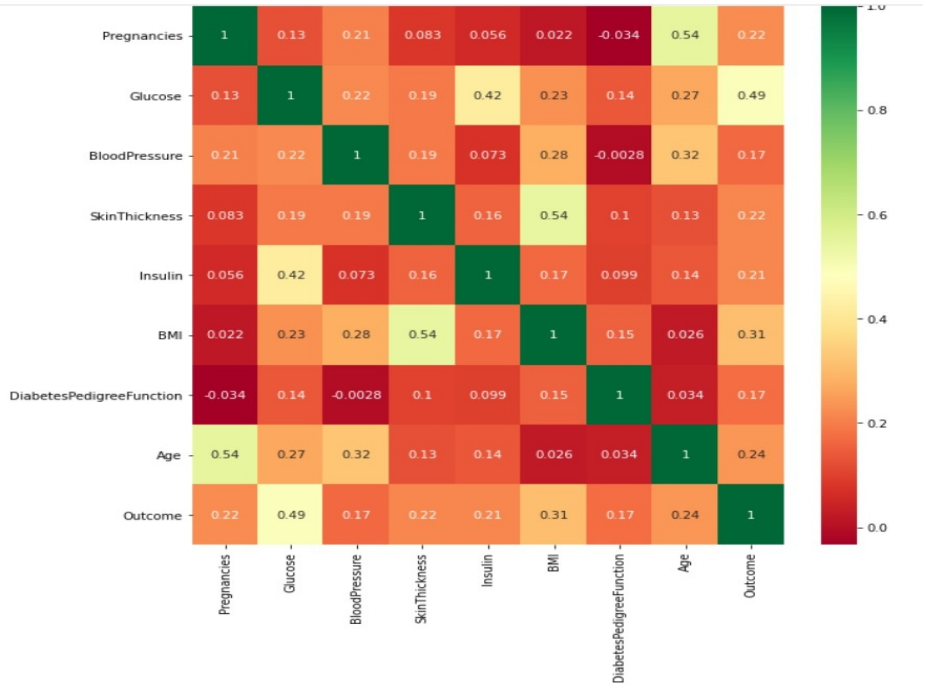
```
from sklearn.impute import SimpleImputer
from sklearn.pipeline import Pipeline
from sklearn.ensemble import RandomForestClassifier
from numpy import mean, std
from sklearn.model_selection import cross_val_score, RepeatedStratifiedKFold

X = data.drop(['Outcome'], axis=1)
y = data.Outcome
results = list()
strategies = ['mean', 'median', 'most_frequent', 'constant']
for s in strategies:
    pipeline = Pipeline(steps=[('i', SimpleImputer(missing_values = 0, strategy=s)), ('m', RandomForestClassifier())
    cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)
    scores = cross_val_score(pipeline, X, y, scoring='accuracy', cv=3, n_jobs=-1)
    results.append(scores)
    print('%s %.3f (%.3f)' % (s, mean(scores), std(scores)))
```

We Imputed all the 4 strategies(mean, median, mode and constant) on our dataset by creating a **pipeline** that after imputing fits our dataset to the **RandomForest** model, which returns the cross validation score of all the four strategies, with mean having highest score.

```
mean 0.757 (0.026)
median 0.768 (0.022)
most_frequent 0.757 (0.018)
constant 0.766 (0.022)
```

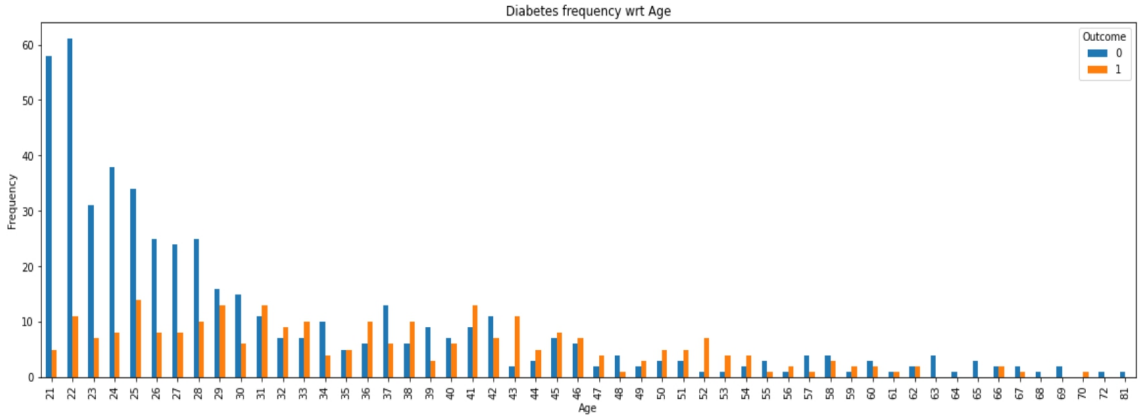
Correlation:

The correlation matrix below uses Pearson's correlation coefficient to illustrate the relationship between variables. From the figure, a significant correlation can be observed between Pregnancies and Age. To further confirm, we calculate the correlation coefficient.

