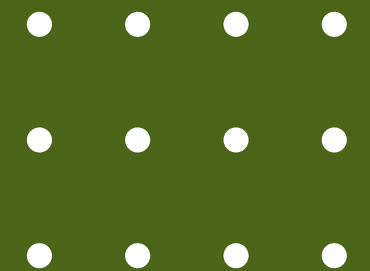
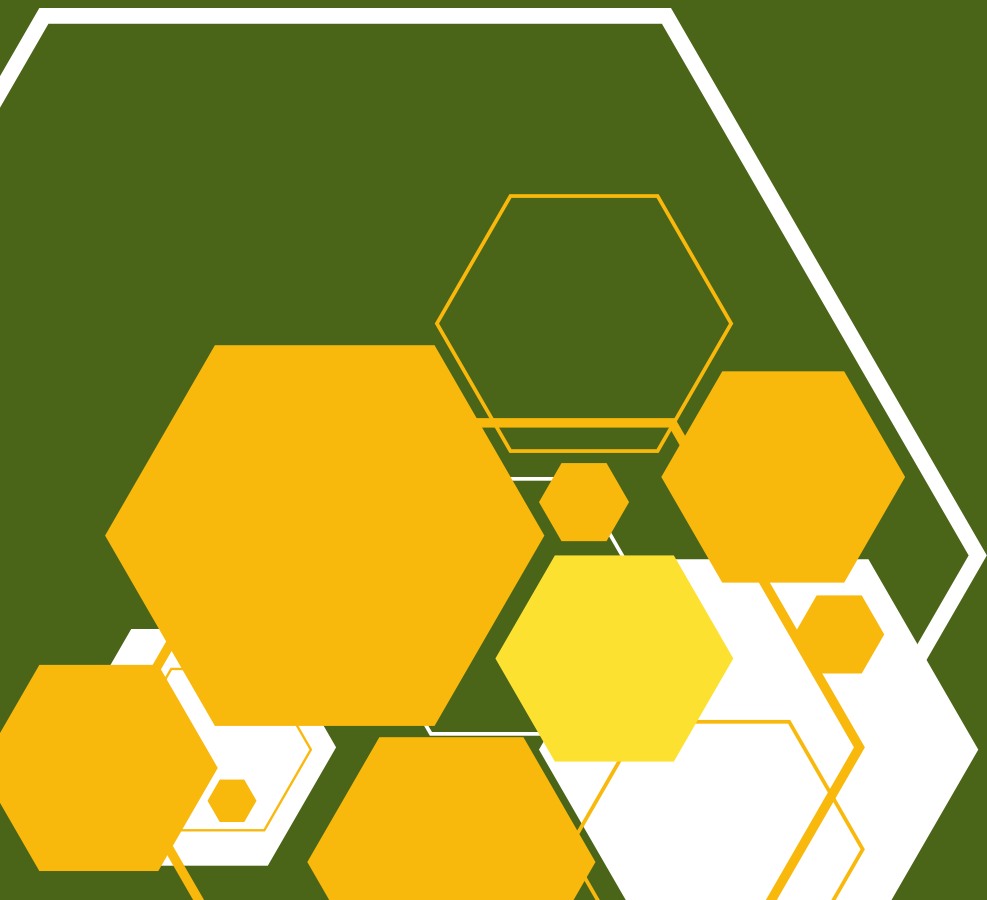


PROJECT

Predict diabetes using

Machine Learning

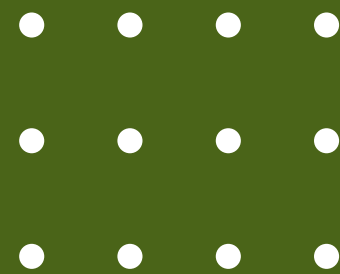
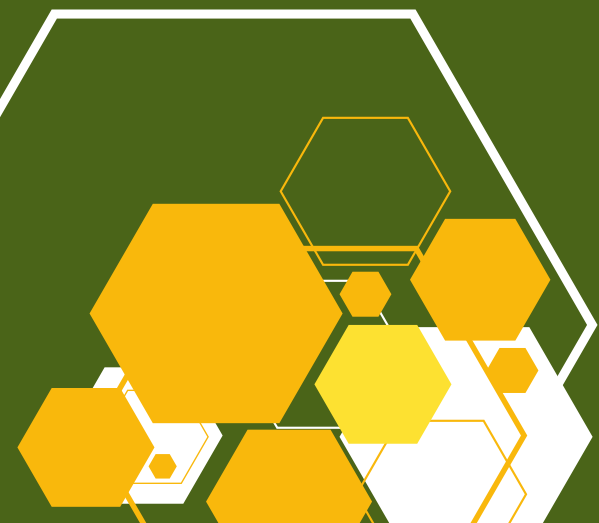
Yohannes Alexander Agusti Sinaga





Data Collection

We got the data from Kaggle.com
(<https://www.kaggle.com/datasets/mathchi/diabetes-data-set>)

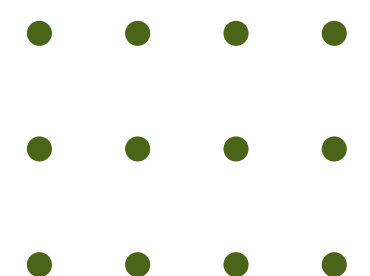


Context

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective is to predict based on diagnostic measurements whether a patient has diabetes.

