A
Comprehensive Analysis
and
Performance Comparison
of Multiple Machine Learning Models
for Classification

Project Overview:

- The objective of this project is to perform in depth analysis and develop multiple Machine Learning Classification Model to classify whether someone has diabetes or not.
- Performance Comparision of Multiple ML Model based on Accuracy Score, Precision Score, Recall Score, F1-Score, Confusion Matrix.

Data overview

- Dataset consists of several Medical Variables(Independent) and one Outcome Variable(Dependent)
- The independent variables in this data set are :-

- Pregnancies :- Number of times a woman has been pregnant
- Glucose :- Plasma Glucose concentration of 2 hours in an oral glucose tolerance test
- 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 (doesn't have diabetes) or 1 (has diabetes)

Data overview

- Data set Contains 768 data instances.
- 6 Independent Variables are of int64 datatype
- 2 Independent Variables are of float64 datatype
- Target Variables is of int64 datatype
- Target variable has data imbalancy:
- - 65% data belongs to non-diabetic class
- - 35% data belongs to diabetic class

Checking Brief Info of DataFrame

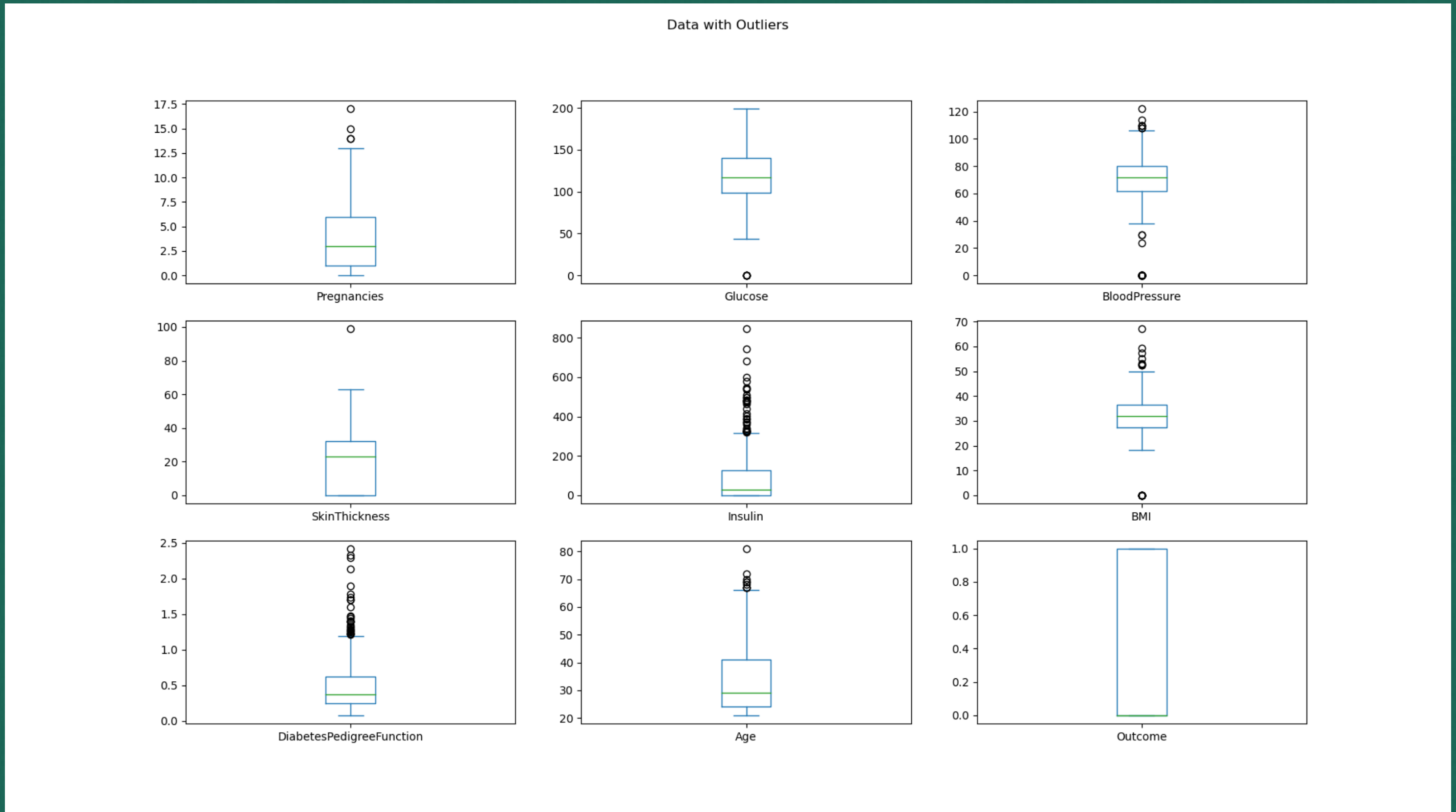
In [65]:

```
1  df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Pregnancies               768 non-null    int64  
 1   Glucose                   768 non-null    int64  
 2   BloodPressure             768 non-null    int64  
 3   SkinThickness             768 non-null    int64  
 4   Insulin                   768 non-null    int64  
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64  
 8   Outcome                   768 non-null    int64  
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

Data Health

- Data Frame is devoid of any Missing values and Duplicated Rows.
- Data Frame has few Outliers associated with some of the dependent variables



Checking for Duplicated Rows

```
In [9]:  Dup_Rows = df[df.duplicated()]
         Dup_Rows.shape

Out[9]: (0, 9)
```

Checking for Missing Values

```
In [64]:  df.isnull().sum()