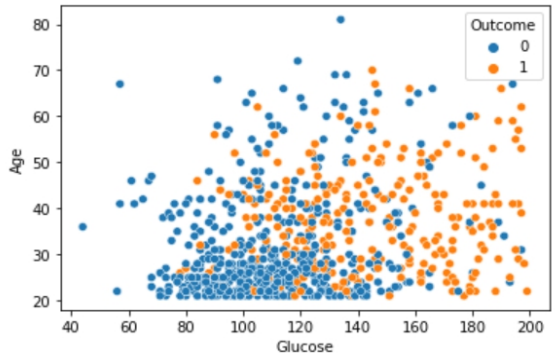


4) Data Visualization:

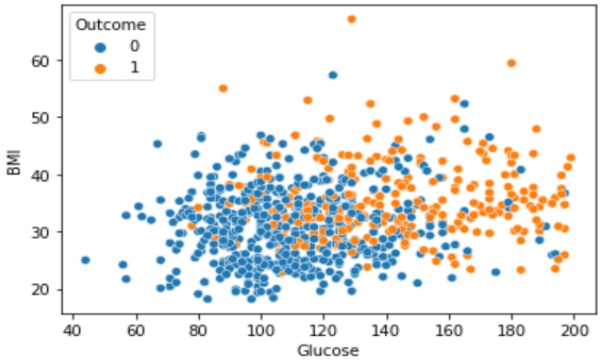
- Histogram plot depicting the frequency of diabetic and non-diabetic patients with respect to age of the patient.
  Min age- 21
  Max age-81



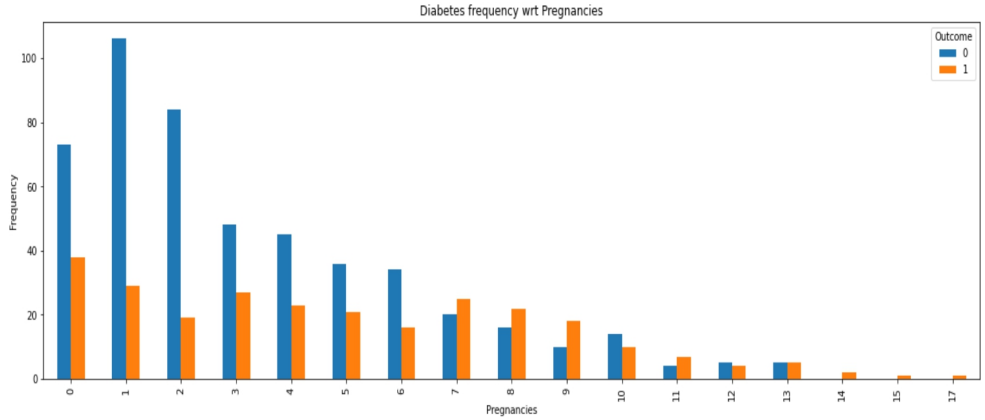
- The following scatter plot depicts the glucose value of diabetic and non-diabetic patient with respect to the age of the patient.
  Patients above the age of **30**, tends to have high glucose level and are more likely to be diabetic.



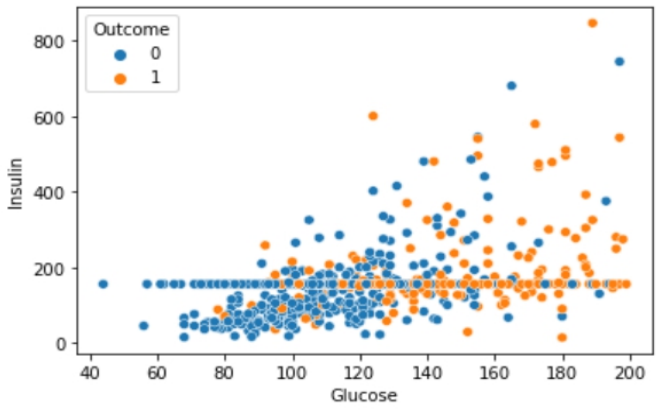
- The following scatter plot clearly depicts that diabetic patients above the age of 30, have high BMI(Body mass Index) i.e less likely to be working out and have a high glucose level.