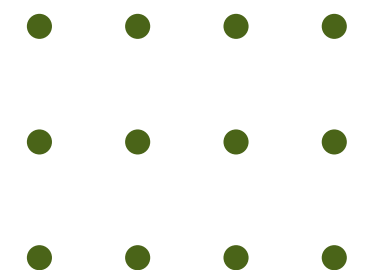
Content

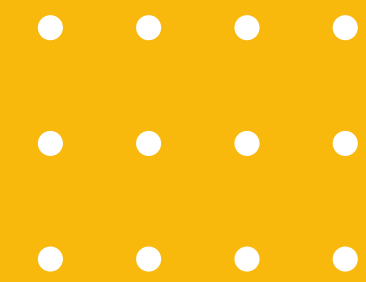
Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.



COLUMNS

- **Pregnancies:** Number of times pregnant
- **Glucose:** Plasma glucose concentration a 2 hours in an oral glucose tolerance test
- **BloodPressure:** Diastolic blood pressure (mm Hg)
- **SkinThickness:** Triceps skin fold thickness (mm)
- **Insulin:** 2-Hour serum insulin (mu U/ml)
- **BMI:** Body mass index (weight in kg/(height in m)²)
- **DiabetesPedigreeFunction:** Diabetes pedigree function
- **Age:** Age (years)
- **Outcome:** Class variable (0 or 1)





STEPS

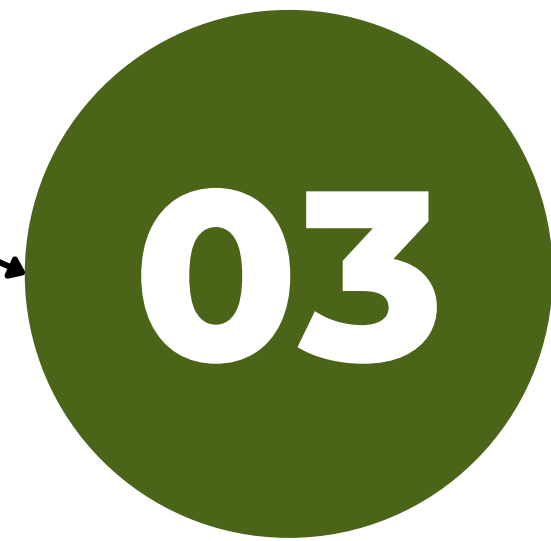
There are many steps to process the data



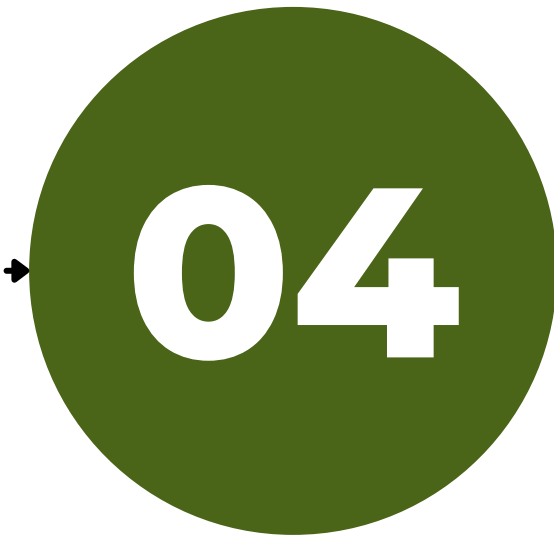
**EXPLANATORY
DATA ANALYSIS**



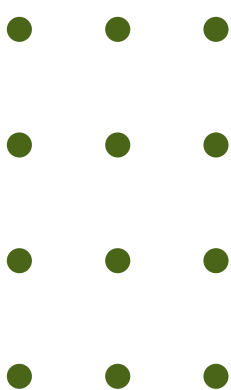
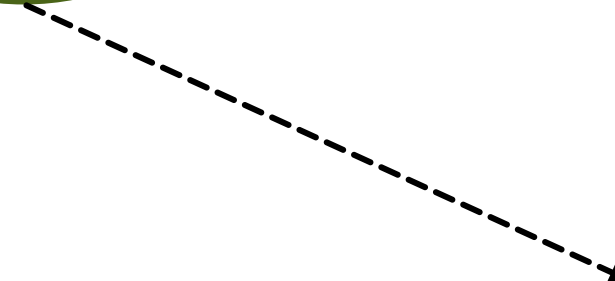
**DATA
TRANSFORMATION
/ NORMALIZATION**



MODELLING



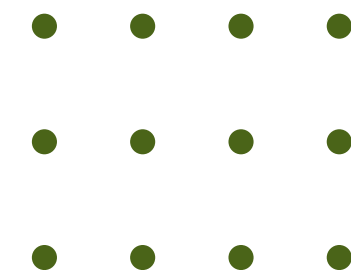
EVALUATION





EXPLANATORY DATA ANALYSIS

1. Data Understanding/Description
2. Handling Duplicate rows
3. Handling Missing Value
4. Data Visualization
5. Handling Outlier