Out[64]: Pregnancies                 0
         Glucose                     0
         BloodPressure               0
         SkinThickness               0
         Insulin                     0
         BMI                         0
         DiabetesPedigreeFunction    0
         Age                         0
         Outcome                     0
         dtype: int64
```

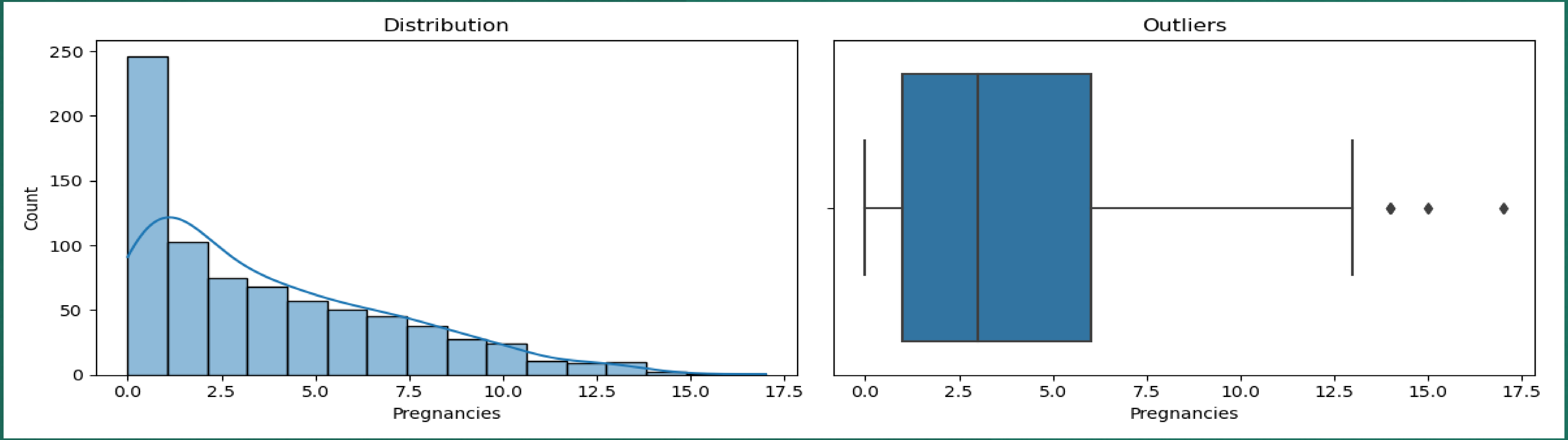
UNIVARIATE ANALYSIS :

Pregnancies : Number of times being pregnant

- Data is varying from 0 time of being pregnant to 17 times being pregnant
- Performed Feature Engineering , Categorized the data on the basis of number of pregnancies
- Data is categorized in Low | High | Very-High band

  - Low 45%
  - Very High 29%
  - High 26%

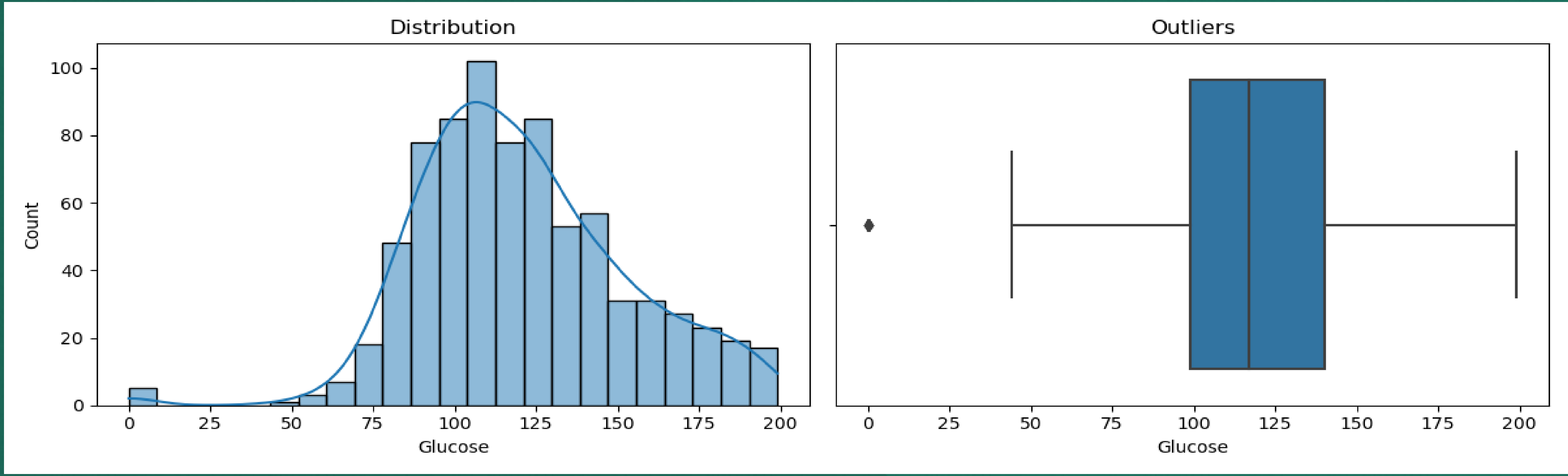
** Data has Outliers, Distribution is skewed towards right.



UNIVARIATE ANALYSIS :

Glucose : Plasma glucose concentration a 2 hours in an oral glucose tolerance test

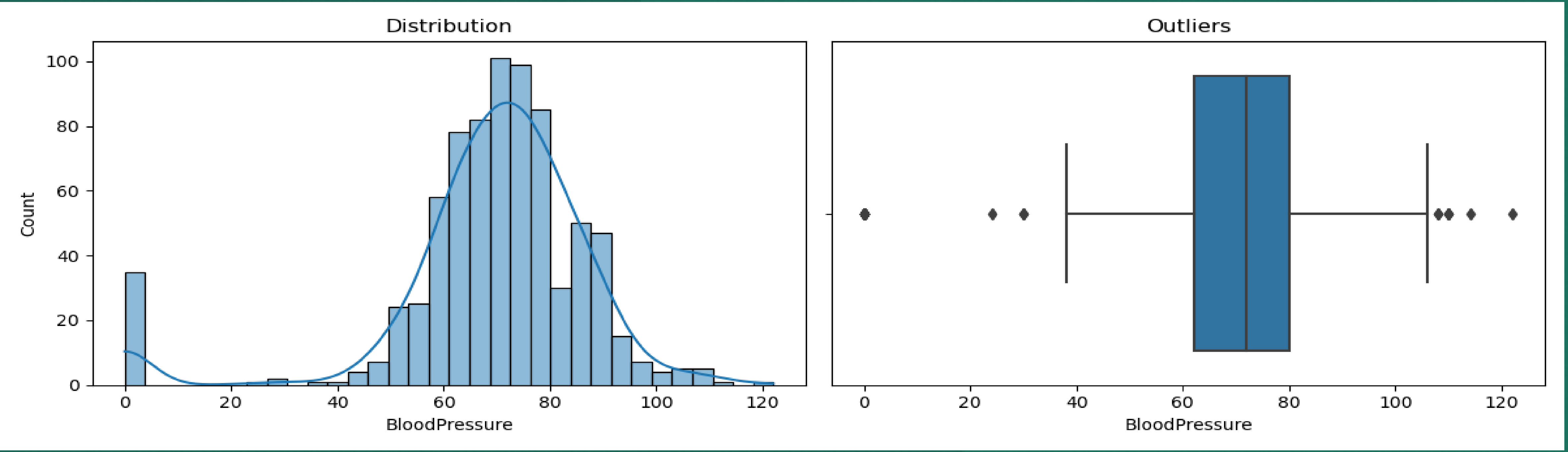
- Data has Skewness Coefficient of: 0.18 (It is not Skewed)
- Distribution is skewed towards left.
- Skewness is because of Outliers(Data does not required any transformation)
- Most of the Population has glucose concentration between 90 - 120



UNIVARIATE ANALYSIS :

Blood Pressure : Diastolic blood pressure (mm Hg)

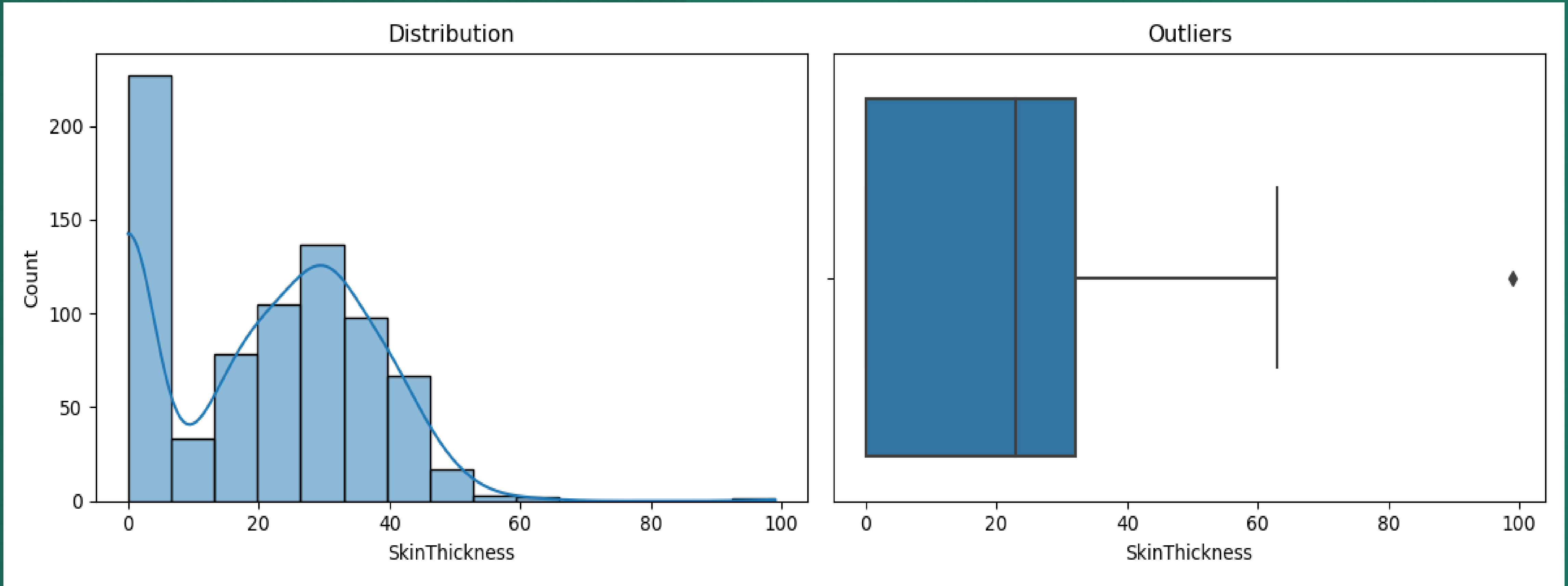
- Blood Pressure has Skewness Coefficient of: -1.8 (Data seems Skewed to the left)
- Distribution is Very much symmetrical, data has outliers.
- Skewness is because of Outliers (Data does not require any transformation)
- Most of the Population has Blood Pressure between 60 mm-hg - 80 mm-hg



UNIVARIATE ANALYSIS :

Skin Thickness : Triceps skin fold thickness (mm)

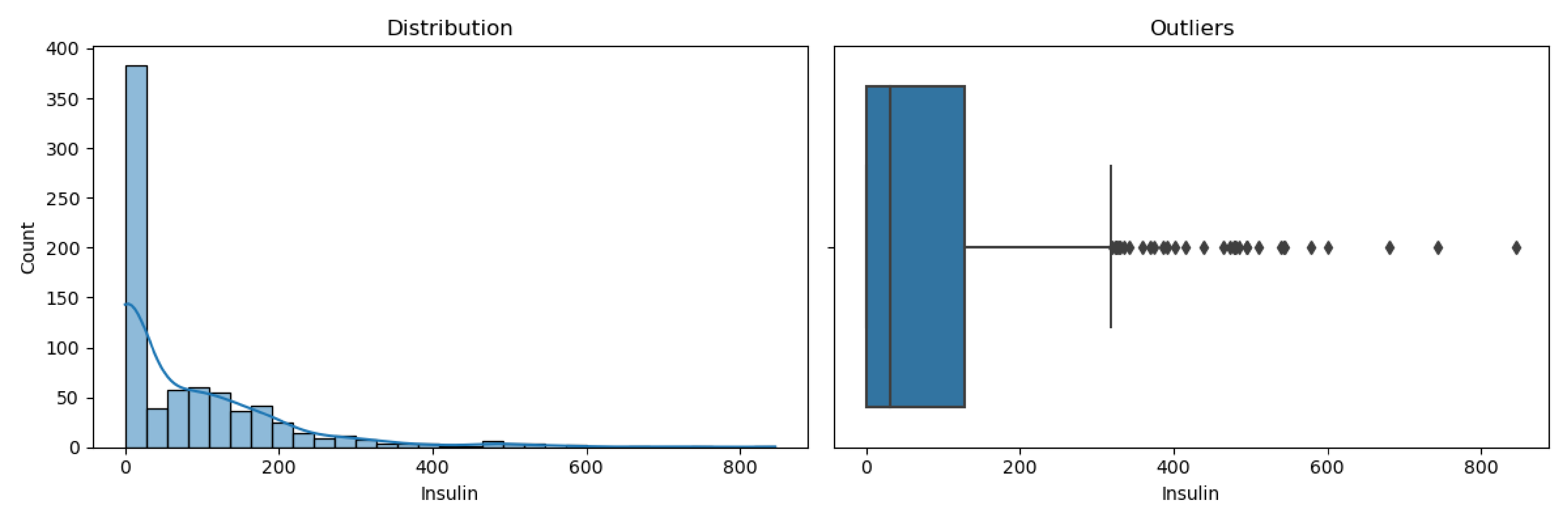
- Skin Thickness has Skewness Coefficient of: 0.1 (Data is not skewed)
- Skewness is because of Outliers (Data does not require any transformation)
- Most of the Population has Skin fold thickness between 20 mm - 40 mm



UNIVARIATE ANALYSIS :

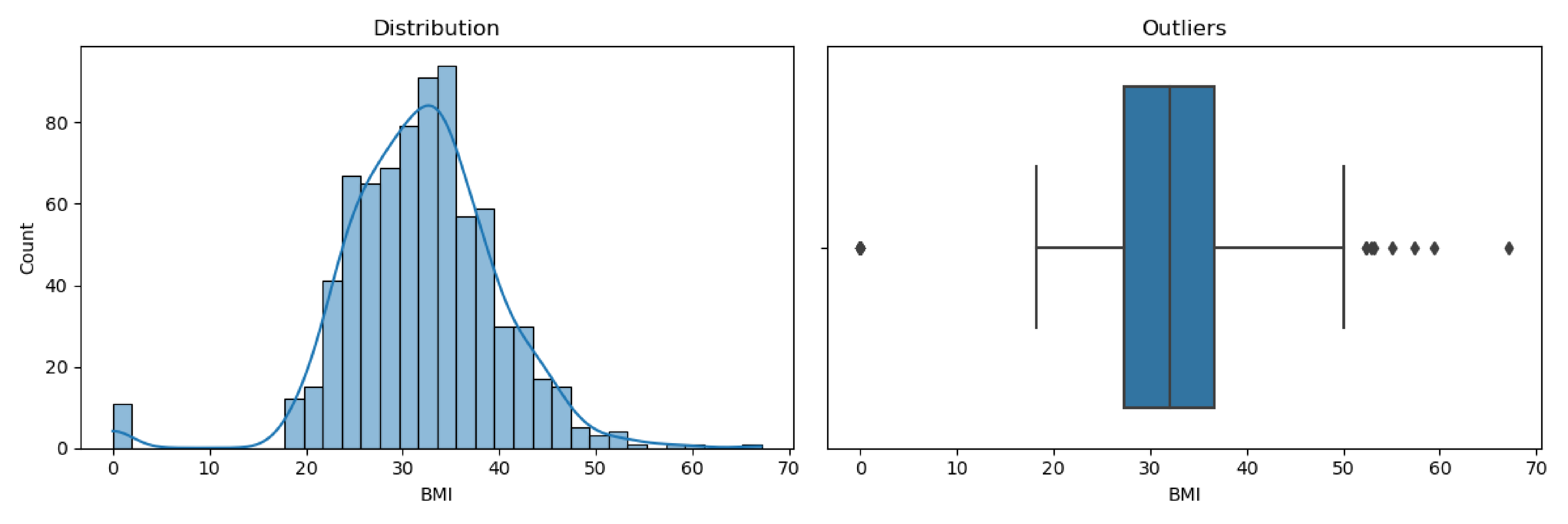
Insulin : 2-Hour serum insulin (mu U/ml)

- Insulin has Skewness Coefficient of: 2.2 (Data is heavily skewed toward right)
- Skewness is because of Outliers (Data does not require any transformation)
- Most of the Population has Insulin level between 50(mu U/ml) - 125(mu U/ml)



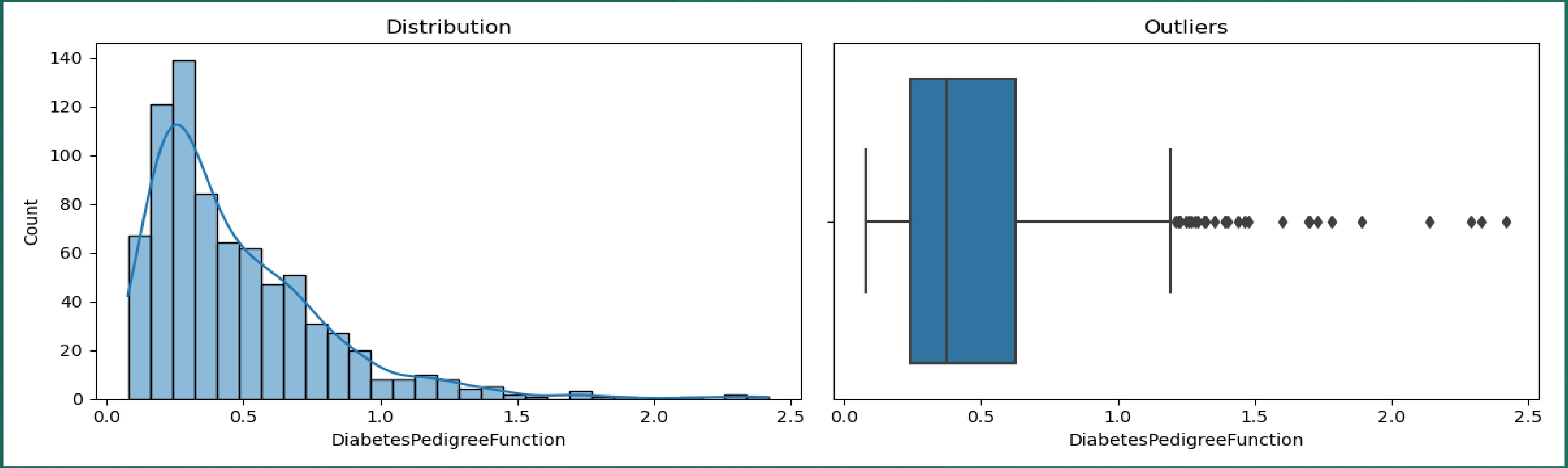
UNIVARIATE ANALYSIS :
BMI : Body Mass Index

- BMI has Skewness Coefficient of: -0.4
- Skewness is because of Outliers (Data does not require any transformation)
- Most of the Population has BMI ratio between 25 - 40 (wt. in kg/(height in m)^2)



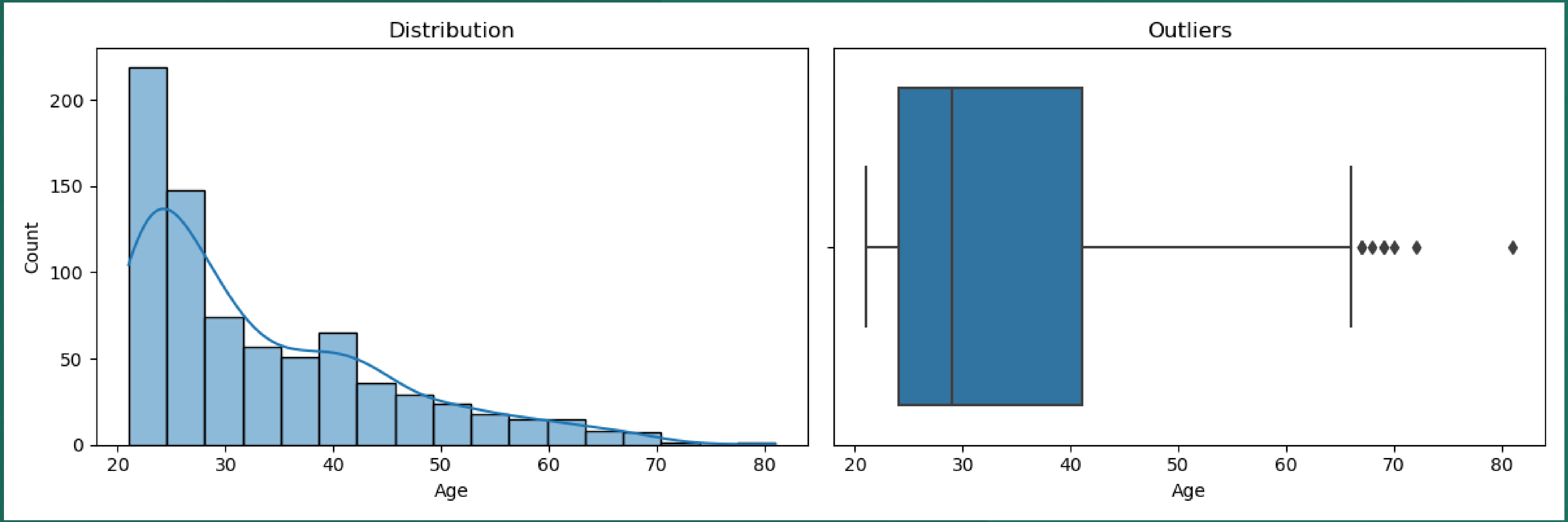
UNIVARIATE ANALYSIS :
Diabetes Pedigree Function

- Diabetes Pedigree Function has Skewness Coefficient of: 1.9
- Distribution is Skewed towards right.
- Skewness is because of Outliers (Data does not require any transformation)
- Most of the Population has Diabetes pedigree function between 20 - 40



UNIVARIATE ANALYSIS :
Age (years)

- Age has Skewness Coefficient of: 1.1
- Distribution is skewed towards right
- Skewness is because of Outliers (Data does not require any transformation)
- Most of the Population has Age between 20 - 30 Years



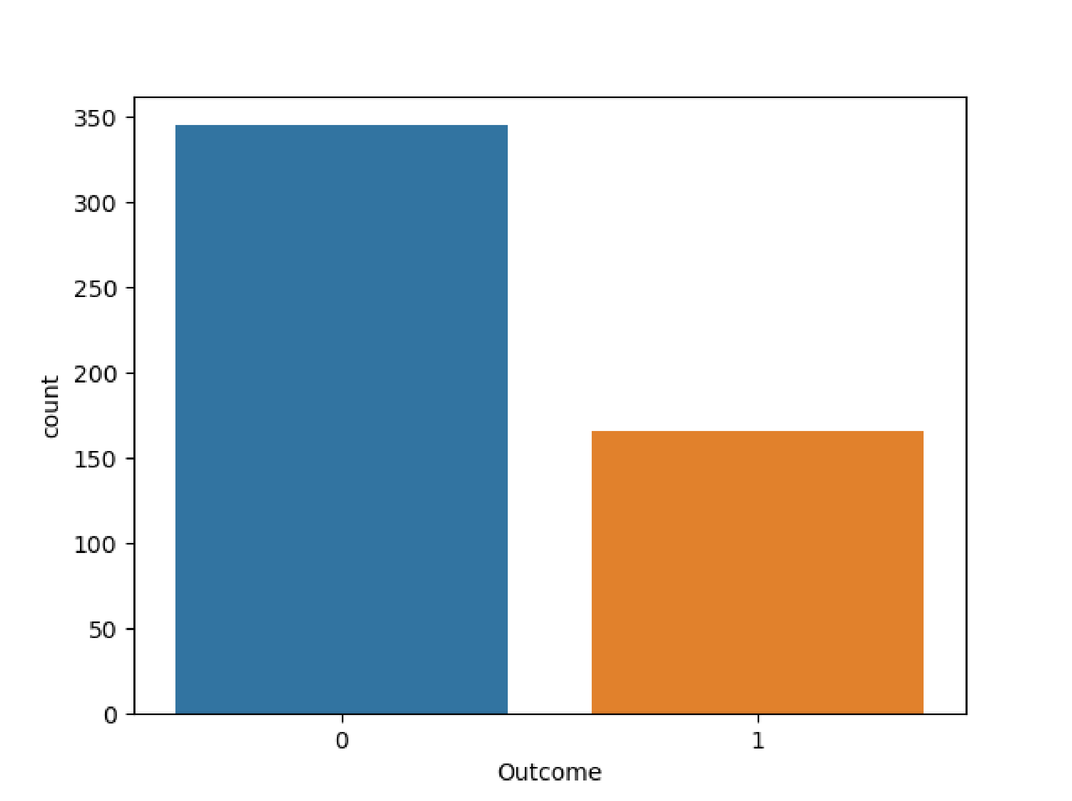
UNIVARIATE ANALYSIS :

Outcome: Target Variable

- Data has Imbalancy
  - 65% of Target data belongs to Class 0 (Non Diabetic)
  - 35% of Target data belongs to Class 1 (Diabetic)
- Data Imbalancy would lead to biased learning (I am Using SMOTE technique to bring balancy in the data)