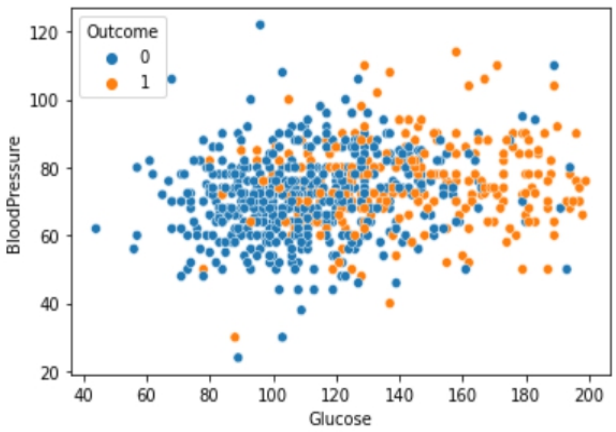
- The following bar plot states that female patients having 0 or 1 pregnancies are more likely to be diabetic than the rest.



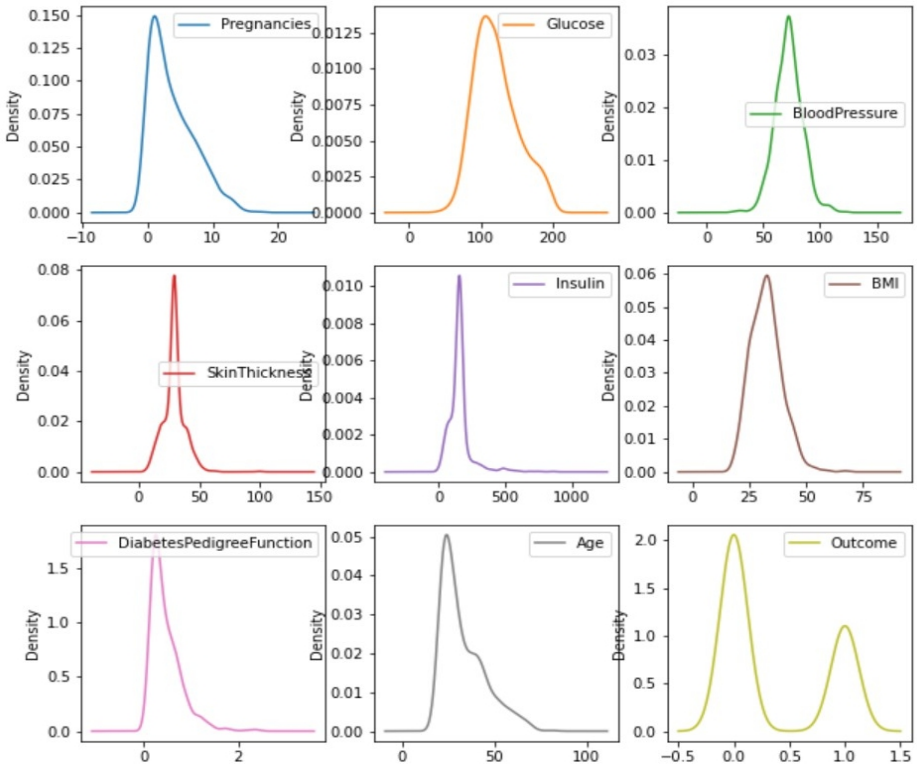
- Diabetic Patients with high glucose level also have high insulin level.



- The scatter plot exactly proves the notion that patients who have **high glucose level** generally have **high Blood Pressure** as well and are more likely to be diabetic.



- Density plot of different features of our imputed dataset.



5) Splitting the Dataset into train and test

We have split the dataset, with 75% of train data and 25% test data.

```
from sklearn.model_selection import train_test_split, cross_val_score

X_train, X_test, y_train, y_test = train_test_split(data.drop(['Outcome'], axis = 1),
                                                    data.Outcome,
                                                    test_size = 0.25,
                                                    random_state = 20)

print("Training set has {} samples.".format(X_train.shape[0]))
print("Testing set has {} samples.".format(X_test.shape[0]))
```

Thus we got 576 samples in train set and 192 samples in test set.