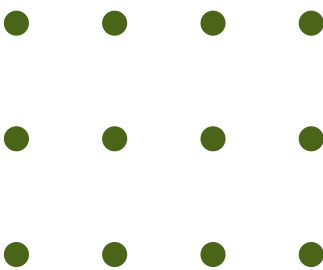


Data Understanding



The dataset contains 768 rows and 9 columns.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1



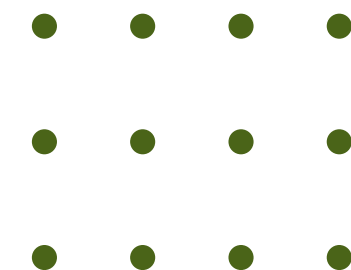


Handling Duplicate Rows

■ The dataset has no duplicate rows. Therefore, we don't conduct handling duplicate rows.





```
data.duplicated().value_counts()
```

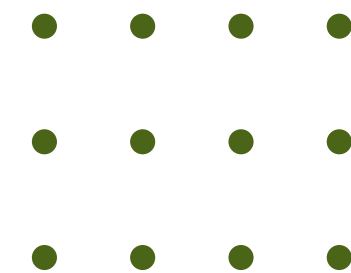
```
False      768  
dtype: int64
```





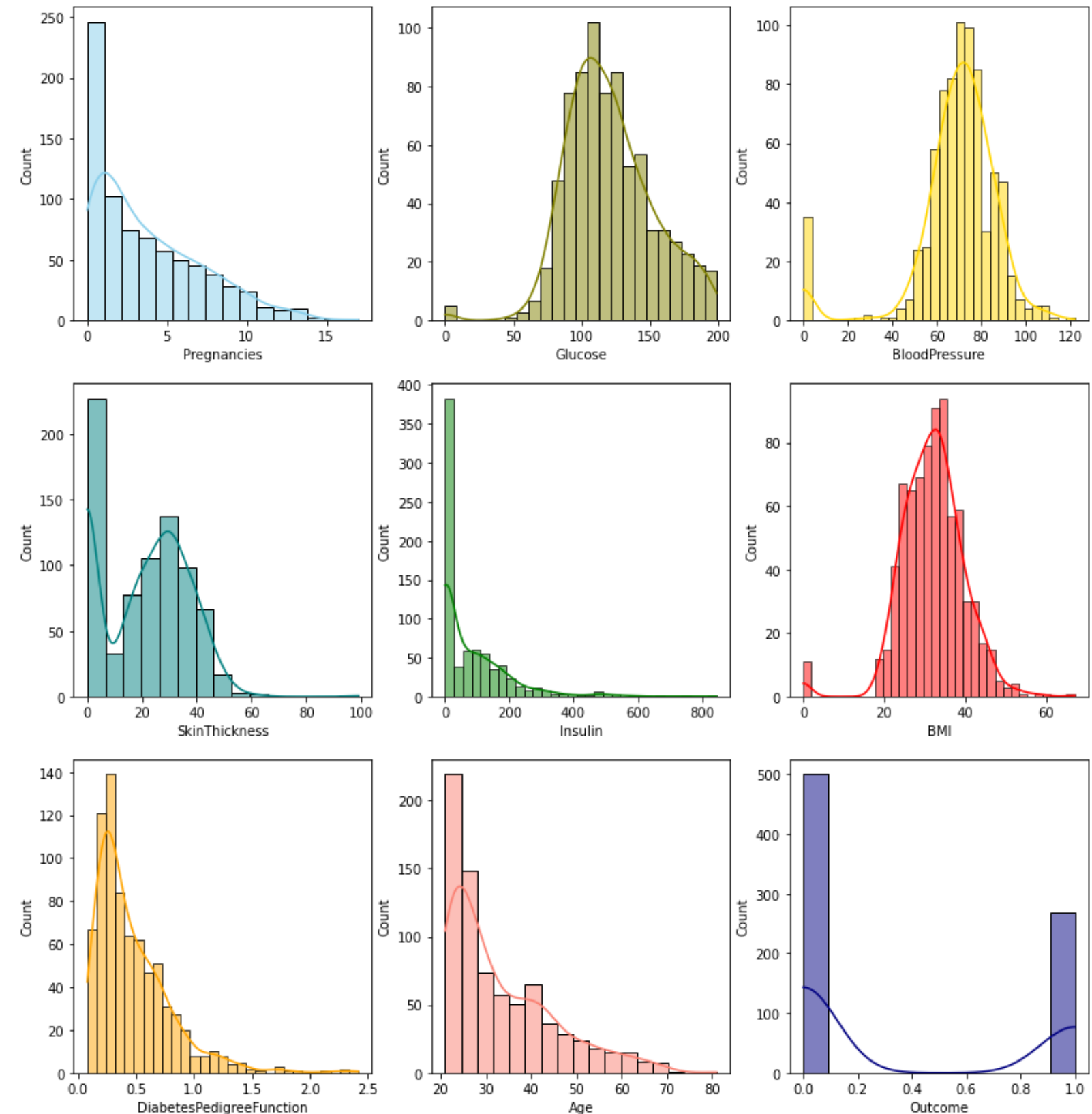
