DIABETES PREDICTION

USING DECISION TREES

NIYATI JAIN- 189301051

DATA SCIENCE

SECTION B

TABLE OF CONTENTS

INTRODUCTION TO THE PROBLEM

Diabetes is a chronic disease that occurs when the pancreas is no longer able to make insulin, or when the body cannot make good use of the insulin it produces.

Insulin is a hormone made by the pancreas, that acts like a key to let glucose from the food we eat pass from the blood stream into the cells in the body to produce energy. All carbohydrate foods are broken down into glucose in the blood. Insulin helps glucose get into the cells.

Not being able to produce insulin or use it effectively leads to raised glucose levels in the blood (known as hyperglycaemia). Over the long-term high glucose levels are associated with damage to the body and failure of various organs and tissues

As the social economy has developed, the population has aged, and urbanisation has accelerated, certain changes in national lifestyles have occurred, leading to an increase in diabetes prevalence. Diabetes is a disorder that can be avoided and managed, and early detection can help to slow down its progression. We were aiming for a way to predict diabetes based on a specific culture, lifestyle, and behavior.

DATASET DESCRIPTION

DATA EXTRACTION

The dataset in consideration has been extracted from kaggle:
https://www.kaggle.com/vikasukani/diabetes-data-set

DETAILS

The dataset used has **2000** unique entries with **9** distinct features. The objective of the dataset is **to diagnostically predict whether or not a patient has diabetes**, based on certain diagnostic measurements included in the dataset. The datasets consist of 9 medical predictor (independent) variables and one target (dependent) variable, Outcome. The features are explained below:

1. Pregnancies: Number of times pregnant
2. Glucose: Plasma glucose conc for 2 hours in an oral glucose tolerance test
3. BloodPressure: Diastolic blood pressure (mm Hg)
4. SkinThickness: 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. DiabetesPedigreeFunction: Diabetes mellitus genetic history in relatives
8. Age: Age (years)
9. Outcome: Class variable (0 or 1) 268 of 768 are 1, the others are 0

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               2000 non-null   int64
 1   Glucose                   2000 non-null   int64
 2   BloodPressure             2000 non-null   int64
 3   SkinThickness             2000 non-null   int64
 4   Insulin                   2000 non-null   int64
 5   BMI                       2000 non-null   float64
 6   DiabetesPedigreeFunction  2000 non-null   float64
 7   Age                       2000 non-null   int64
 8   Outcome                   2000 non-null   int64
dtypes: float64(2), int64(7)
memory usage: 140.8 KB
```

```
df.describe()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000
mean	3.703500	121.182500	69.145500	20.935000	80.254000	32.193000	0.470930	33.090500	0.342000
std	3.306063	32.068636	19.188315	16.103243	111.180534	8.149901	0.323553	11.786423	0.474498
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	63.500000	0.000000	0.000000	27.375000	0.244000	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	40.000000	32.300000	0.376000	29.000000	0.000000
75%	6.000000	141.000000	80.000000	32.000000	130.000000	36.800000	0.624000	40.000000	1.000000
max	17.000000	199.000000	122.000000	110.000000	744.000000	80.600000	2.420000	81.000000	1.000000

PRE-PROCESSING

1. REMOVE DUPLICATES

```
df.drop_duplicates(inplace=True)
```

2. CHECK NULL VALUES TO REMOVE IF ANY

```
df.isna().sum()
```

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

3. REPLACE '0' VALUES WITH MEAN

```
featureList = ['Glucose', 'BloodPressure',  'SkinThickness',  'Insulin',  'BMI']
for col in featureList:
  df[col] = df[col].replace({ 0 : df[col].mean() })
```

4. REMOVING OUTLIERS

```
df.loc[ df.Pregnancies > 6, 'Pregnancies' ] = 6
df.loc[ df.Glucose < 70 , 'Glucose' ] = 70
df.loc[ df.BloodPressure < 60 , 'BloodPressure' ] = 60
df.loc[ df.BMI < 18 , 'BMI' ] = 18
df.loc[ df.BMI > 40 , 'BMI' ] = 40
```