6) DATA NORMALIZATION:

Outliers are extreme values existing in the dataset. It is necessary to treat outliers if a distance-based algorithm is applied on the dataset. Therefore we have used **minmax scaler** to normalize the numerical values in the range of 0-1, so that the model can effectively understand the relation between independent features in accordance to the target feature.

```
#import MinMaxScaler from sklearn to normalize the numerical features
from sklearn.preprocessing import MinMaxScaler


#initializing the scaler function
scaler = MinMaxScaler()
scaler.fit(X_train[:])
#fitting and transposing all the numerical features
X_train[:] = scaler.transform(X_train[:])
X_test[:] = scaler.transform(X_test[:])
```

7) FEATURE SELECTION:

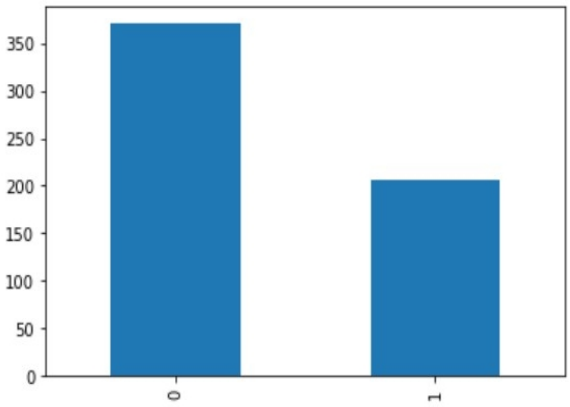
Feature Selection is the process of selecting important independent features which are providing more value in predicting the target feature. This is the really important part of preparing our data for modeling.
For this purpose we have used mutual info classification, which returns the score of each feature, depicting their importance in predicting the target feature(Outcome).

```
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.feature_selection import mutual_info_classif

mutual_info = mutual_info_classif(X_train, y_train)
mutual_info = pd.Series(mutual_info)
mutual_info.index = X_train.columns
mutual_info.sort_values(ascending = False)
```

Thus giving the following scores for each independent feature. Out of which **Skin Thickness** is of least importance and thus we have dropped this column to get better performance metrics.

```
Out[30]: Glucose                    0.156471
         Pregnancies                0.090263
         BMI                        0.075732
         Age                        0.053130
         Insulin                    0.050362
         BloodPressure              0.035952
         DiabetesPedigreeFunction   0.033350
         SkinThickness              0.006458
         dtype: float64
```

8) CLASS CHECK IMBALANCE:



Out of 576 samples in the train set, approximately 200 patients are diabetic and rest 376 are non diabetic. This shows high class imbalance which will result in poor prediction on test data as the model will not be able to learn appropriately and thus to deal with this problem we will **up-sample** the records for better performance and training of the model.

For this we will use **SMOTE** technique, though there are various up sampling and down sampling techniques, but our dataset is very small and it will lead to excessive loss of data if we will down sample the records of non-diabetic patients.

SMOTE :

Synthetic Minority Oversampling Technique(SMOTE) is used to remove the imbalance in the training data by creating samples using the current data.
It does not create duplicates and it is always done on the training data and not on the original data as the test data should only contain real life values and not synthetic samples.

9) Model Building:

Here we are going to train two classification algorithms-

I) Support Vector Machine(SVM)

II) Multi Layer Perceptron(MLP)

Performance Metrics which we are going to use to analyse the performance of our models are:

I) Accuracy
II) F-beta Score
III) Confusion Metrix
IV) AUC Score

9.1) Model building with default parameters:

	SVC	MLPClassifier
train_time	0.040790	1.483877
pred_time	0.048795	0.006472
acc_train	0.803333	0.783333
acc_test	0.760417	0.744792
f_train	0.755187	0.732510
f_test	0.628492	0.608808

✓ Both models are approximately performing equally, so next we will optimize both the models using **GridSearchCV** to get the best model out of the two.

9.2) Model Hyperparameter Optimization:

Now we have optimized both the models using GridSearchCV and calculated the performane matrix of each model.

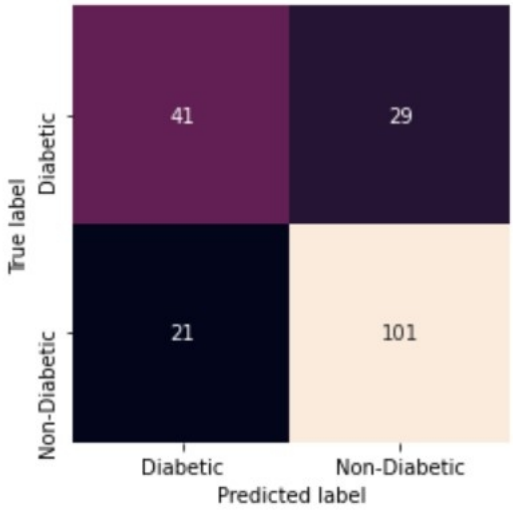
I) SVM:

Initializing the model and setting up appropriate parameters to optimize the SVM model and got final optimize hyper parameters.

```
#Initializing the model
clf = SVC()
#parameters that we are going to optimize
parameters = { 'C': [0.1, 0.5, 1, 10, 50, 100],
               'gamma': [0.001, 0.01, 0.05, 0.1, 0.5, 1, 10],
               'kernel': ['rbf']
             }
#creating the optimizer object
grid_obj_SVC = GridSearchCV(clf , parameters, n_jobs=-1)

#fitting the obj to out dataset
grid_fit_SVC = grid_obj_SVC.fit(X_train,y_train)
#finding the best parameters
best_clf1 = grid_fit_SVC.best_estimator_

#predictions on unoptimized model
predictions = (clf.fit(X_train, y_train)).predict(X_test)

#predictions on optimized model
best_predictions = best_clf1.predict(X_test)

print("Unoptimized model\n------")
print("Accuracy score on testing data: {:.4f}".format(accuracy_score(y_test, predictions)))
print("F-score on testing data: {:.4f}".format(fbeta_score(y_test, predictions, beta = 0.5)))
print("\nOptimized Model\n------")
print("Final accuracy score on the testing data: {:.4f}".format(accuracy_score(y_test, best_predictions)))
print("Final F-score on the testing data: {:.4f}".format(fbeta_score(y_test, best_predictions, beta = 0.5)))
print("Final Hyperparameters: %s" %grid_fit_SVC.best_params_)
```

```
Unoptimized model
------
Accuracy score on testing data: 0.7604
F-score on testing data: 0.6285

Optimized Model
------
Final accuracy score on the testing data: 0.7396
Final F-score on the testing data: 0.5994
Final Hyperparameters: {'C': 1, 'gamma': 10, 'kernel': 'rbf'}
```

Confusion Matrix and AUC Score of SVM:



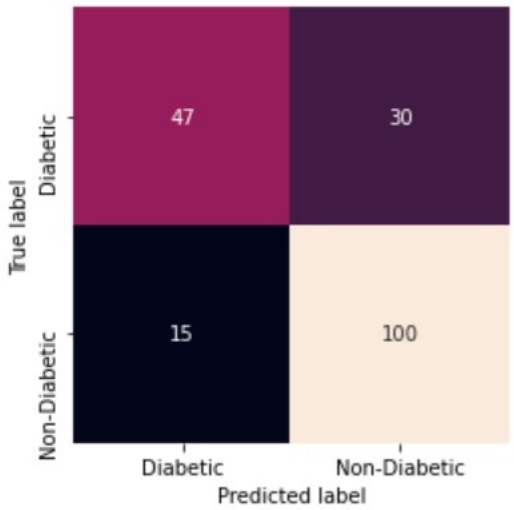
- **Accuracy:** 0.7396
- **Precision:** TP/ TP+ FP = 0.66
- **Recall:** TP/ TP+FN = 0.585
- **F-beta Score:** 0.5994
- **AUC Score:** 0.719

II) MLP:

Initializing the model and setting up appropriate parameters to optimize the MLP model and get optimized hyper parameters.

```
Unoptimized model
------
Accuracy score on testing data: 0.7708
F-score on testing data: 0.6417

Optimized Model
------
Final accuracy score on the testing data: 0.7656
Final F-score on the testing data: 0.6351
Final Hyperparameters: {'activation': 'relu', 'alpha': 0.1, 'hidden_layer_sizes': (120,), 'learning_rate': 'constan
t', 'solver': 'adam'}
```

Confusion Matrix and AUC score of MLP:



- ✧ **Accuracy:** 0.7656
- ✧ **Precision:** 0.758
- ✧ **Recall:** 0.61
- ✧ **F-beta Score:** 0.6351
- ✧ **AUC Score:** 0.763

Final Result:

1) So looking at all the performance matrix of the model, **MLP** outshines **SVM** in all respect.

Why MLP comes out to be better than SVM?

2) Here in diabetes prediction, type 1 error is more worse than the type 2 error i.e the goal is to reduce false positive. Therefore, beta in f beta score is 0.5 as we want to give more weith to precision then to recall, thus the model with more f-beta score is better.
3) As well AUC score of MLP is much better than SVM.

Conclusion:

1) Multilayer perceptron(MLP) is a class of feedforward artificial neural network. The term MLP is used ambiguously, sometimes loosely to any feedforward ANN, sometimes strictly to refer to networks composed of multiple layers of perceptrons.

2) A support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems. After giving an SVM model sets of labeled training data for each category, they're able to categorize new text.

3) Machine learning was employed based on steps of feature extraction, feature selection and classification. In this proposed report, we achieve the highest accuracy value of (Jo aae likh Dena) in diagnosing diabetes using MLP and SVM.