DATA VISUALIZATION

-  Histogram
-  Box Plot
-  Scatter Plot
-  Scatter Plot by Outcome

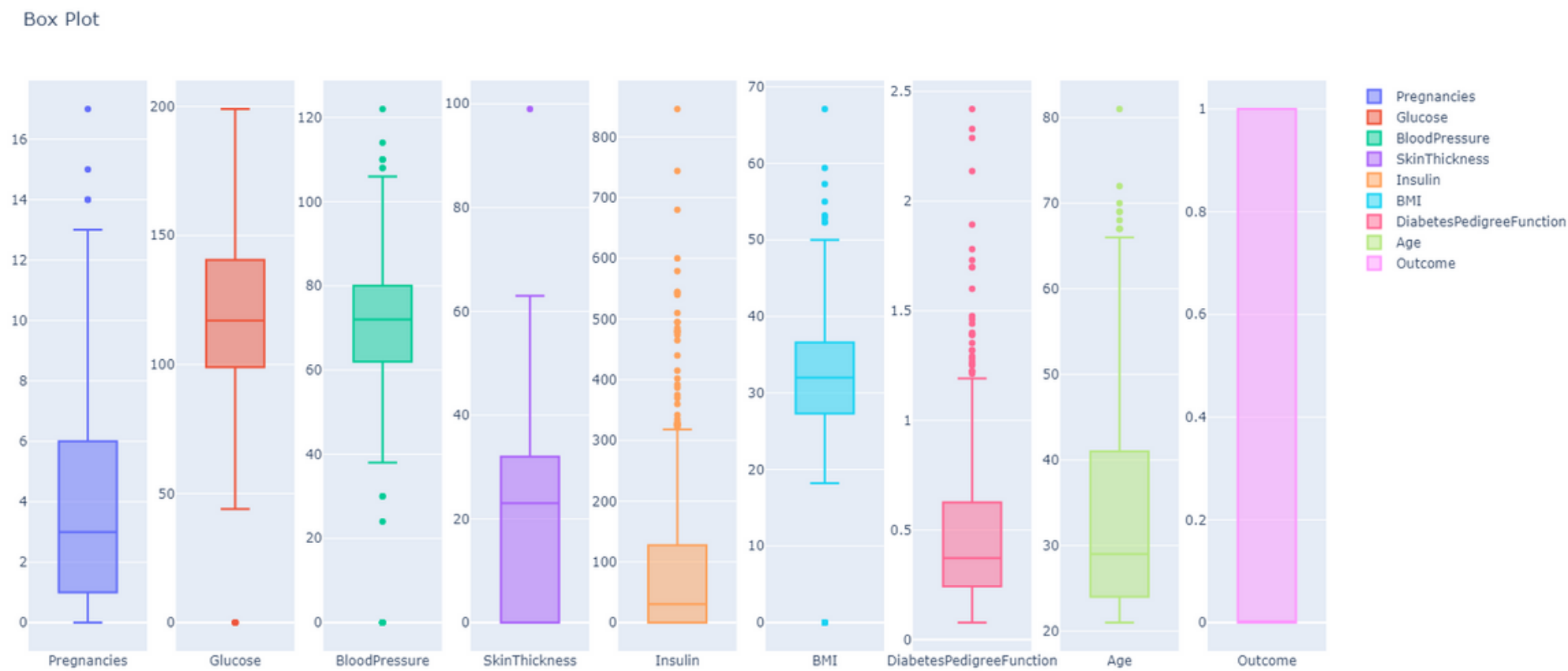


Histogram

- Histogram of Pregnancies, Insulin, DiabetesPedigreeFunction, Age are positively skewed.
- Histogram of Glucose, BloodPressure, and BMI tend to be normal distribution or symmetrical.
- Glucose, BloodPressure, BMI have outliers at 0 on the X axis.
- The peak of histogram represents that the highest values of the data.