```
In [29]:   1  df['Outcome'].value_counts(normalize = True)*100

Out[29]: 0    65.104167
         1    34.895833
         Name: Outcome, dtype: float64
```



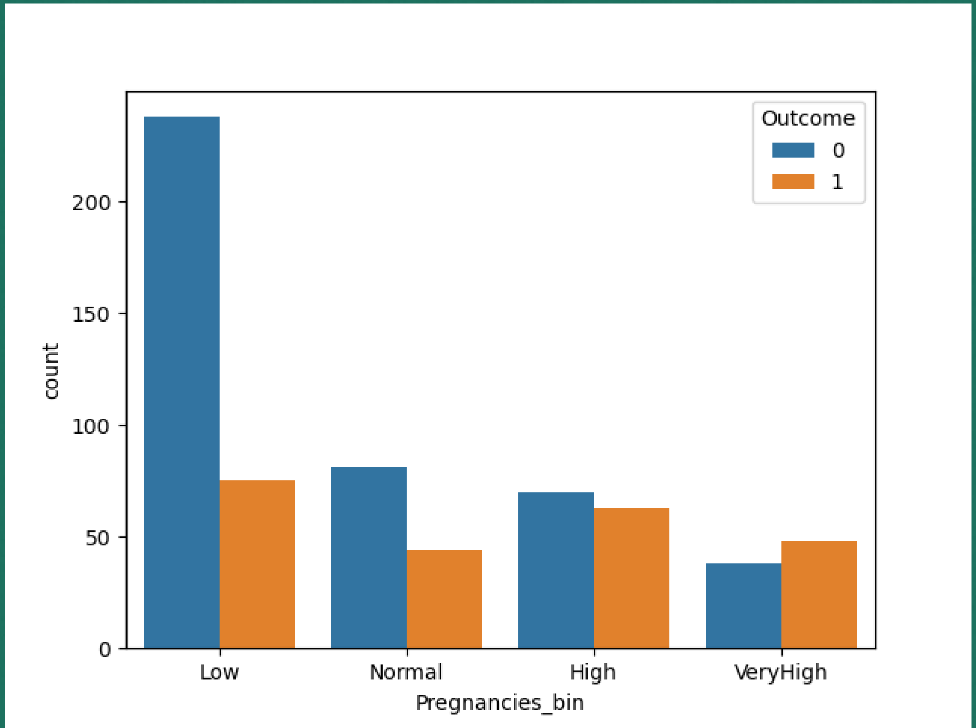
BI-VARIATE ANALYSYS

- I have performed ANOVA Test and Point Biserial Coefficient Test to Analyse the relationship among my Target Variable (Outcome) and others independent variables.
- I have perform the Hypothesis Testing to establish the relationship among Target and Independent Variables.
- Null Hypothesis - H(0) : There is no any statistically significant relationship among Target and Predictor variables
- Alternate Hypothesis - H(A): There is statistically significant relationship among Target and Predictor variables
- Here in this Project, I am taking level of Significance (alpha = 0.05).

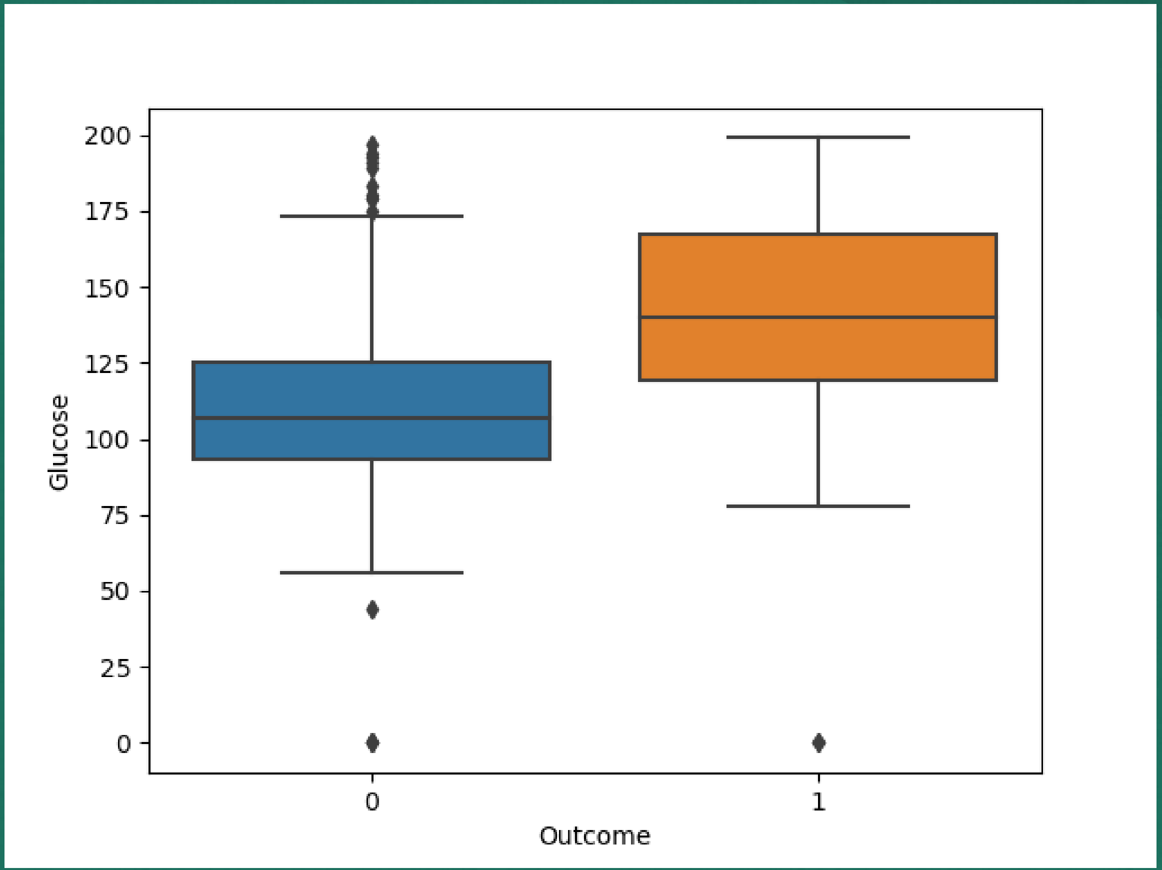
  I will reject the Null hypothesis when p-value is less than alpha.

BI- VARIATE ANALYSIS:
Outcome Vs Pregnancies

- Anova-Test :
  - F_oneway Result (statistic=810.5150593469092, p-value=1.7236475242343698e-143)
  - High F-static Score suggests strong relation, in this case, F-static score is low
- Point Biserial-Test :
  - SignificanceResult(statistic=0.22189815303398686, p-value=5.065127298053635e-10)
  - Score of F-static close to zero suggest no significant relations, close to 1 or -1 suggests relationship
- Also from the plot, it is evident that there is no significant relations between Pregnancies and Outcome

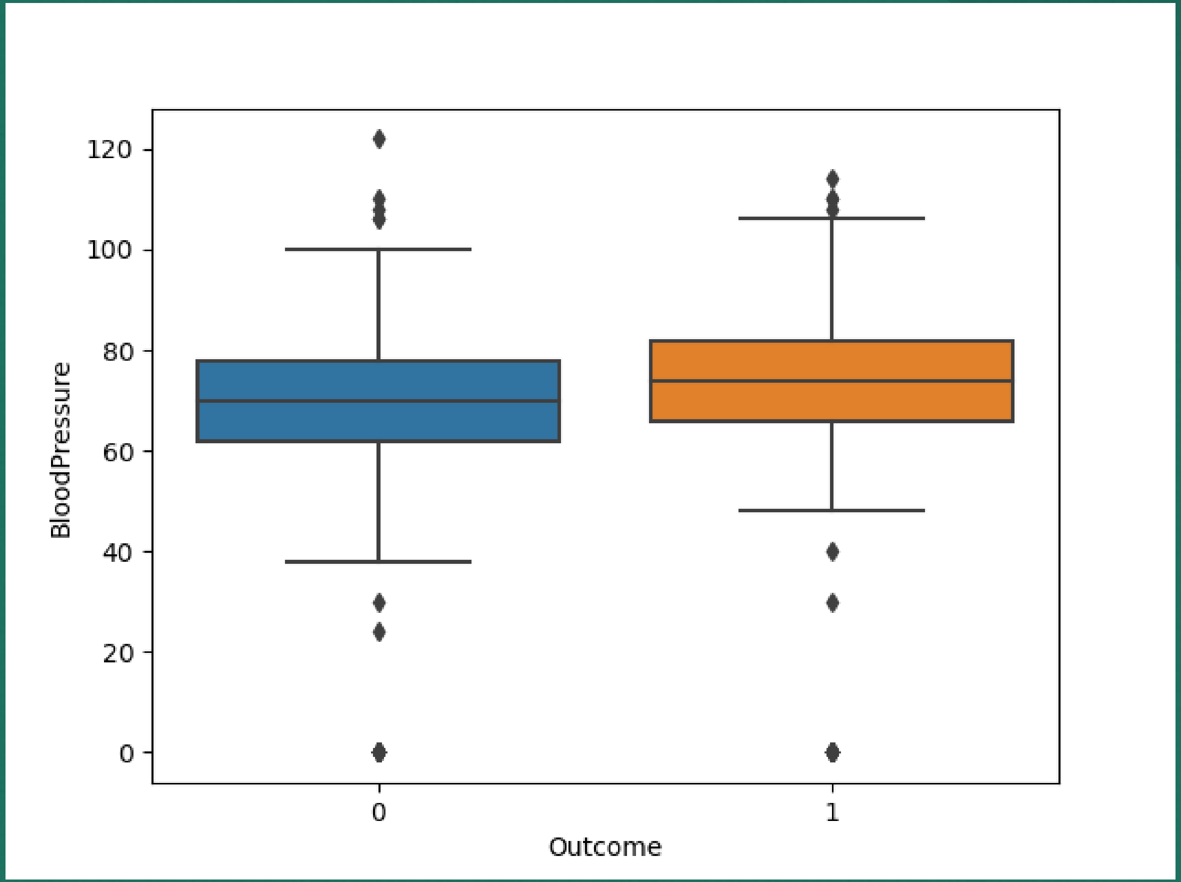


BI- VARIATE ANALYSIS:
Outcome Vs Glucose



- Anova-Test :
  - F_onewayResult(statistic=10914.672630134193,
  - pvalue=0.0
  - High F-static Score suggests strong relation, in this case, F-static score is High
- PointBiserial-Test :
  - Significance Result (statistic=0.466581398306874,
  - pvalue = 8.935431645289576e-43
  - Score of F-static is not close to zero suggest some significant relations
- Also from the plot, it is evident that there is significant difference among the mean of both classes, suggesting relations between Glucose and Outcome

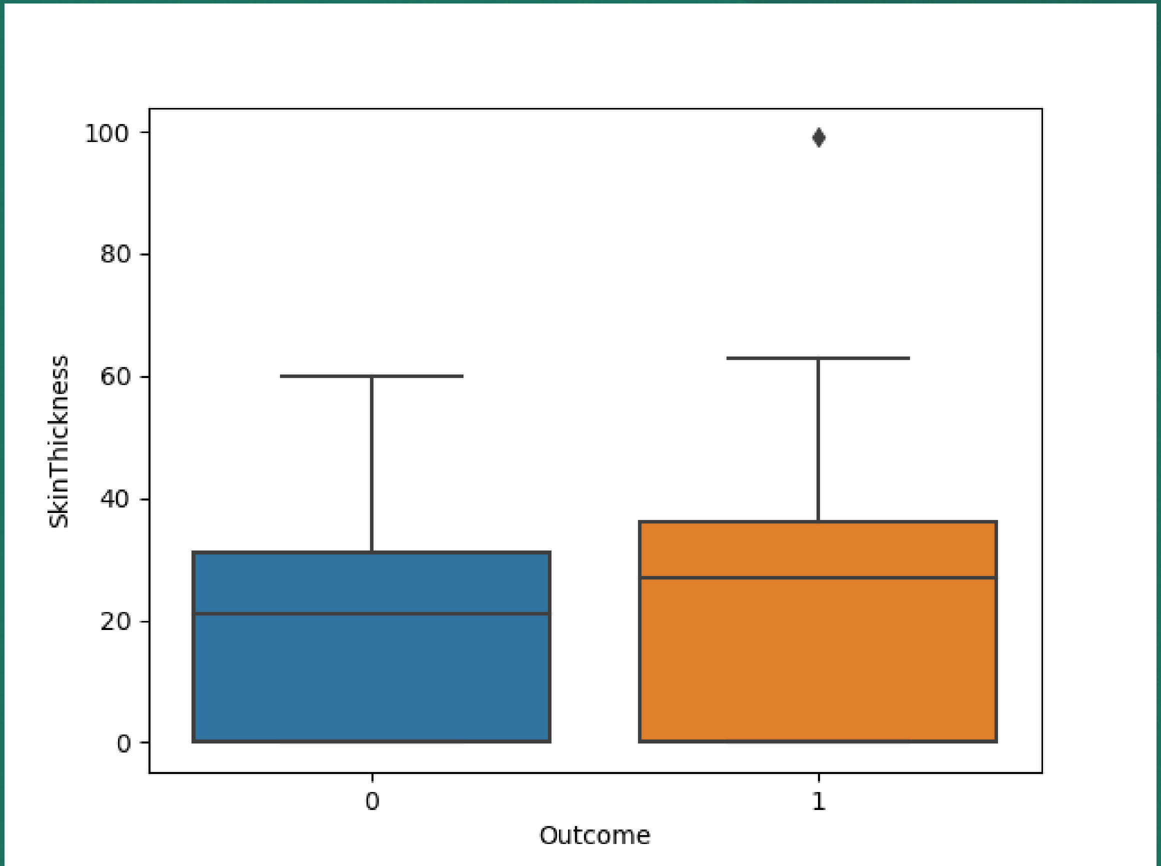
BI- VARIATE ANALYSIS:
Outcome Vs Blood Pressure



- Anova-Test :
  - F_onewayResult(statistic=9685.068133631648,
  - pvalue=0.0)