5. STANDARD SCALER

```
sc = StandardScaler()
```

The above function has been imported from Scikit library in Python3. StandardScaler() function to standardize the data values such that its distribution will have a **mean value 0** and **standard deviation of 1**. This is done so that the variance of the features are in the same range. It standardizes features by removing the mean and scaling to unit variance.The standard score of a sample x is calculated as:

$$z = (x - u) / s$$

where u is the mean of the training samples and s is the standard deviation of the training samples.

ALGORITHM USED

DECISION TREE

Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

Gini index(impurity measure): It decreases from top to bottom of the tree. The lower the Gini Index the higher the purity of the split.

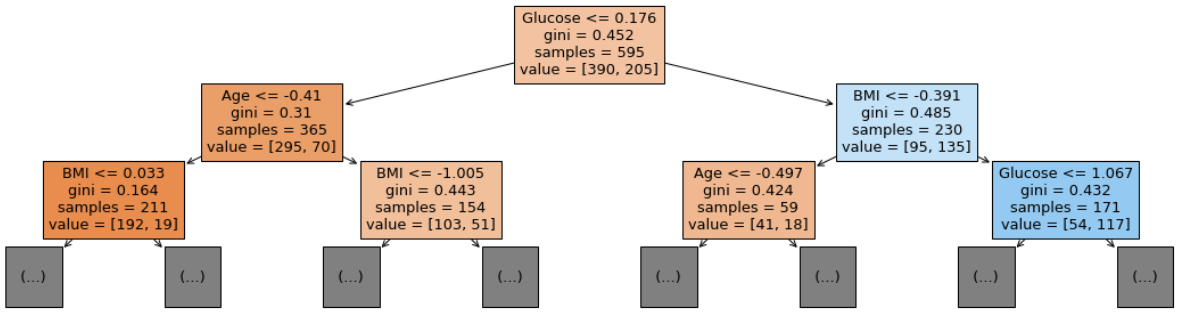
STEPS IN CONTRUCTING A DECISION TREE

- Split data into pure regions
- Start with all samples in a node
- Partition samples recursively based on input to create pure subsets

WHY DECISION TREE?

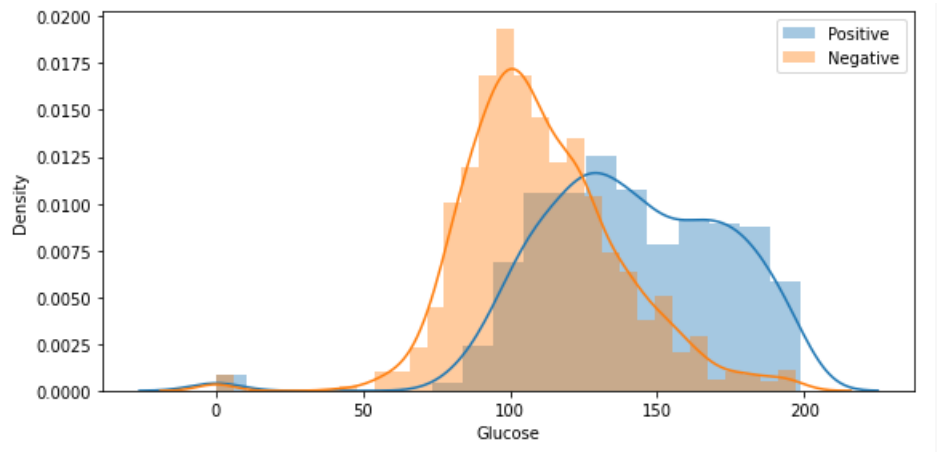
- The construction of decision tree classifier does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery.
- Decision trees can handle high dimensional data.
- In general, decision trees classifier have good accuracy.
- **They are widely preferred in medical field since apart from predicting if the patient is suffering from an ailment or not, doctors also prefer to know the key parameters that determines the decision.** Hence, through an illustration of a decision tree, they can analyse which feature is how much important.