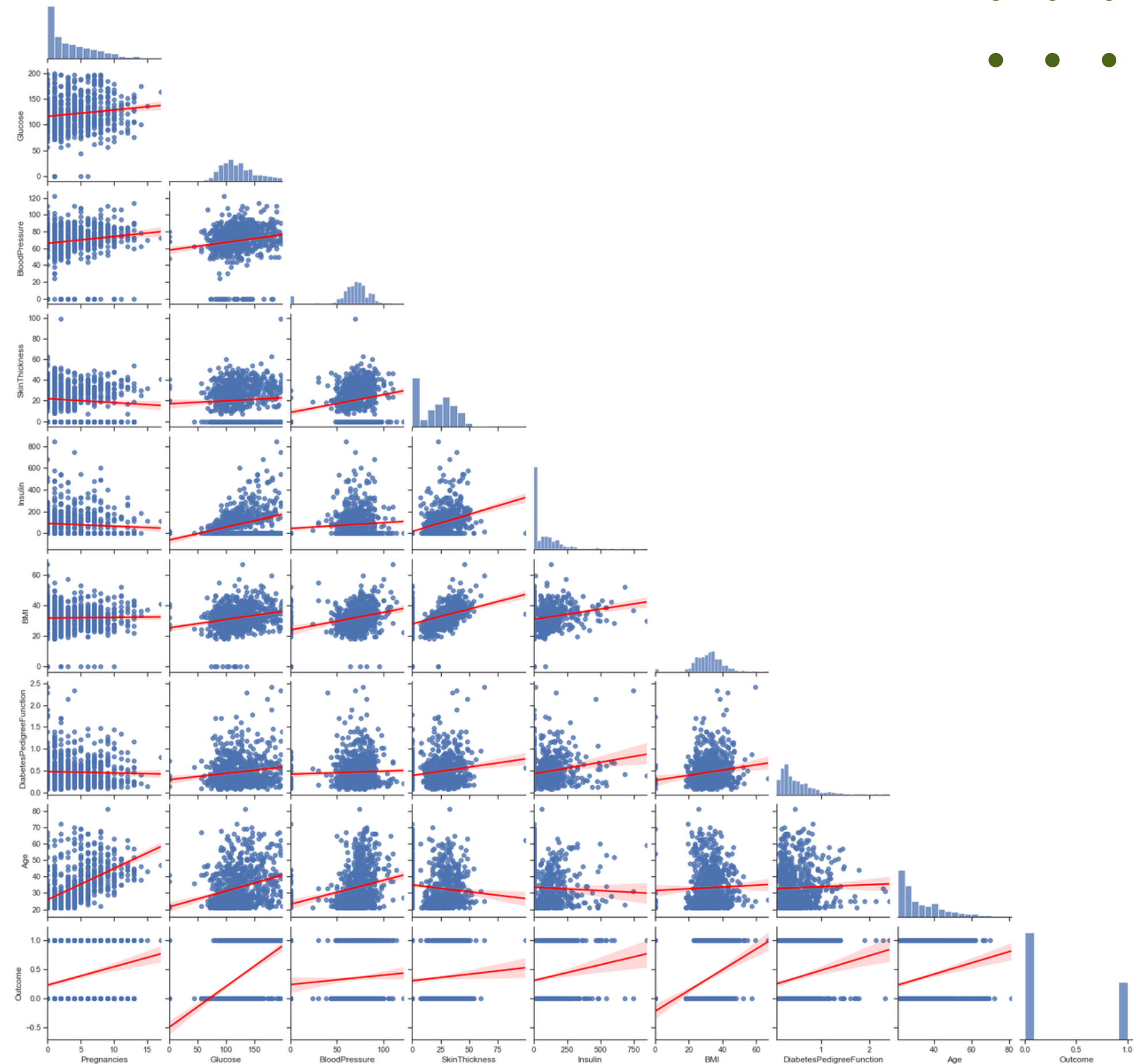
Box Plot



- To identify outliers, we emphasize in Boxplot.
- All columns have outliers marked with dots above the top fence and below the bottom fence.
- Glucose and SkinThickness have the least outliers.
- Insulin and DiabetesPedigreeFunction have the most outliers.
- Therefore, we have to conduct handling outliers.

Scatter Plot

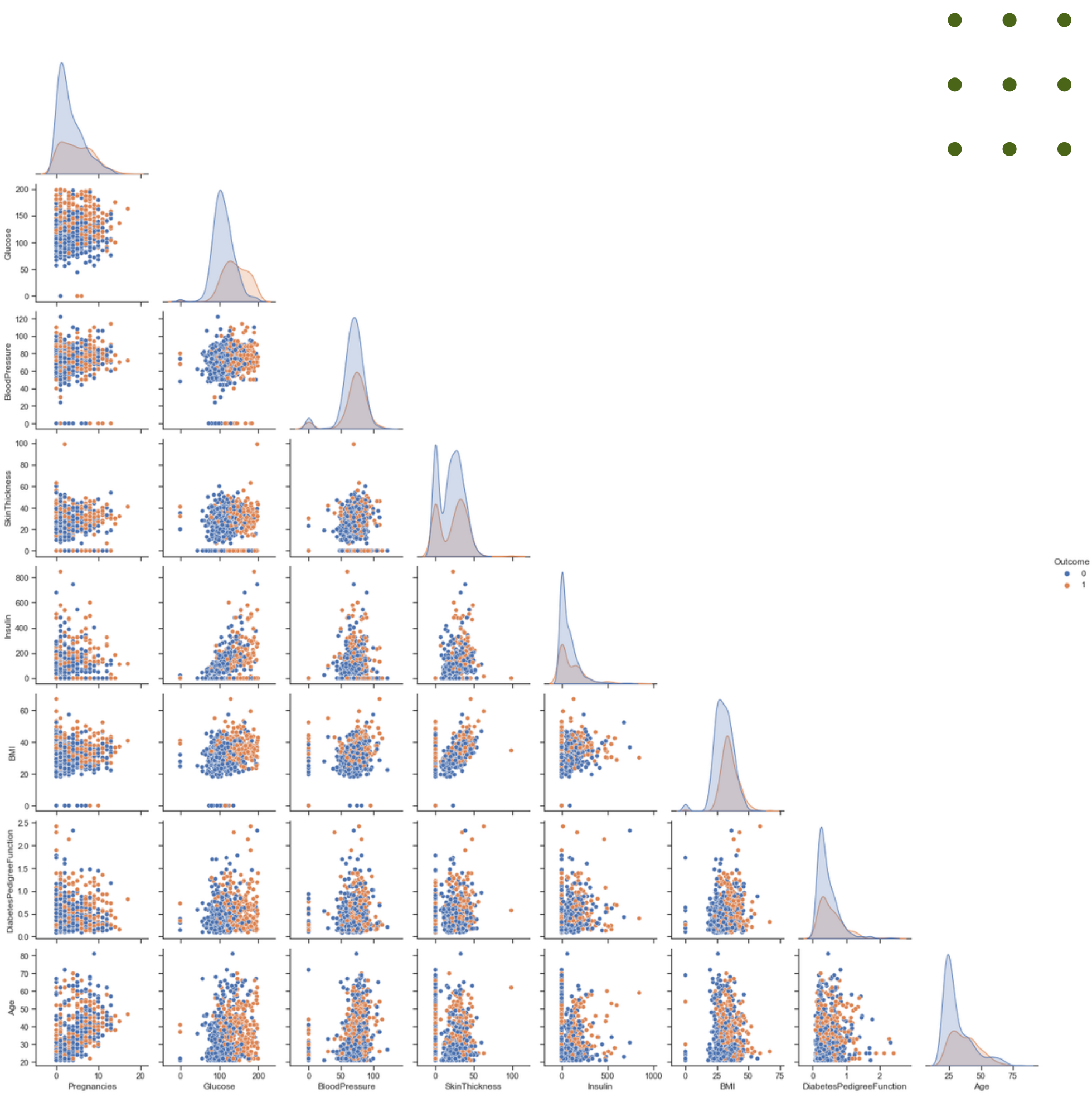
- Most of the scatter plots represent having a positive correlation.
- Pregnancies - SkinThickness, Pregnancies - Insulin, Pregnancies - DiabetesPedigreeFunction, Age - SkinThickness, Age - Insulin are negative correlation.