  - High F-static Score suggests strong relation, in this case,
  - F-static score is considrably High
- PointBiserial-Test :
  - SignificanceResult(statistic=0.06506835955033283, pvalue=0.07151390009776264)
  - Score of F-static is so close to zero suggest no any significant relations

- Also from the plot, it is evident that there is not much significant difference among the mean of both classes, suggesting no relation between BloodPressure and Outcome

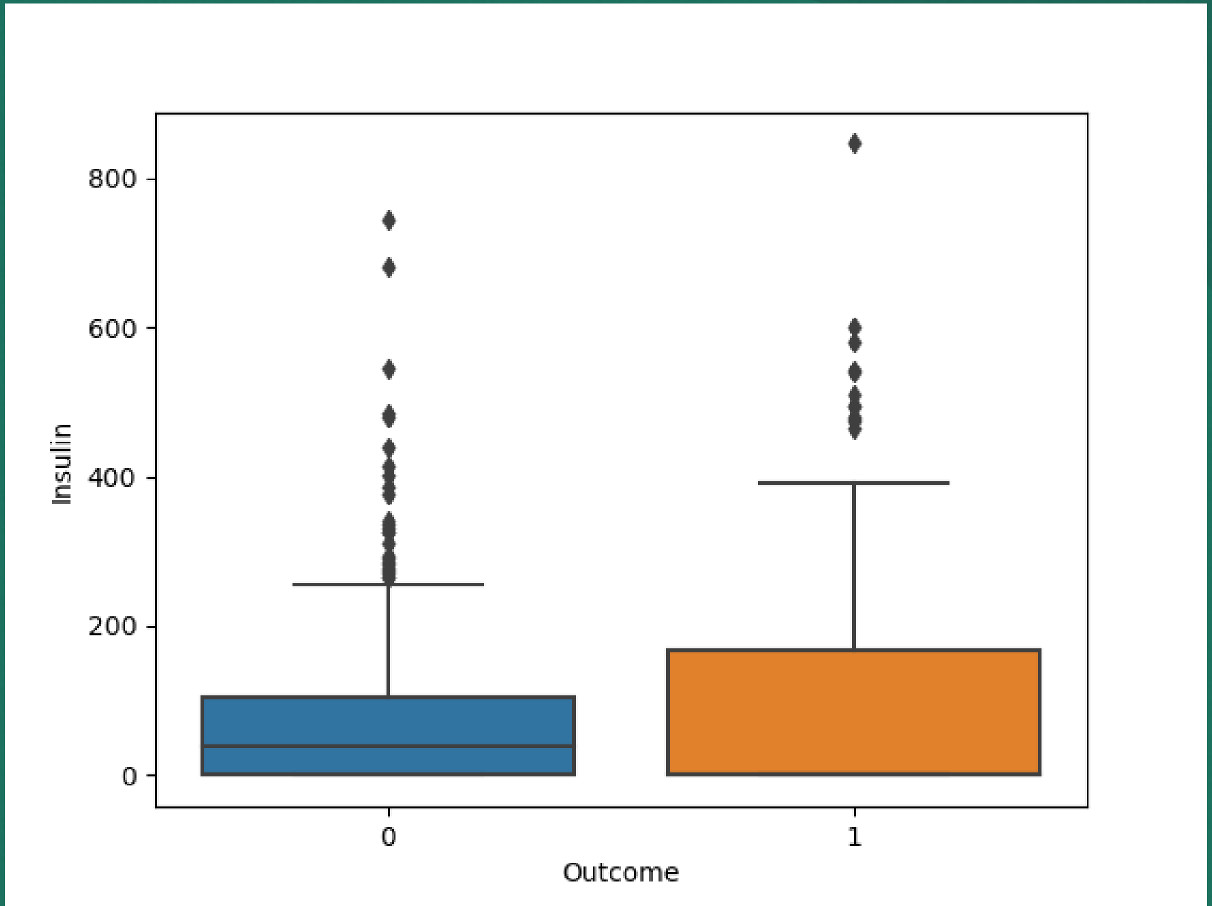
BI- VARIATE ANALYSIS:
Outcome Vs Skin Thickness



- Anova-Test :
  - F_onewayResult(statistic=1228.8421887367274, pvalue=3.1079391612307788e-198)
  - High F-static Score suggests strong relation, in this case,
  - F-static score is not High
- PointBiserial-Test :
  - SignificanceResult(statistic=0.0747522319183194, pvalue=0.038347704820490915)
  - Score of F-static is so close to zero suggest no any significant relations

- Also from the plot, it is evident that there is not much significant difference among the mean of both classes, suggesting no relation between SkinThickness and Outcome

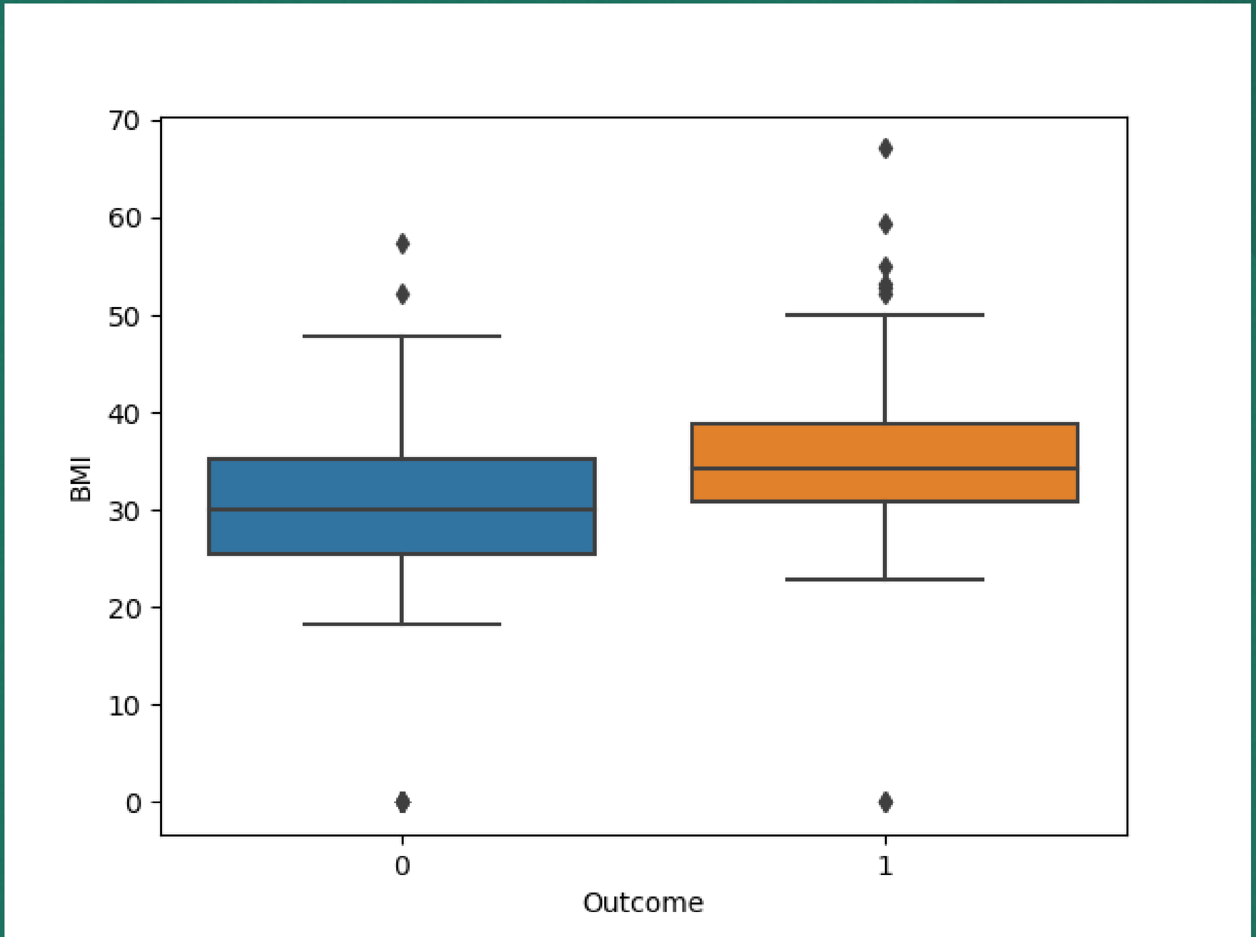
BI- VARIATE ANALYSIS:
Outcome Vs Insulin



Anova-Test :
F_onewayResult(statistic=365.01491713877726, pvalue=3.6531836527047957e-73)
High F-static Score suggests strong relation, in this case, F-static score is not High
PointBiserial-Test :
SignificanceResult(statistic=0.13054795488404775, pvalue=0.0002861864603603164)
Score of F-static is so close to zero suggest no any significant relations

Also from the plot, it is evident that there is not much significant difference among the mean of both classes, suggesting no relation between Insulin and Outcome

BI- VARIATE ANALYSIS:
Outcome Vs BMI

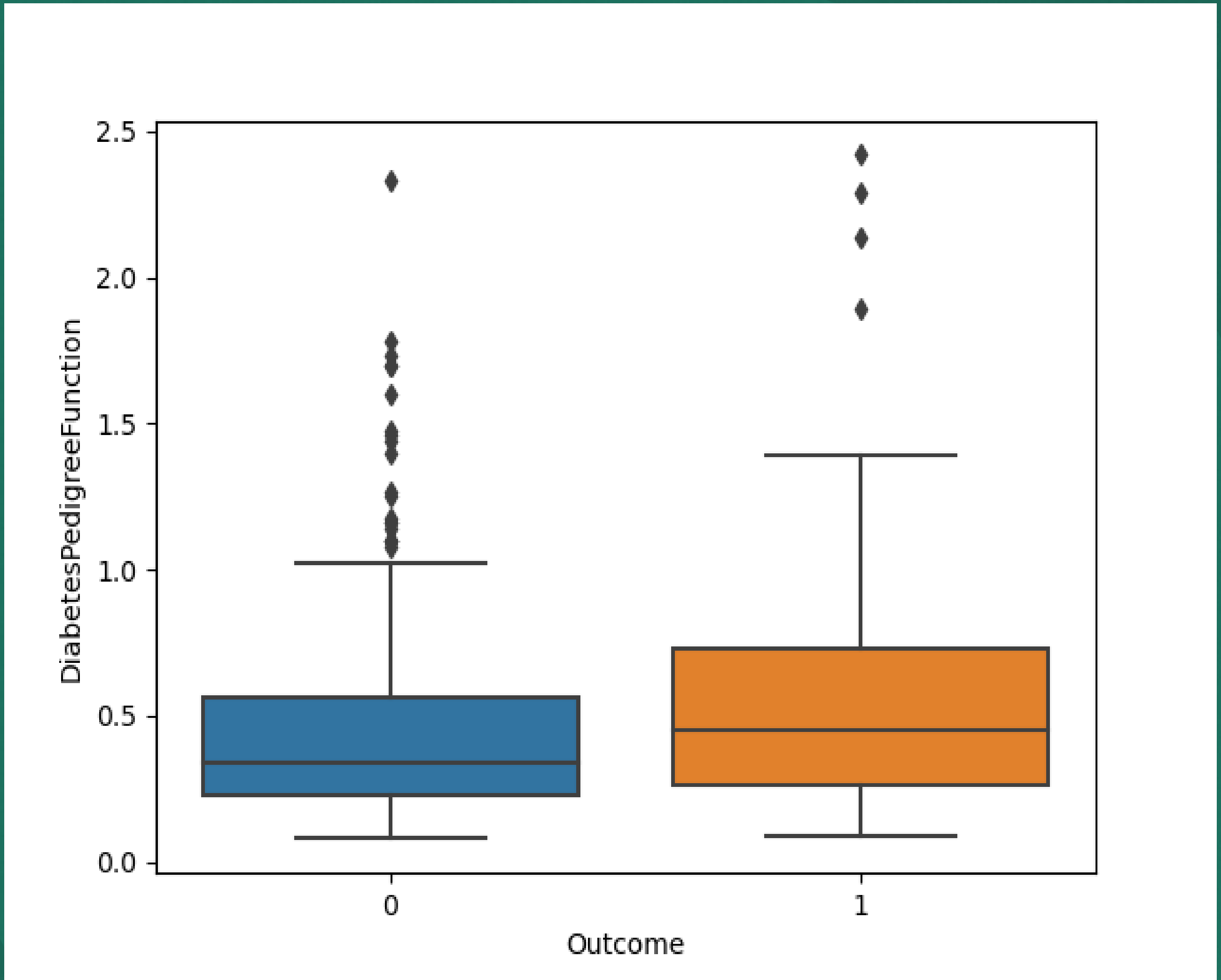


Anova-Test :
F_onewayResult(statistic=12326.397968979043, pvalue=0.0)
High F-static Score suggests strong relation, in this case, F-static score is High

PointBiserial-Test :
SignificanceResult(statistic=0.29269466264444544, pvalue=1.2298074873116917e-16)
Score of F-static is not so close to zero suggest some significant relations

Also from the plot, it is evident that there is not much but some significant difference among the mean of both classes, suggesting some relation between BMI and Outcome