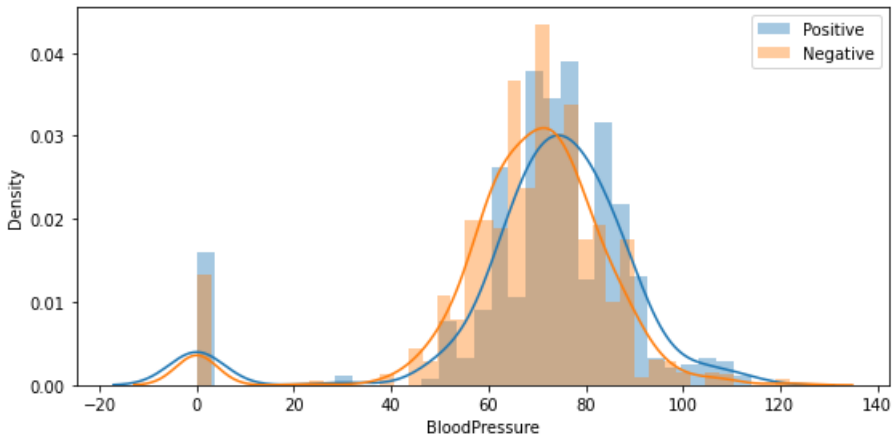
```
plt.figure(figsize=(20,5))
plot_tree(clf,feature_names=df.drop('Outcome',axis=1).columns,max_depth=2, filled=True)
plt.show()
```