Scatter Plot

To be more clearly, we separate the scatter plot based on Outcome.

Scatter plot separation based on Outcome can be seen clearly with the association of color groups



HANDLING OUTLIER

After detecting outliers, handling outliers have to be implement to make cleaner data.

Handling outliers use Z-score method.
The dataset contains 129 rows outlier.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
4	0	137	40	35	168	43.1	2.288	33
7	10	115	0	0	0	35.3	0.134	29
8	2	197	70	45	543	30.5	0.158	53
9	8	125	96	0	0	0.0	0.232	54
12	10	139	80	0	0	27.1	1.441	57
...
706	10	115	0	0	0	0.0	0.261	30
707	2	127	46	21	335	34.4	0.176	22
710	3	158	64	13	387	31.2	0.295	24
715	7	187	50	33	392	33.9	0.826	34
753	0	181	88	44	510	43.3	0.222	26

129 rows × 8 columns

The datasets are cleaned from outliers, so the datasets have 639 rows.

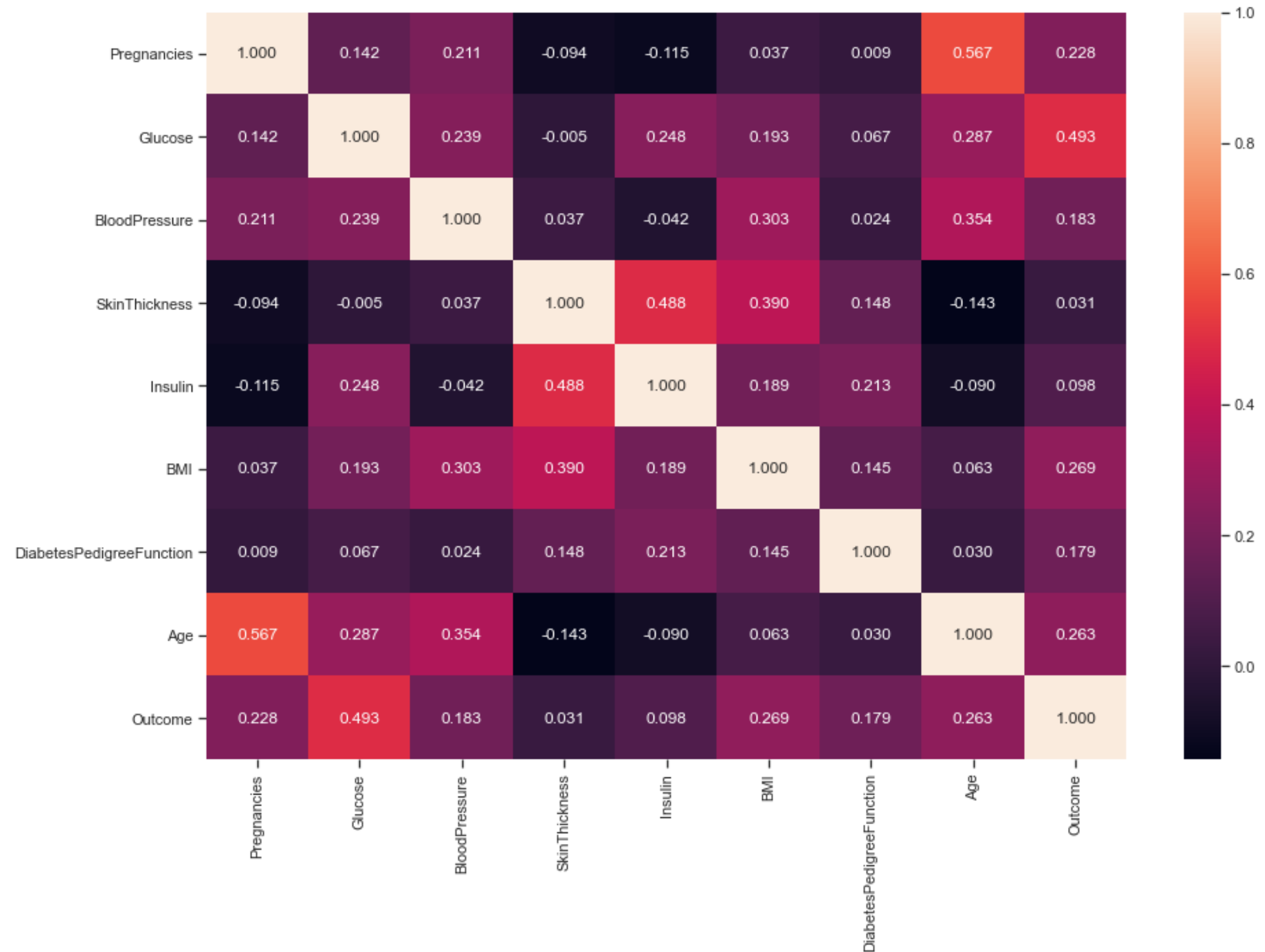
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
5	5	116	74	0	0	25.6	0.201	30	0
...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