BI- VARIATE ANALYSIS:
Outcome Vs Diabetes Pedigree Function



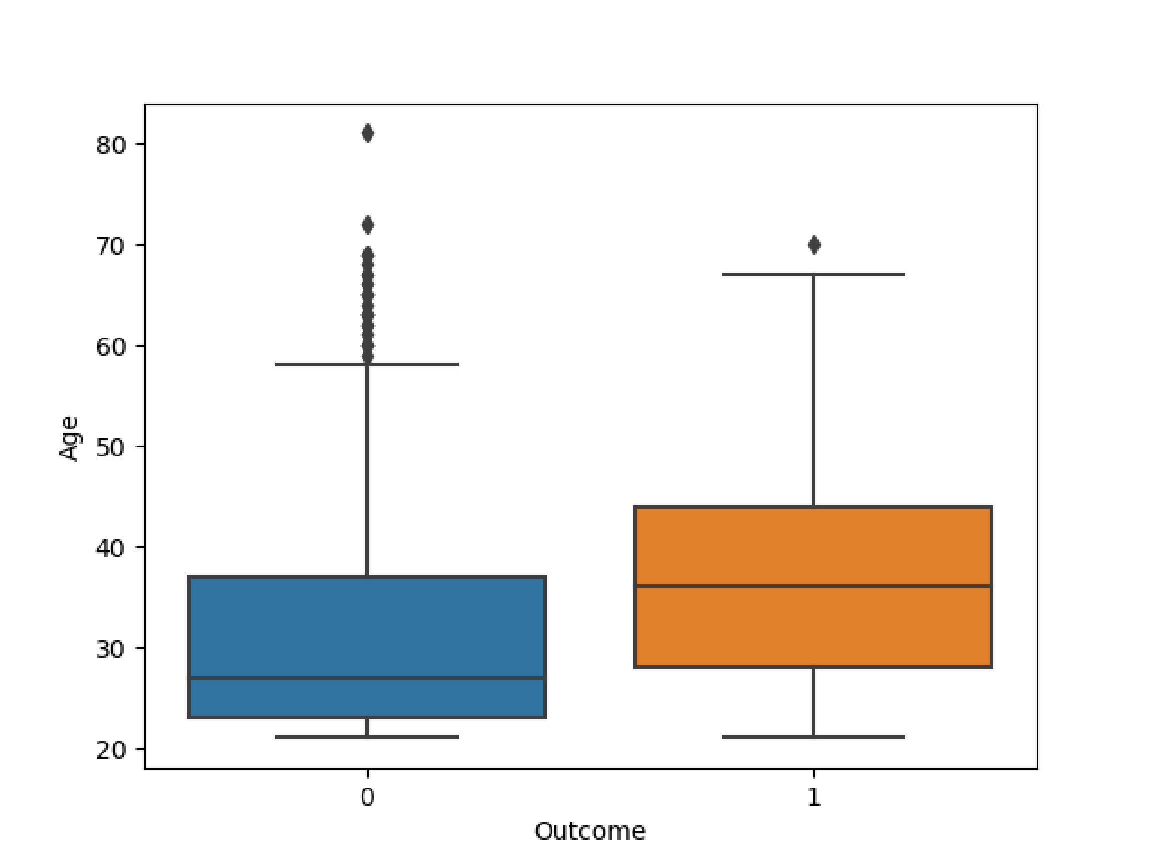
Anova-Test :
F_onewayResult(statistic=34.40531346539221,
pvalue=5.471443802691407e-09)
High F-static Score suggests strong relation, in this case,
F-static score is Low

PointBiserial-Test :
SignificanceResult(statistic=0.17384406565296007,
pvalue=1.2546070101487771e-06)
Score of F-static is close to zero suggest
no significant relations

Also from the plot, it is evident that there is not much significant difference among the mean of both classes, suggesting no relation between Diabetes Pedigree Function and Outcome

BI- VARIATE ANALYSIS:

Outcome Vs Age

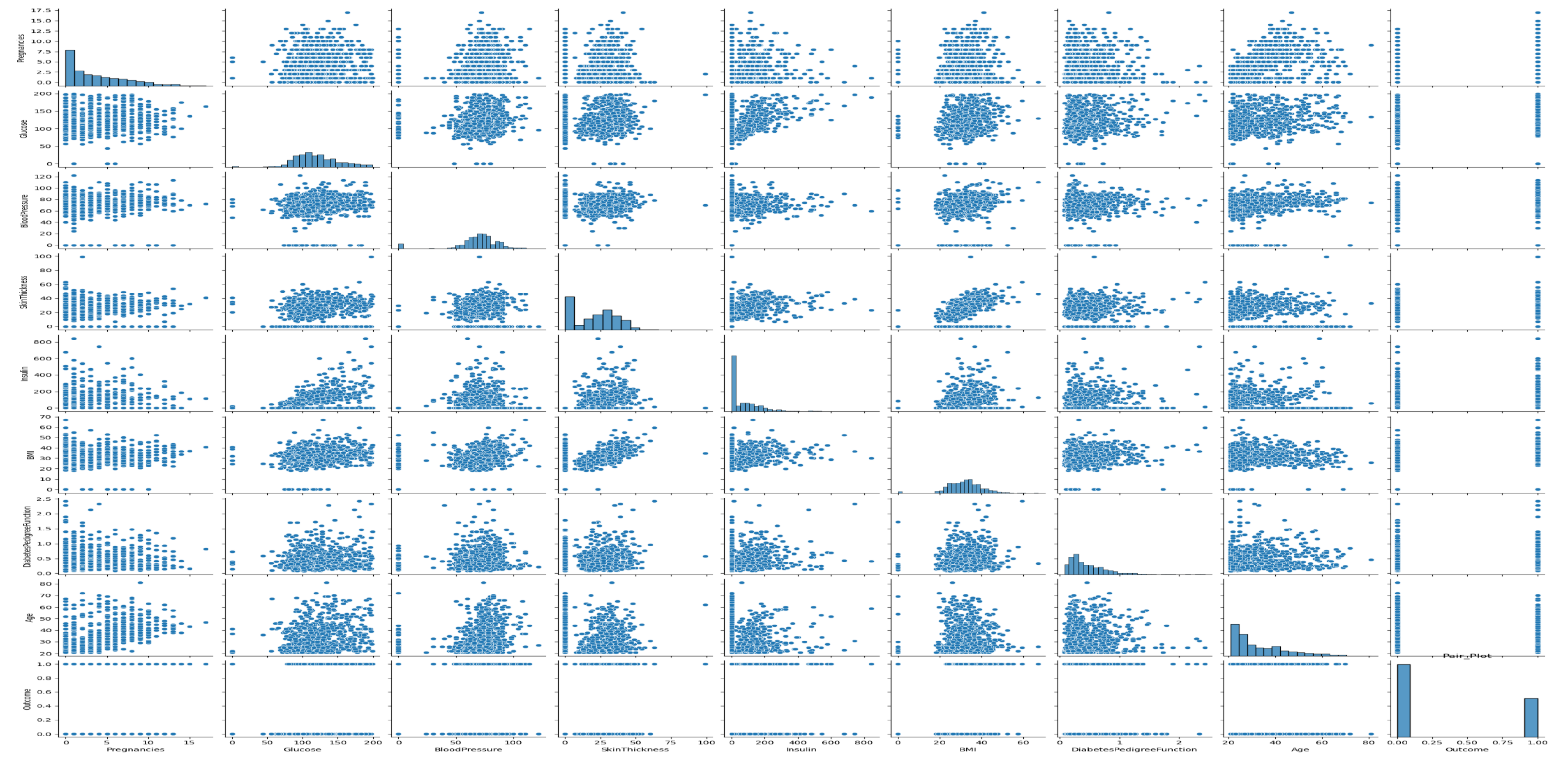


Anova-Test :
F_onewayResult(statistic=5997.832961507344, pvalue=0.0)
High F-static Score suggests strong relation, in this case, F-static score is considerably high

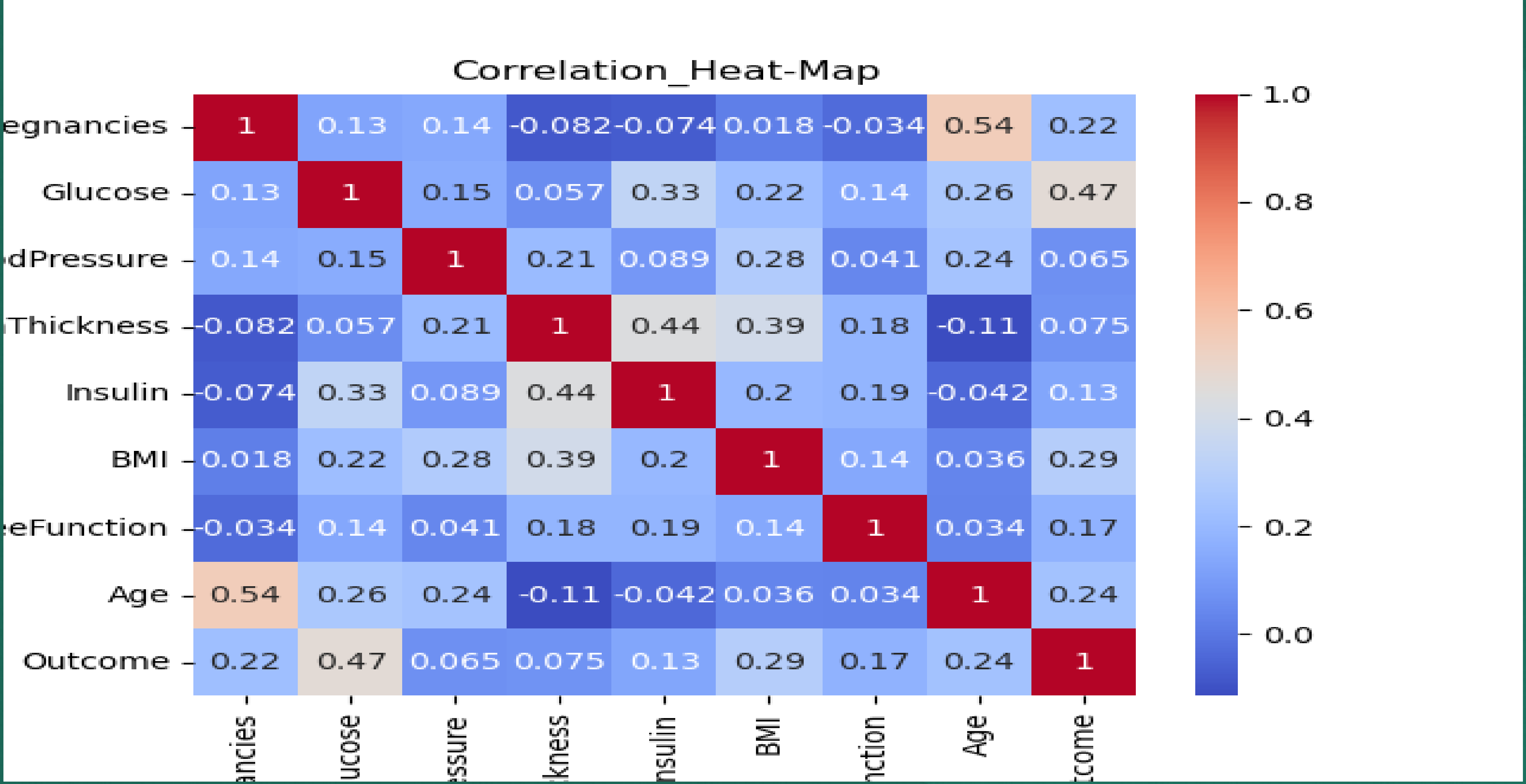
PointBiserial-Test :
SignificanceResult(statistic=0.2383559830271977, pvalue=2.209975460665451e-11)
Score of F-static is not so close to zero suggest some significant relations

Also from the plot, it is evident that there is significant difference among the mean of both classes, suggesting some relation between Age and Outcome

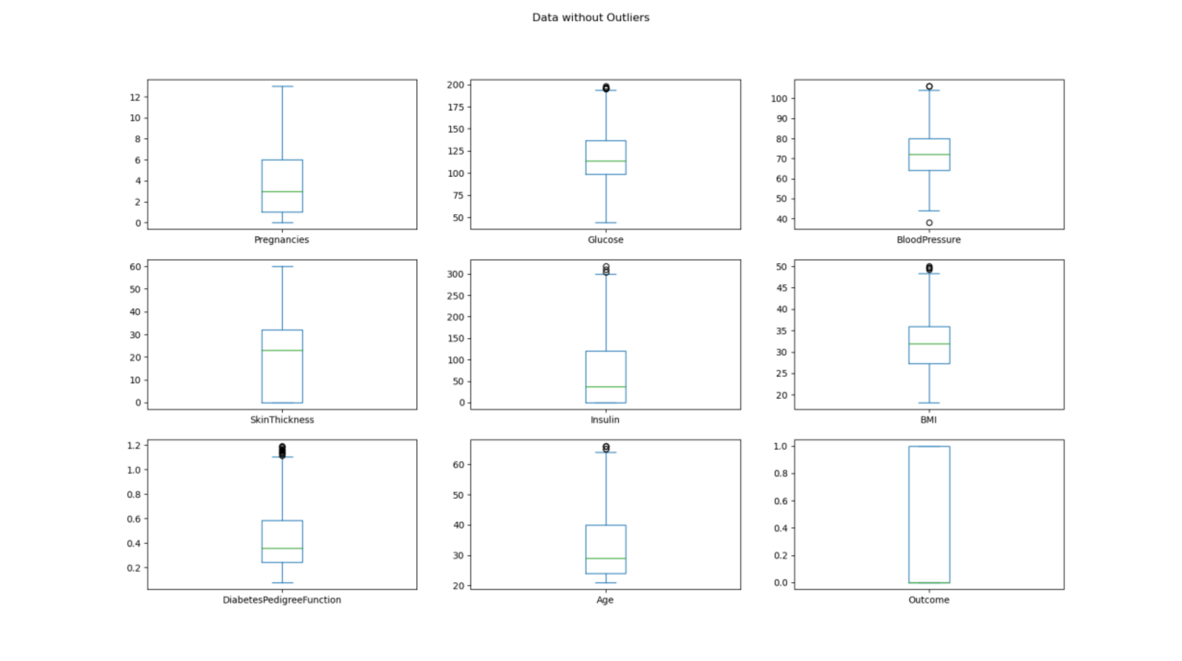
MULTI - VARIATE ANALYSIS:
Pair-Plot

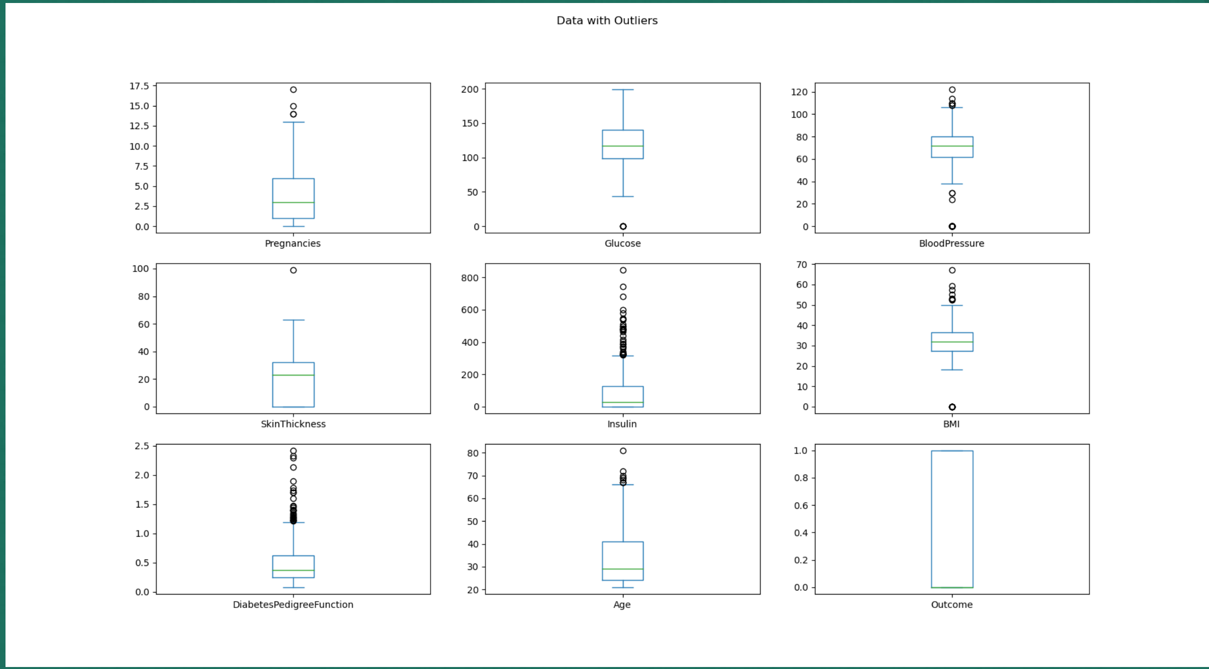


MULTI - VARIATE ANALYSIS:
HeatMap – Correlation Coefficient

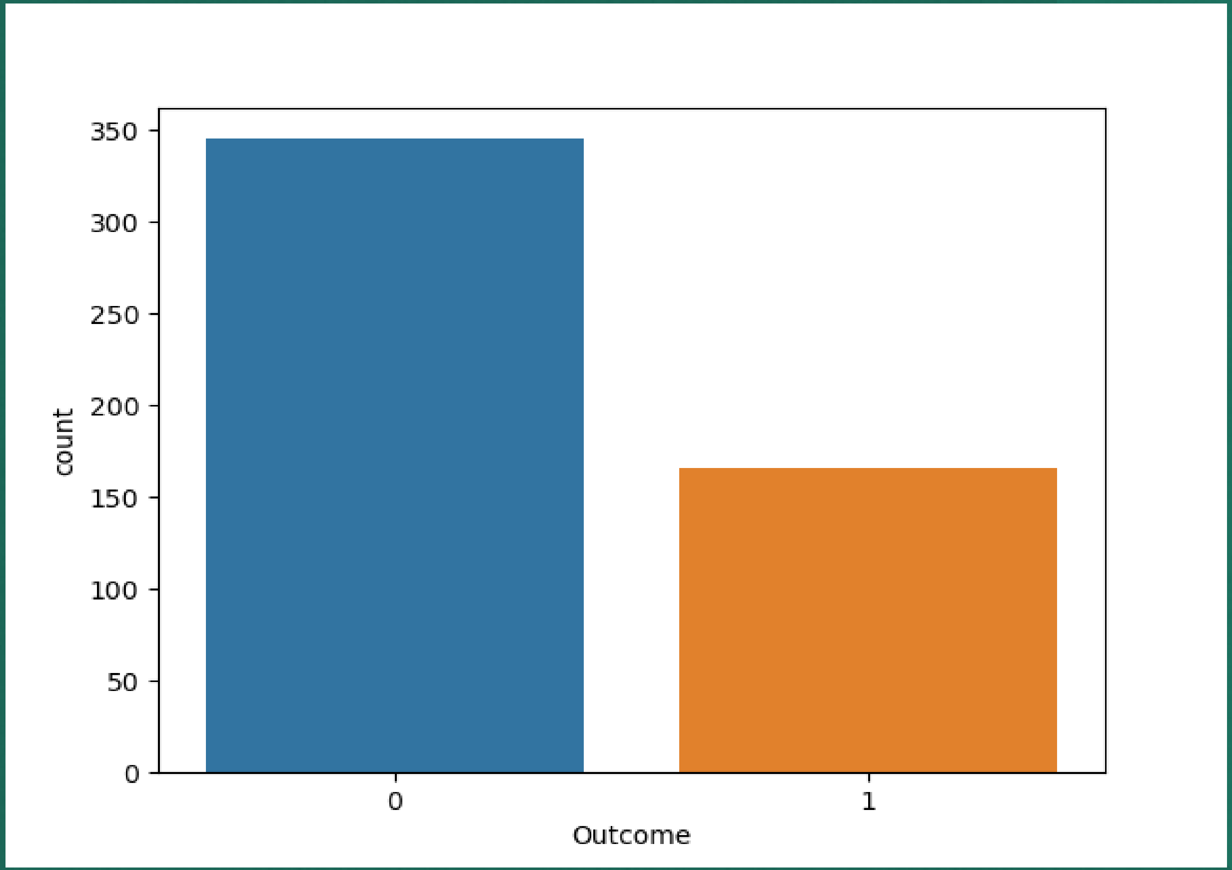


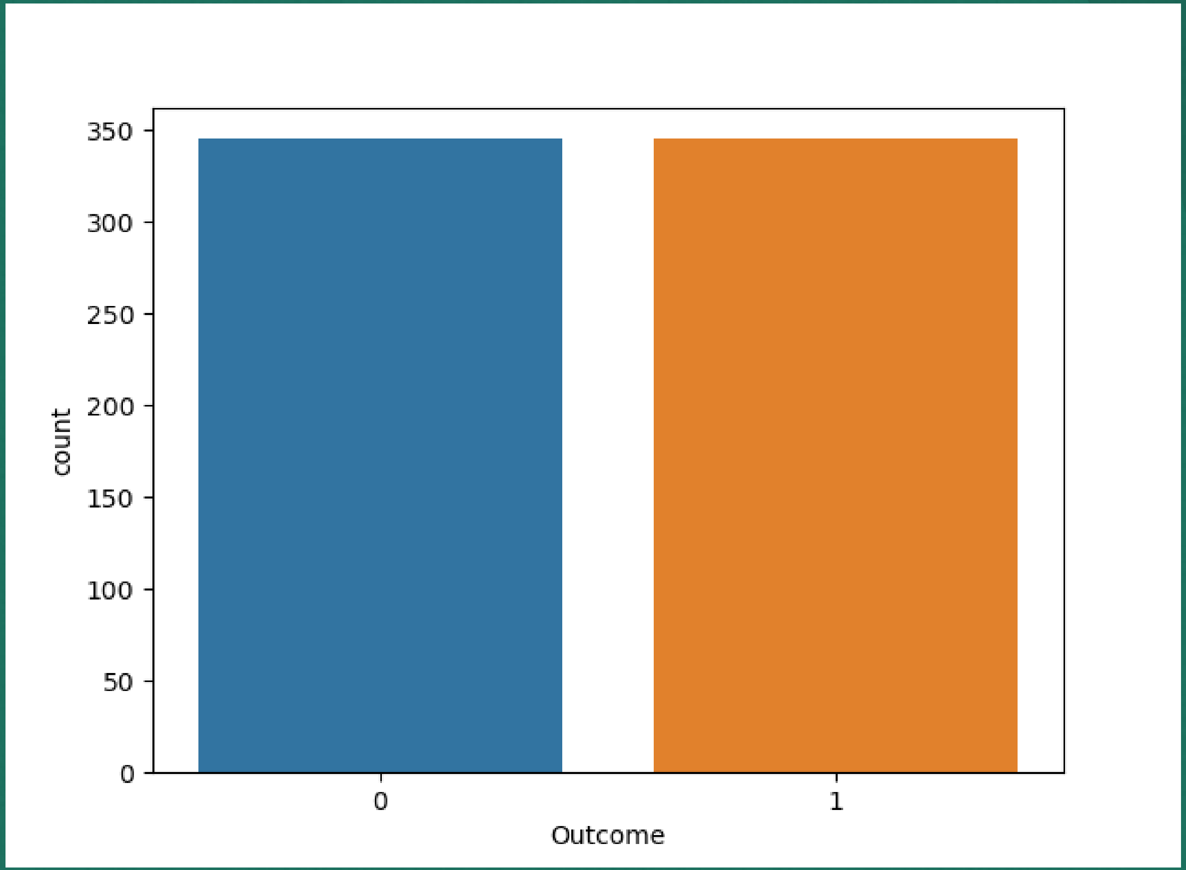
Machine Learning – Data Preparation
Removing Outliers





Machine Learning – Data Preparation
Treating Data Imbalancy in Target Variable using SMOTE





Machine Learning Model

- Multiple Machine Learning Models have been taken into consideration.
  - **Logistic Regression**
  - **Support Vector Classifier (SVC)**
  - **DecisionTreeClassifier()**
  - **BaggingClassifier()**
  - **RandomForestClassifier()**
  - **GradientBoostClassifier()**

- Models performance has been analyzed on the basis of following :
  - **Accuracy_Score**
  - **Precision Score**
  - **Recall Score**
  - **F1 – Score**
  - **Confusion Matrix**

Machine Learning Model : Logistic Regression

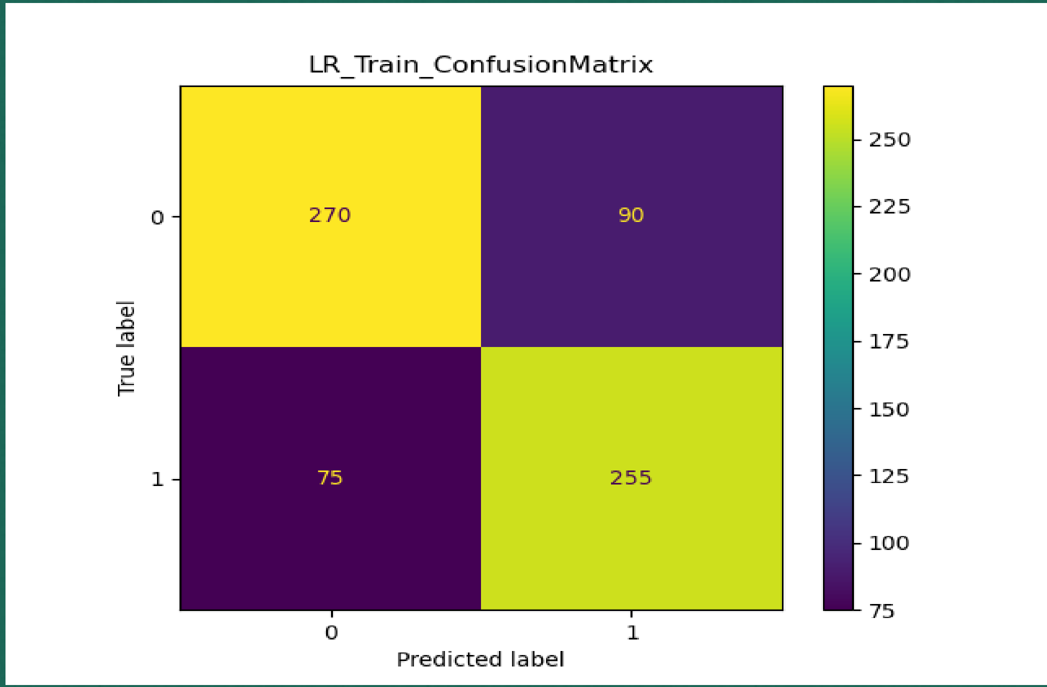
Model Training