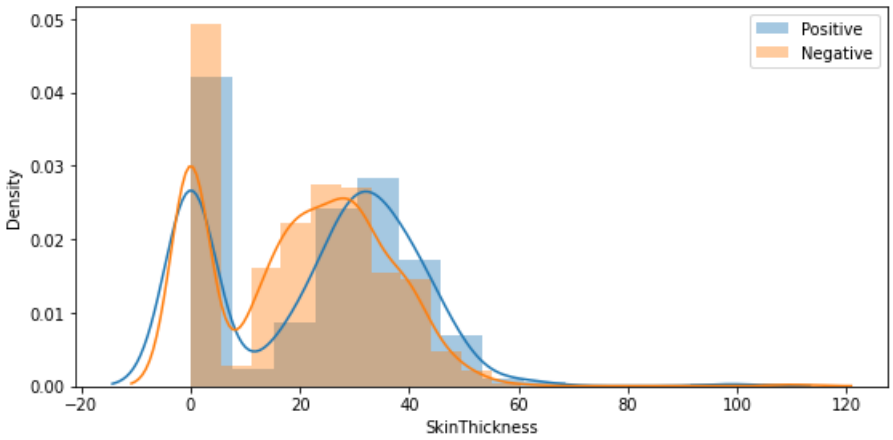


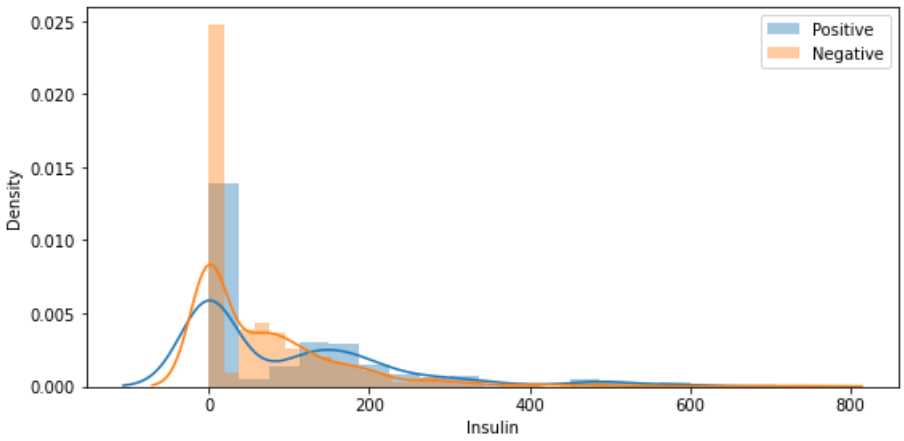
DATA VISUALISATION

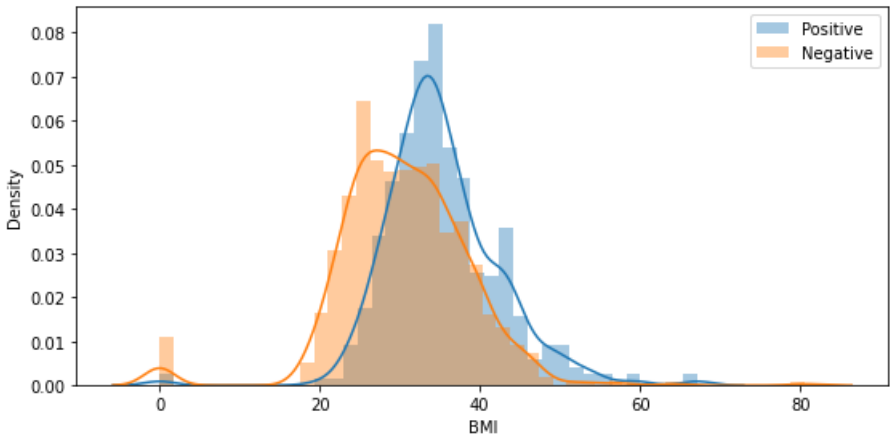
```
fig=plt.figure(figsize=(20,20))
for i,col in enumerate(df.drop(['Pregnancies','Outcome'],axis=1)):
    ax=fig.add_subplot(4,2,i+1)
    ax1=sns.distplot(df[col][df['Outcome']==1],label='Positive')
    sns.distplot(df[col][df['Outcome']==0],label='Negative',ax=ax1)
    plt.legend()
```