639 rows × 9 columns

HEATMAP

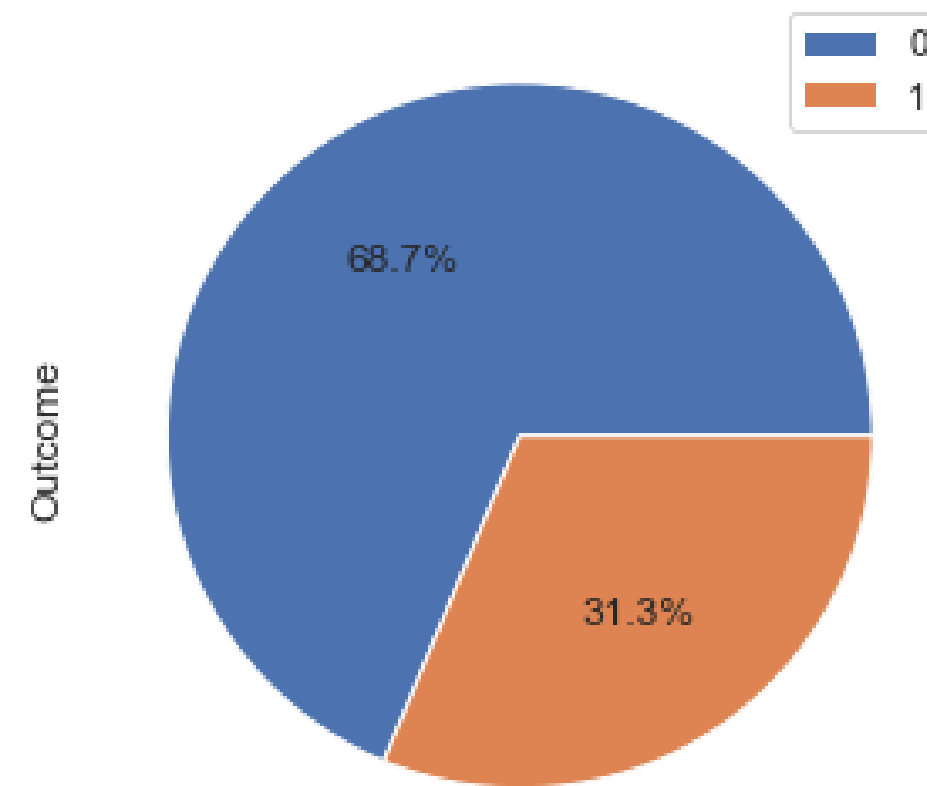
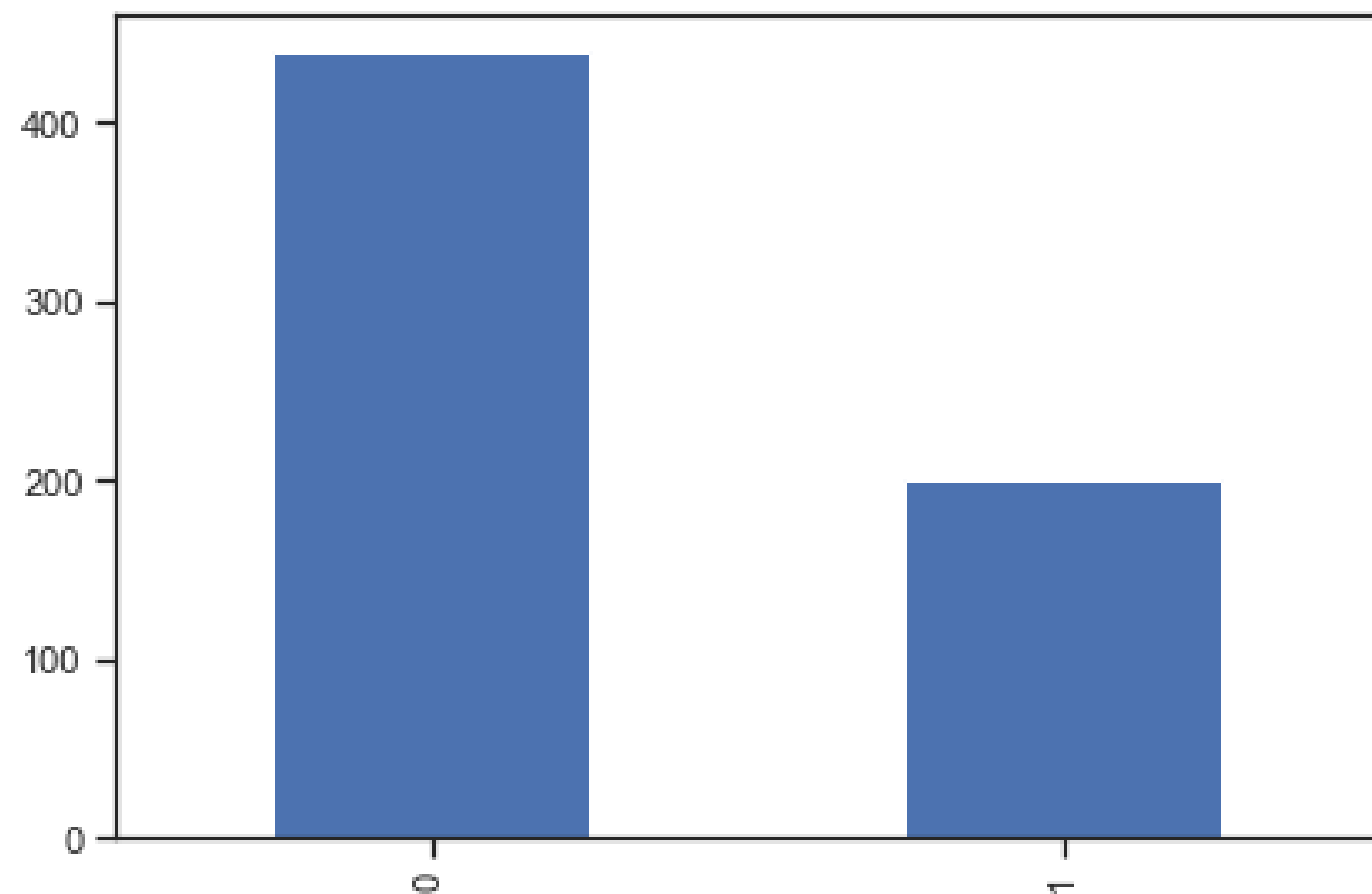
The heatmap explains the correlation between columns.