```
In [70]:  1 print(classification_report(y_pred,y_train))

              precision    recall  f1-score   support

           0       0.78      0.75      0.77       360
           1       0.74      0.77      0.76       330

    accuracy                           0.76       690
   macro avg       0.76      0.76      0.76       690
weighted avg       0.76      0.76      0.76       690
```



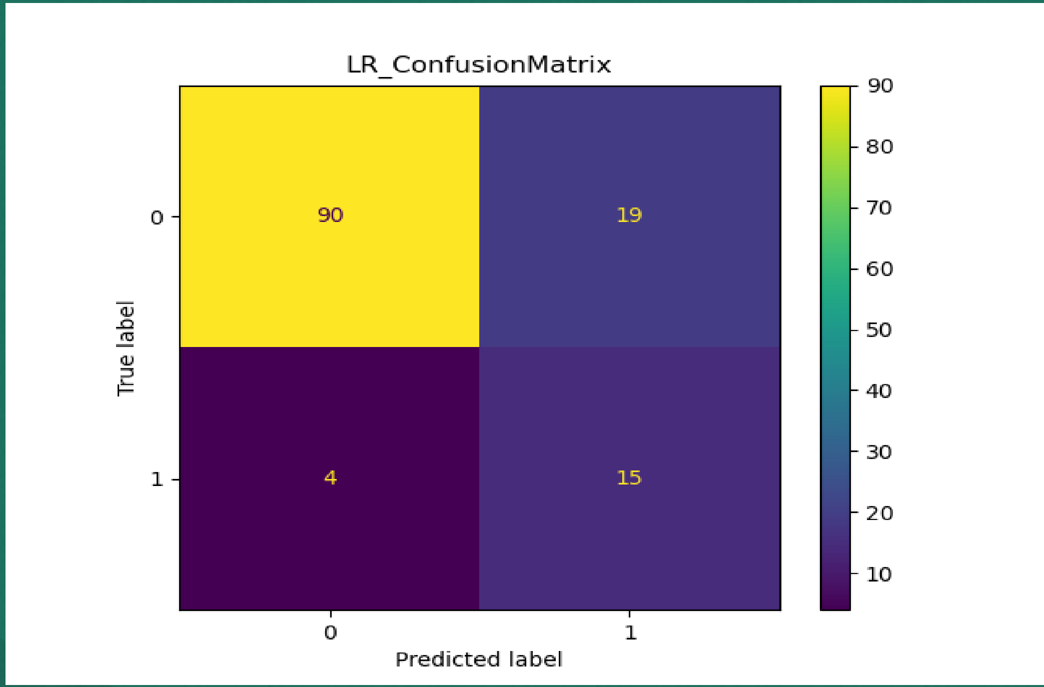
Model Validation