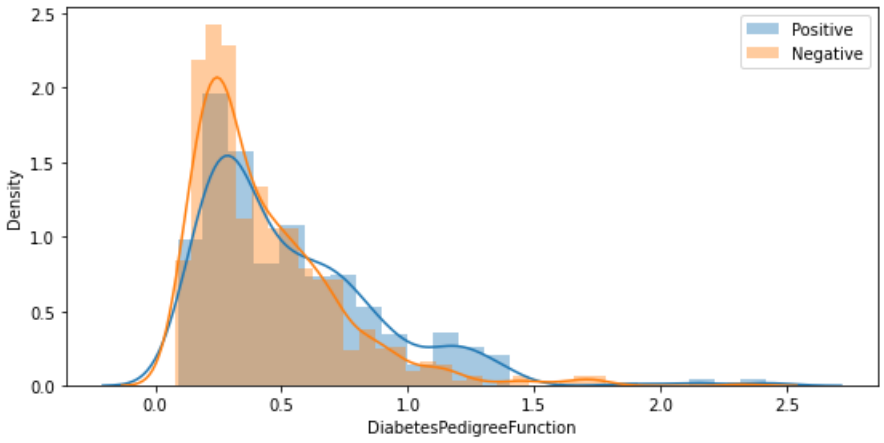


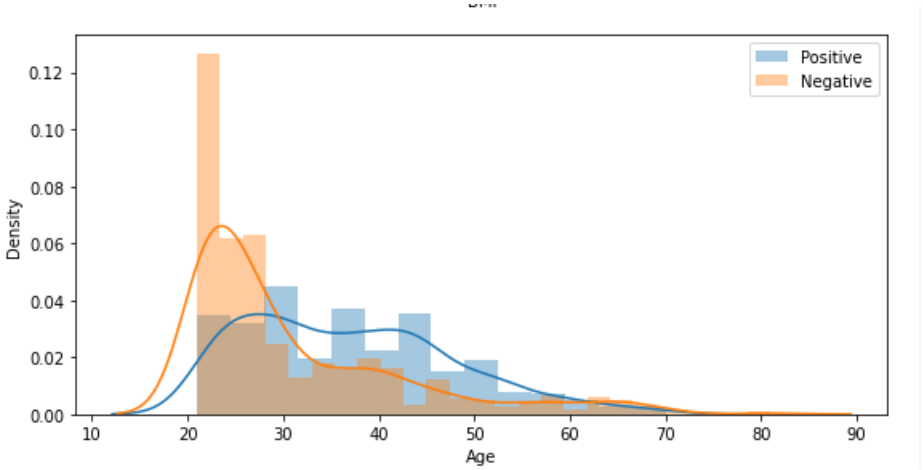





Positive
Negative
Density
Insulin
0.025
0.020
0.015
0.010
0.005
0.000
0
200
400
600
800


Positive
Negative
Density
BMI
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
0
20
40
60
80


Positive
Negative
Density
DiabetesPedigreeFunction
2.5
2.0
1.5
1.0
0.5
0.0
0.0
0.5
1.0
1.5
2.0
2.5



OBSERVATIONS

1. Persons having Glucose approxly in range between 120 to 200 have high chances of having Positive Outcome.

2. People having BloodPressure approx in range 70 to 120 have high chance of having Positive Outcome.

3. Persons having SkinThickness more then 30 -35 have high chances of having Positive Outcome.

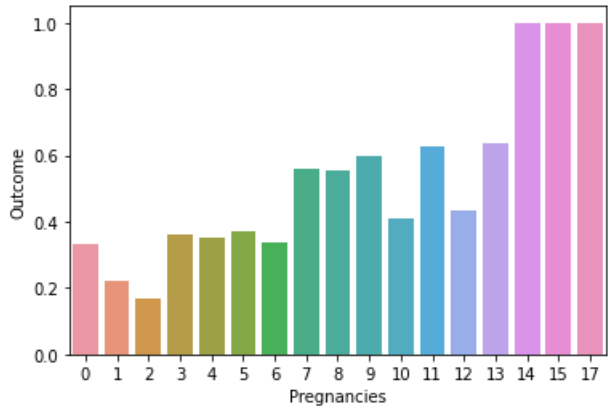
4. If a Person is having very low or high Insulin have high risk of Positive Outcome.

5. Person with BMI more then 30-35 have high chances of having Diabetes.

6. After age of 30 peoples usually have high chances of having Diabetes.

```
sns.barplot(x='Pregnancies',y='Outcome',data=df,ci=None)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4d87889290>
```