```
In [75]:  1 print(classification_report(y_pred,y_test))

              precision    recall  f1-score   support

           0       0.96      0.83      0.89       109
           1       0.44      0.79      0.57        19

    accuracy                           0.82       128
   macro avg       0.70      0.81      0.73       128
weighted avg       0.88      0.82      0.84       128
```



Machine Learning Model : SVC

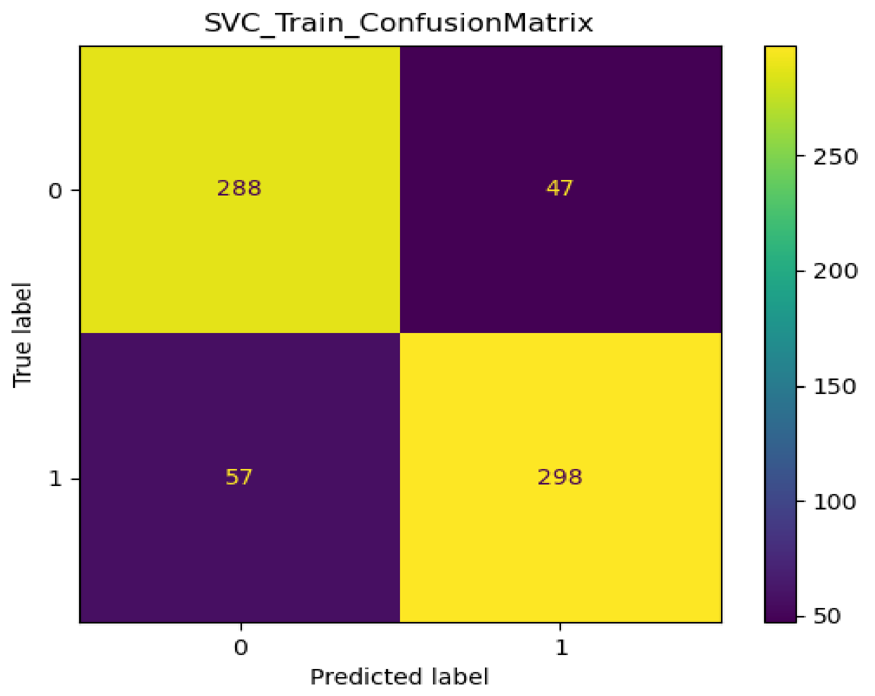
Model Training

```
In [79]:   1 print(classification_report(y_pred,y_train))

              precision    recall  f1-score   support

           0       0.83      0.86      0.85       335
           1       0.86      0.84      0.85       355

    accuracy                           0.85       690
   macro avg       0.85      0.85      0.85       690
weighted avg       0.85      0.85      0.85       690
```



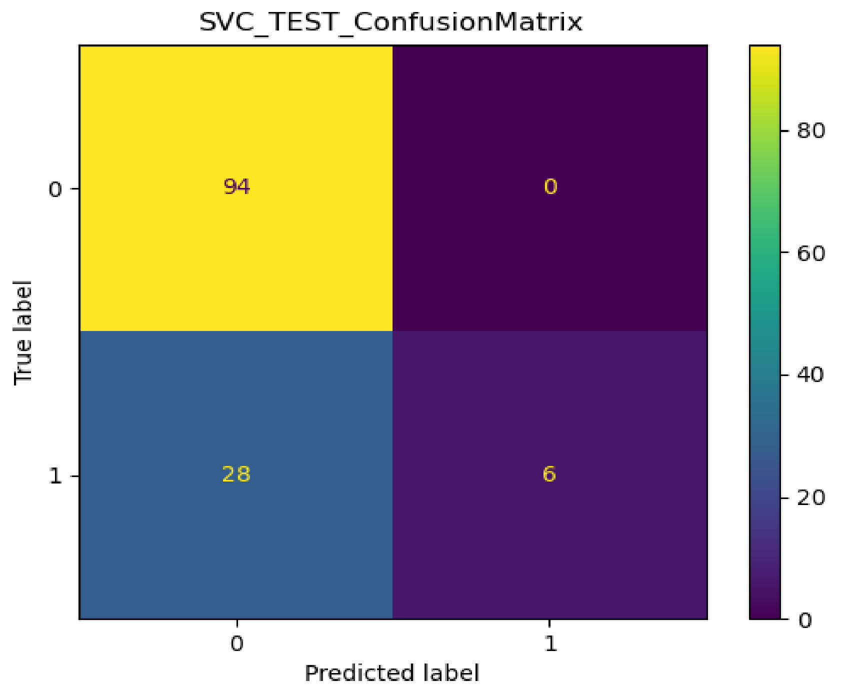
Model Validation

```
In [83]:   1 print(classification_report(y_test,y_pred))

              precision    recall  f1-score   support

           0       0.77      1.00      0.87        94
           1       1.00      0.18      0.30        34

    accuracy                           0.78       128
   macro avg       0.89      0.59      0.59       128
weighted avg       0.83      0.78      0.72       128
```



Machine Learning Model : Decision Tree

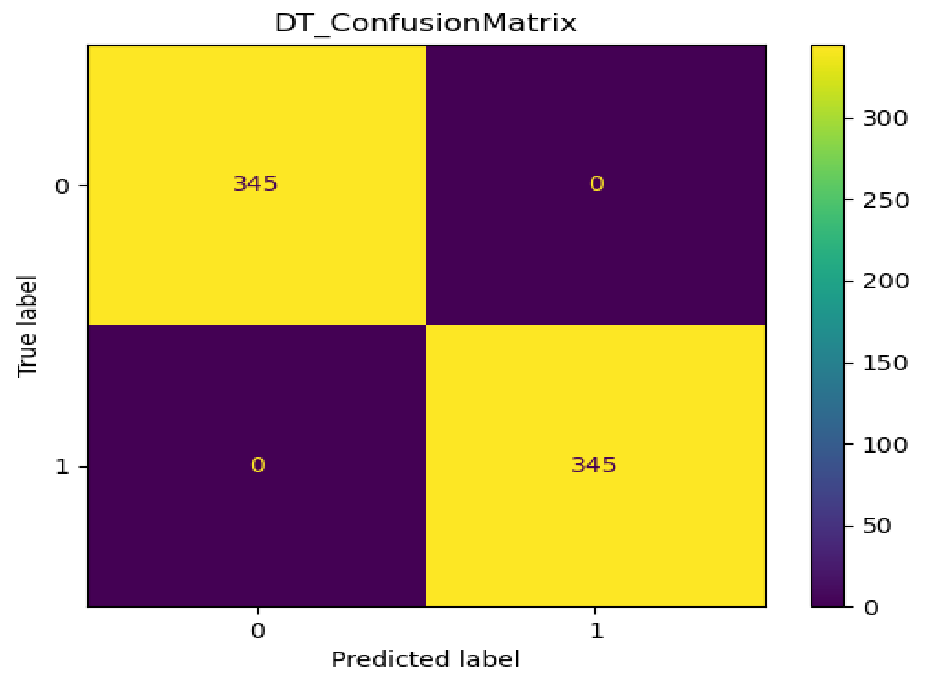
Model Training

```
In [88]:  1  print(classification_report(y_pred,y_train))

              precision    recall  f1-score   support

           0       1.00      1.00      1.00       345
           1       1.00      1.00      1.00       345

    accuracy                           1.00       690
   macro avg       1.00      1.00      1.00       690
weighted avg       1.00      1.00      1.00       690
```



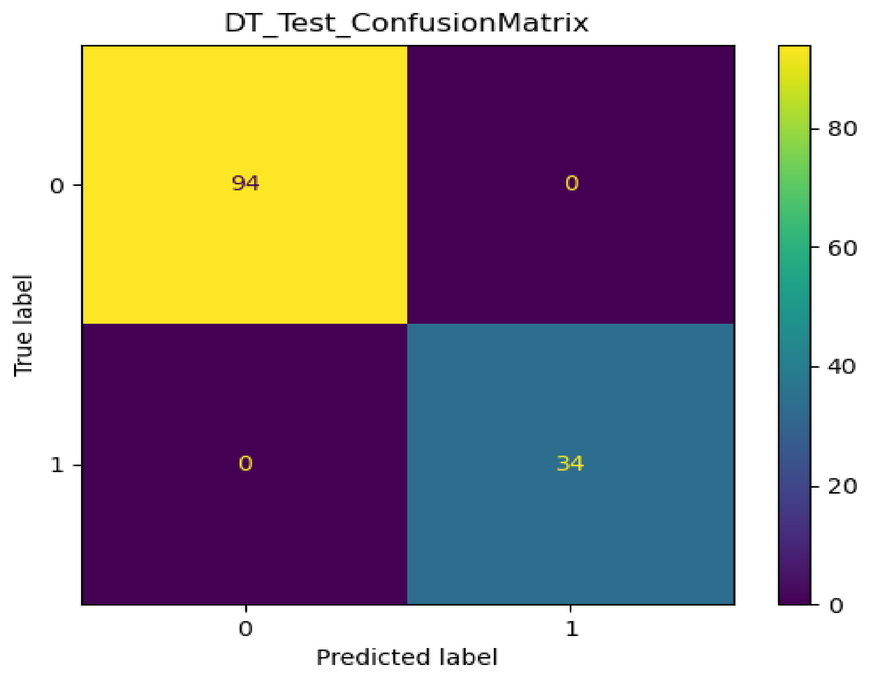
Model Validation

```
In [91]:  1  print(classification_report(y_test,y_pred))

              precision    recall  f1-score   support

           0       1.00      1.00      1.00        94
           1       1.00      1.00      1.00        34

    accuracy                           1.00       128
   macro avg       1.00      1.00      1.00       128
weighted avg       1.00      1.00      1.00       128
```

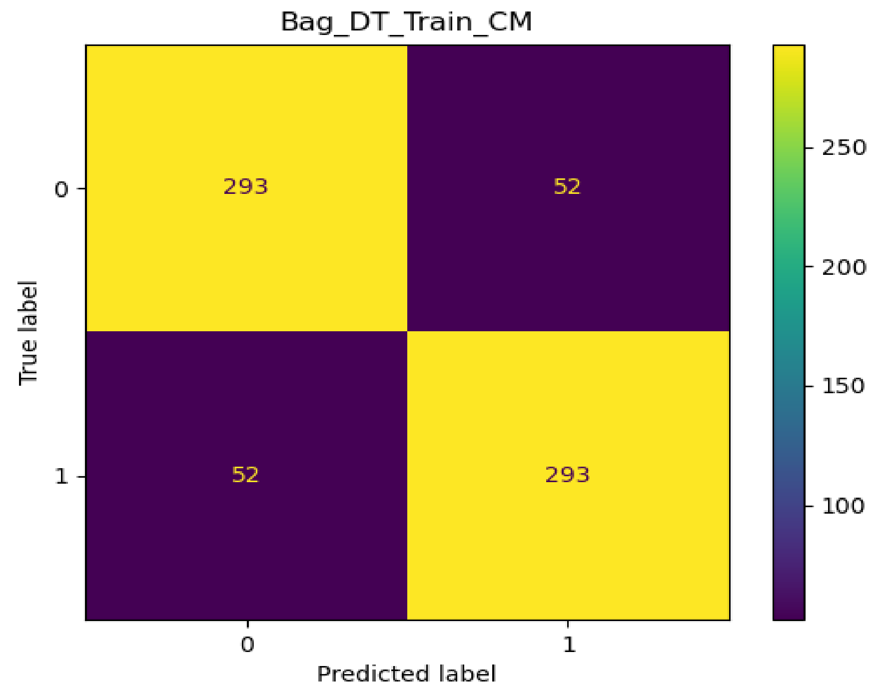


Machine Learning Model : BaggingClassifier()

Model Training

```
In [96]:  1 print(classification_report(y_train,y_pred))
```

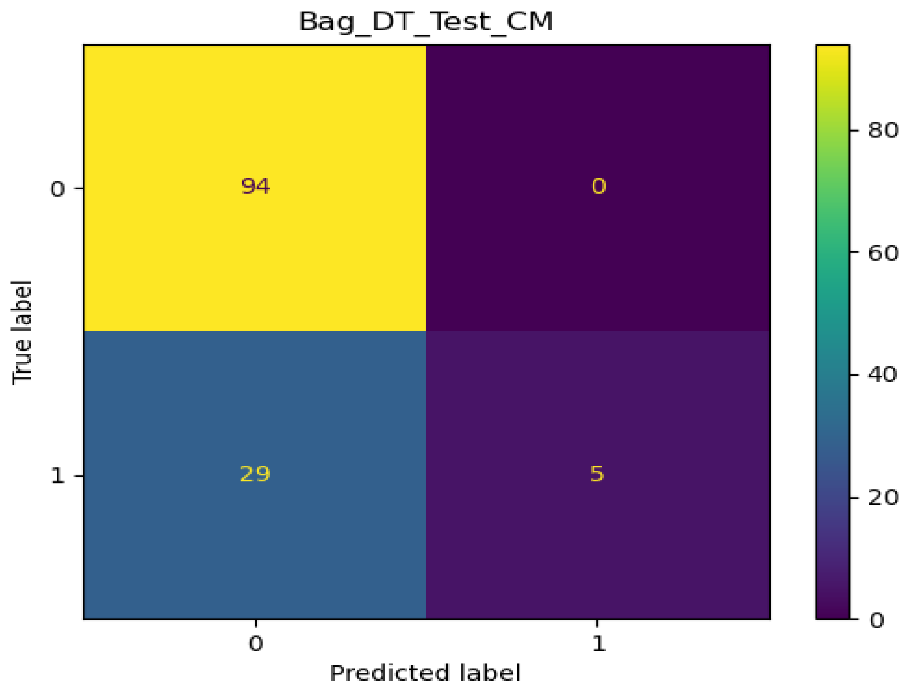
	precision	recall	f1-score	support
0	0.99	0.99	0.99	345
1	0.99	0.99	0.99	345
accuracy			0.99	690
macro avg	0.99	0.99	0.99	690
weighted avg	0.99	0.99	0.99	690



Model Validation

```
In [100]:  1 print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	0.97	1.00	0.98	94
1	1.00	0.91	0.95	34
accuracy			0.98	128
macro avg	0.98	0.96	0.97	128
weighted avg	0.98	0.98	0.98	128



Machine Learning Model : RandomForestClassifier()

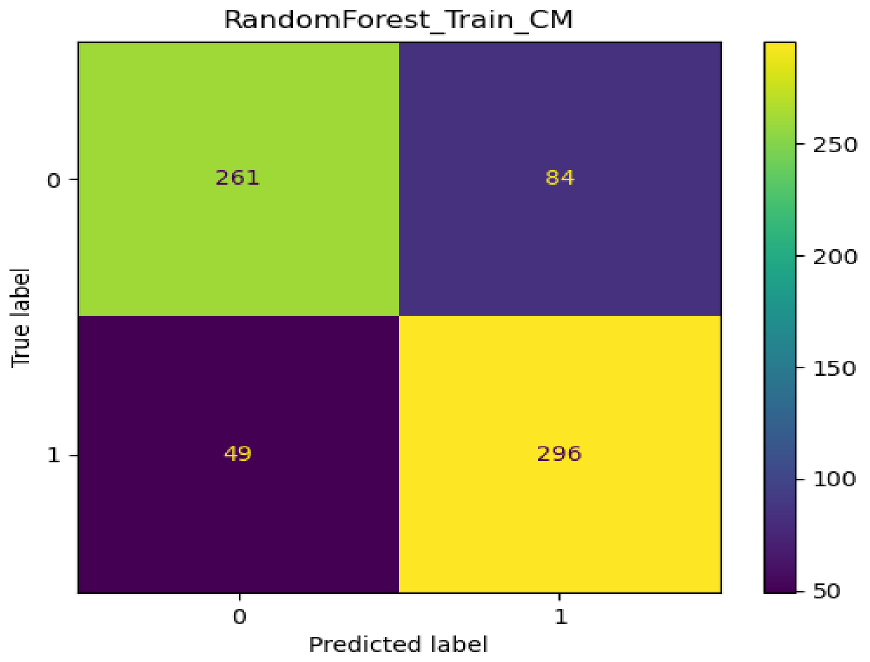
Model Training

In [114]:
```
print(classification_report(y_train,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.84      0.76      0.80       345
           1       0.78      0.86      0.82       345

    accuracy                           0.81       690
   macro avg       0.81      0.81      0.81       690
weighted avg       0.81      0.81      0.81       690
```



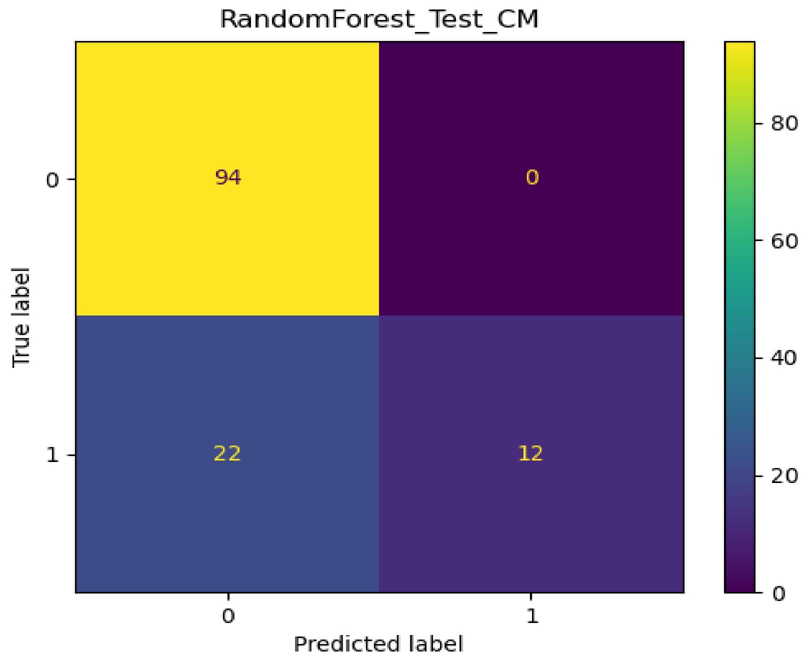
Model Validation

In [118]:
```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.81      1.00      0.90        94
           1       1.00      0.35      0.52        34

    accuracy                           0.83       128
   macro avg       0.91      0.68      0.71       128
weighted avg       0.86      0.83      0.80       128
```



Machine Learning Model : GradientBoostingClassifier()

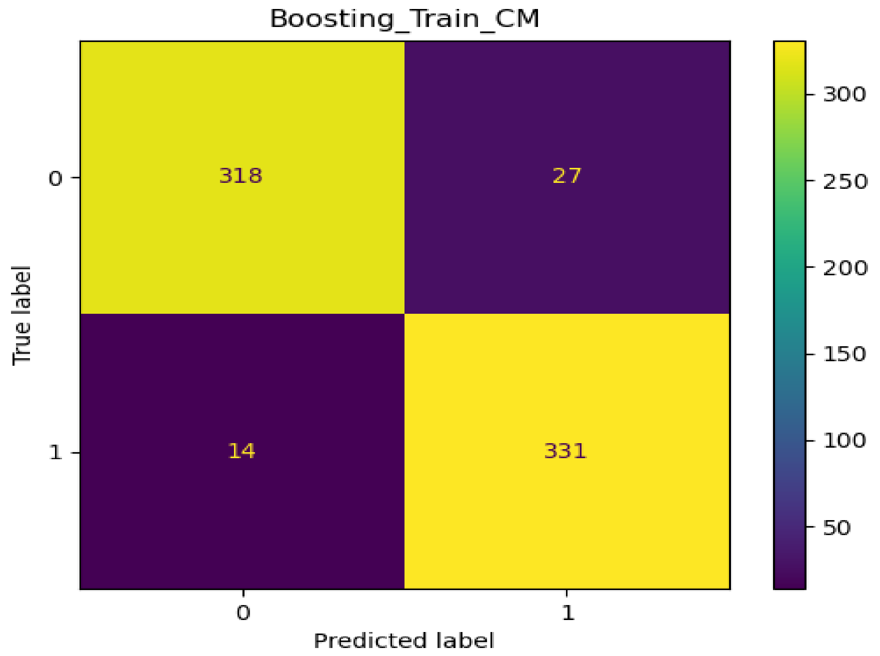
Model Training

In [123]:
```
print(classification_report(y_train,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.96      0.92      0.94       345
           1       0.92      0.96      0.94       345

    accuracy                           0.94       690
   macro avg       0.94      0.94      0.94       690
weighted avg       0.94      0.94      0.94       690
```



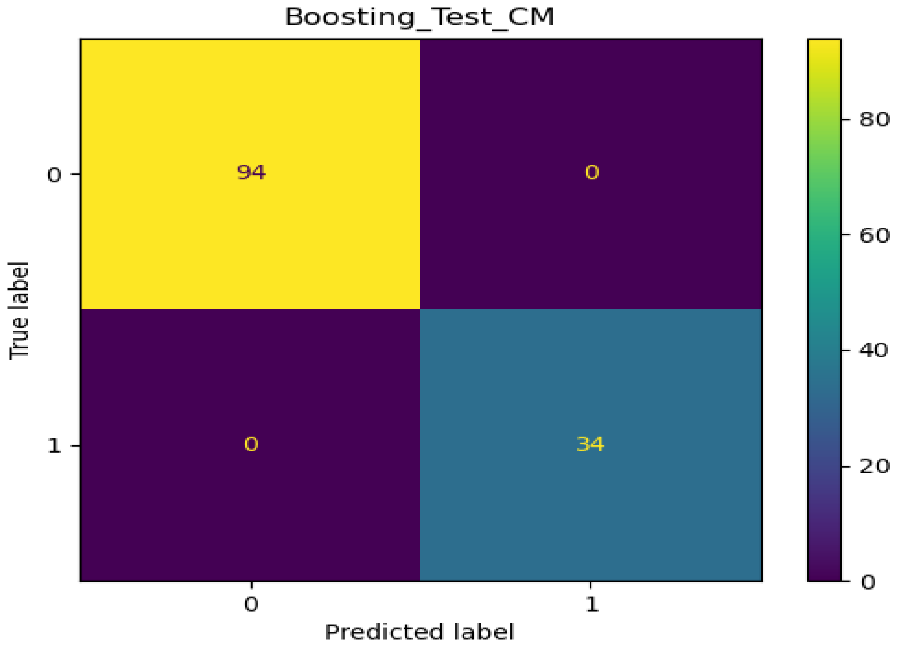
Model Validation

In [127]:
```
print(classification_report(y_test,y_pred))
```

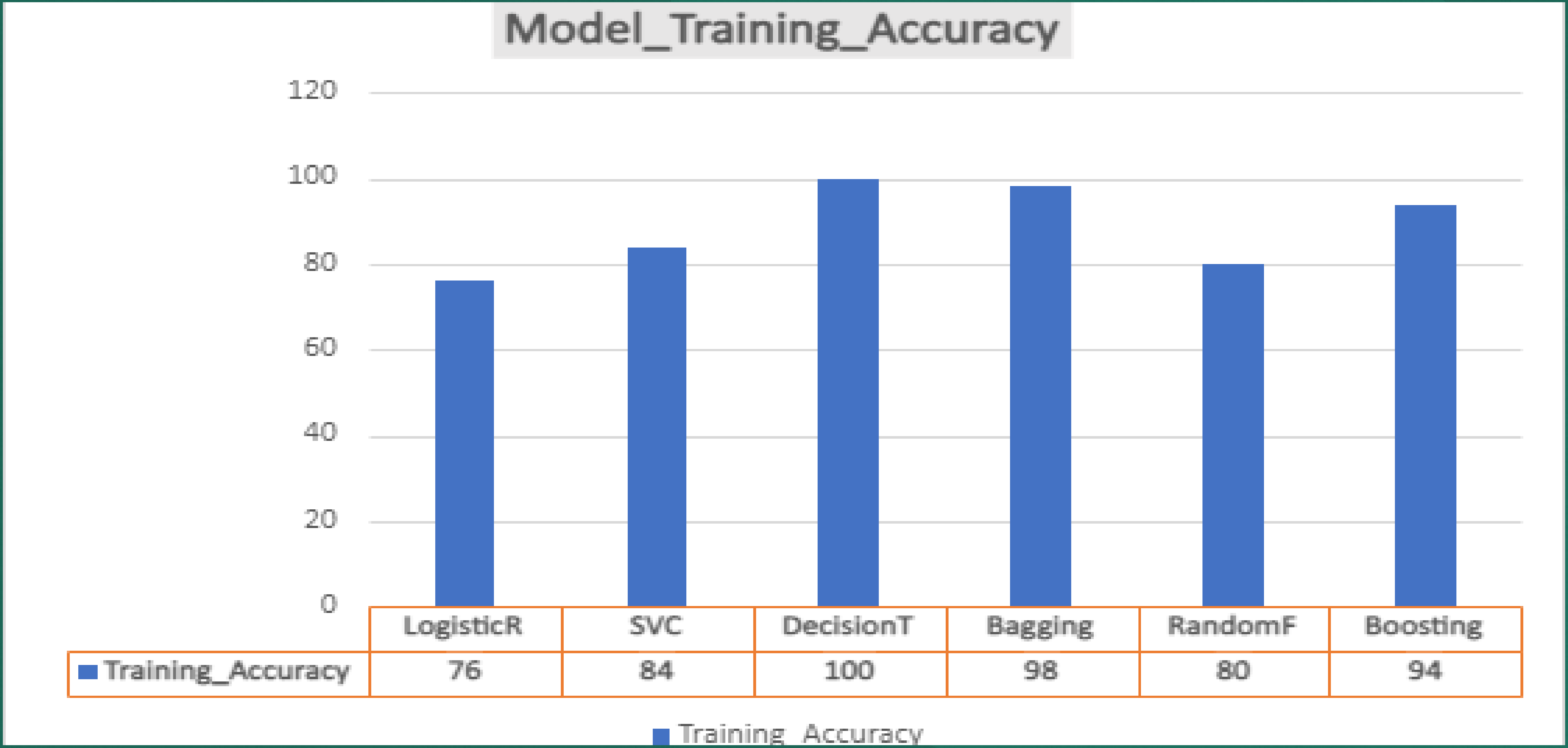
```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        94
           1       1.00      1.00      1.00        34

    accuracy                           1.00       128
   macro avg       1.00      1.00      1.00       128
weighted avg       1.00      1.00      1.00       128
```

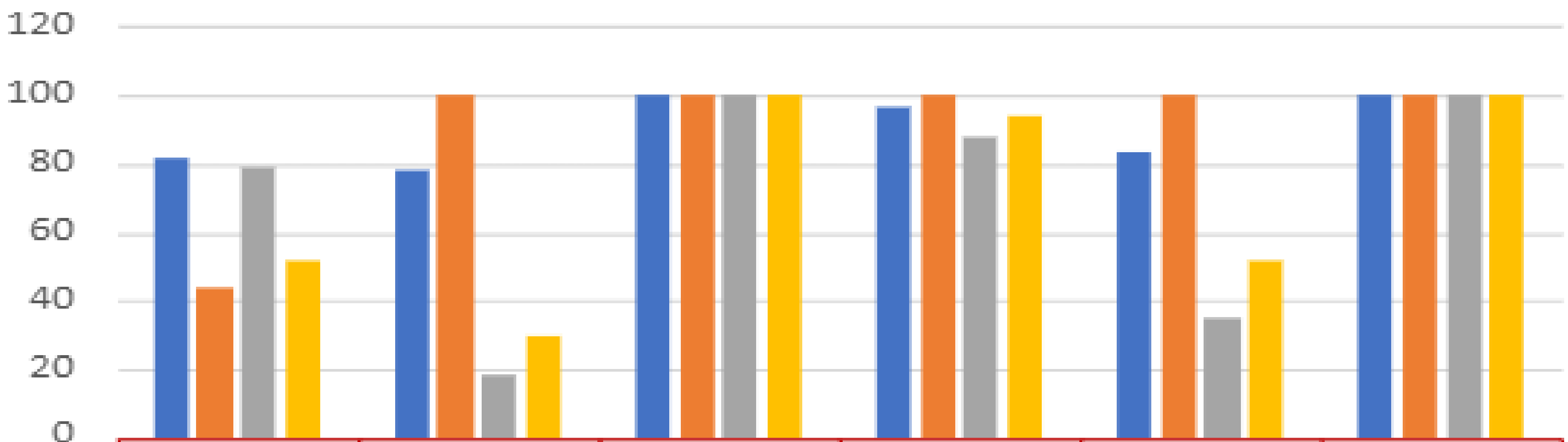


Machine Learning Model : Training Accuracy Score



Machine Learning Model : Model Validation Comparision

Model Performance Comparision



	LogisticR	SVC	DecisionT	Bagging	RandomF	Boosting
Accuracy_Score	82	78	100	97	83	100
Precision	44	100	100	100	100	100
Recall	79	18	100	88	35	100
F1-Score	52	30	100	94	52	100

Accuracy_Score Precision Recall F1-Score

CONCLUSION : -

After an in-depth analysis of various machine learning models, including Logistic Regression, SVC, Decision Tree, Bagging, Random Forest, and Gradient Boosting, I have identified two standout performers for our task.

Decision Tree Model:

Training Accuracy: 100%
Validation Accuracy: 100%

Gradient Boosting Model:

Training Accuracy: 98%
Validation Accuracy: 100%

Observations:

The Decision Tree model demonstrates exceptional accuracy on both the training and validation datasets, achieving a perfect score of 100%.
The Boosting ML Model also exhibits outstanding performance, with a training accuracy of 98% and maintaining a flawless 100% validation accuracy.

Key Considerations:

The Decision Tree model excels in simplicity and interpretability, providing a robust solution with no signs of overfitting.
Boosting, with its ensemble approach, showcases impressive generalization capabilities, making it a reliable choice for accurate predictions.

Final Decision:

Based on the comprehensive evaluation of accuracy metrics and model behavior, both the Decision Tree and Boosting ML Models have proven to be highly effective. The choice between them may depend on specific project requirements, interpretability, and the desired balance between simplicity and predictive power.