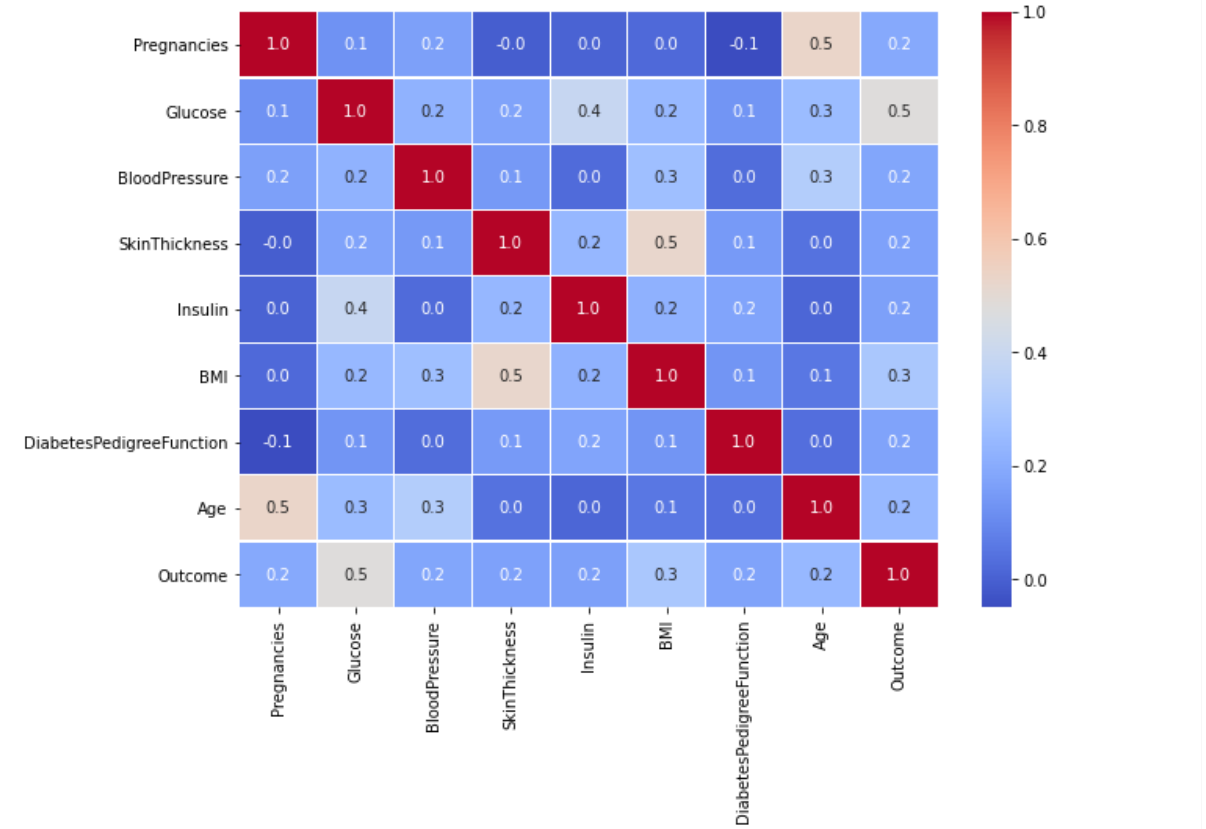


OBSERVATIONS

Graph clearly indicates that high number of Pregnancies have high risk of having Diabe tes or Positive Outcome

```
plt.figure(figsize=(10, 7))

sns.heatmap(df.corr(), annot=True, linewidths=0.2, fmt='.1f', cmap='coolwarm') # cmap='RdYlBu'
plt.show()
```



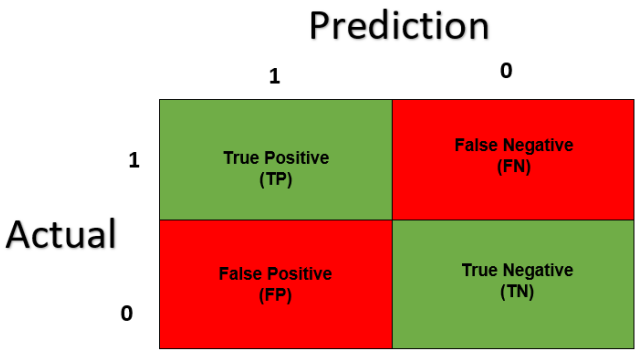
OBSERVATIONS

We observe strong correlation between:

1. Glucose – Outcome
2. SkinThickness – BMI
3. Age – Pregnancies
4. Glucose - Insulin

ANALYSIS OF RESULT

CONFUSION MATRIX



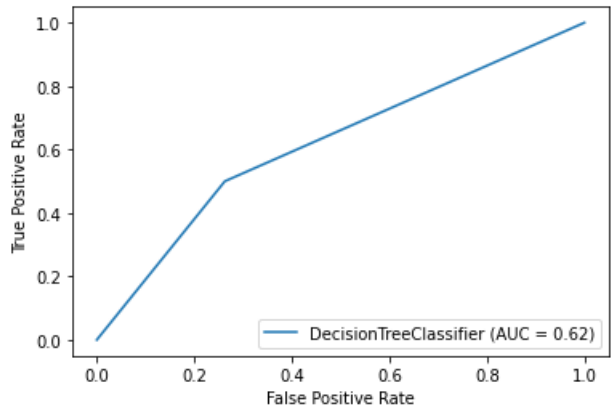
Traning Percentage	Train Samples	Test Samples	True Positives	True Negatives	False Positives	False Negatives
40	446	298	53	153	45	47
30	520	224	36	119	31	38
20	595	149	31	81	20	17

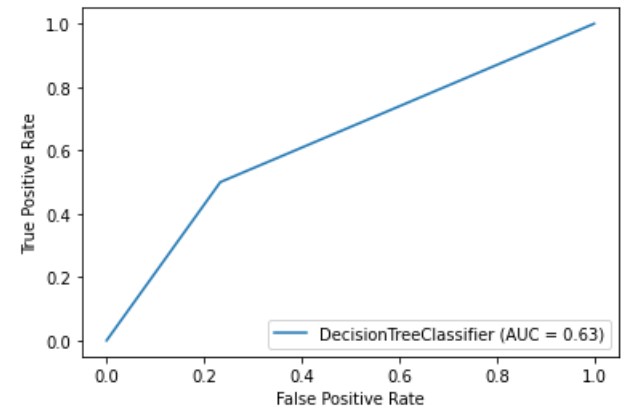
RECIEVER OPERATING CHARACTERISTIC(ROC) CURVE

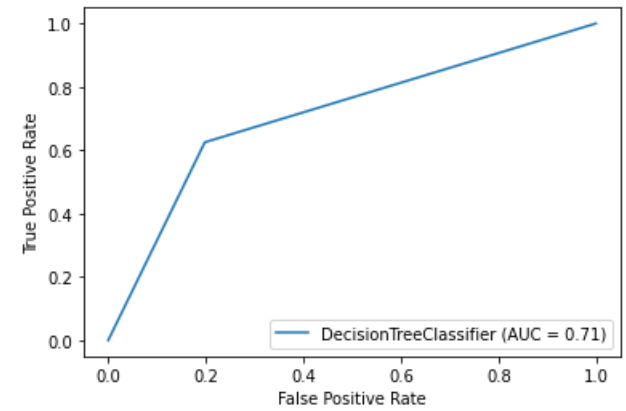
ROC is a probability curve. The ROC curve is plotted with TPR against the FPR where TPR is on the y-axis and FPR is on the x-axis.

AUC represents the degree or measure of separability. Higher the AUC, the better the model is at predicting 0s as 0s and 1s as 1s.

Both the parameters tell us how much the model is capable of distinguishing between classes







We can see that that the ROC curve shows better results in the **third set i.e 80:20 train:test ratio** since the tip of the graph lies more towards the True Positive Rate and less towards the False Positive Rate.

Also, the AUC denoted in the third condition is also seen to be the maximum.

EVALUATING PARAMETERS

Accuracy	Precison	Recall	F1Measure	Sensitivity	Specificity	FPR	FNR	NPV	FDR	MCC
0.691275	0.540816	0.530000	0.535354	0.530000	0.772727	0.227273	0.470000	0.765000	0.459184	0.304268
0.691964	0.537313	0.486486	0.510638	0.486486	0.793333	0.206667	0.513514	0.757962	0.462687	0.287444
0.751678	0.607843	0.645833	0.626263	0.645833	0.801980	0.198020	0.354167	0.826531	0.392157	0.441042

CONCLUSION

Using Decision Trees, we are able to predict diabetes with an accuracy of 75.16%.

FUTURE WORK

More preprocessing techniques and advanced models such as XG Boost, Deep Learning, etc. can be used to increase the accuracy and other evaluating parameters. This will help the doctors make an informed decision regarding diabetes.