- Age and Pregnancies have high correlation.
- Glucose and Outcome have high correlation



HANDLING OUTLIER

```
0    439
1    200
Name: Outcome, dtype: int64
```



After conducting data preprocessing,

Outcome consists of 0 and 1. The total value of 0 is 439 or 68.7%

The total value of 1 is 200 or 31.3%

DATA TRANSFORMATION/ NORMALIZATION

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	0.461538	0.675325	0.500000	0.583333	0.000000	0.484277	0.493261	0.644444	1.0
1	0.076923	0.266234	0.411765	0.483333	0.000000	0.264151	0.245283	0.222222	0.0
2	0.615385	0.902597	0.382353	0.000000	0.000000	0.160377	0.533693	0.244444	1.0
3	0.076923	0.292208	0.411765	0.383333	0.295597	0.311321	0.079964	0.000000	0.0
4	0.384615	0.467532	0.529412	0.000000	0.000000	0.232704	0.110512	0.200000	0.0
...
634	0.769231	0.370130	0.558824	0.800000	0.566038	0.462264	0.083558	0.933333	0.0
635	0.153846	0.506494	0.470588	0.450000	0.000000	0.584906	0.235400	0.133333	0.0
636	0.384615	0.500000	0.500000	0.383333	0.352201	0.251572	0.150045	0.200000	0.0
637	0.076923	0.532468	0.323529	0.000000	0.000000	0.374214	0.243486	0.577778	1.0
638	0.076923	0.318182	0.470588	0.516667	0.000000	0.383648	0.212938	0.044444	0.0

639 rows × 9 columns

The method of data normalization use MinMaxScaler.

The scale of data between 0 and 1.



MODELLING

Decision Tree

```
model_dt = DecisionTreeClassifier()  
model_dt.fit(X_train,y_train)
```

Logistic Regression

```
model_lr = LogisticRegression(solver='lbfgs')  
model_lr.fit(X_train,y_train)
```

K-Nearest Neighbors

```
model_knn = KNeighborsClassifier(n_neighbors=2)  
model_knn.fit(X_train, y_train)
```

Random Forest

```
model_rf = RandomForestClassifier(n_estimators=100)  
model_rf.fit(X_train,y_train)
```

Support Vector Machine

```
model_nb = GaussianNB()  
model_nb.fit(X_train,y_train)
```

Naïve Bayes

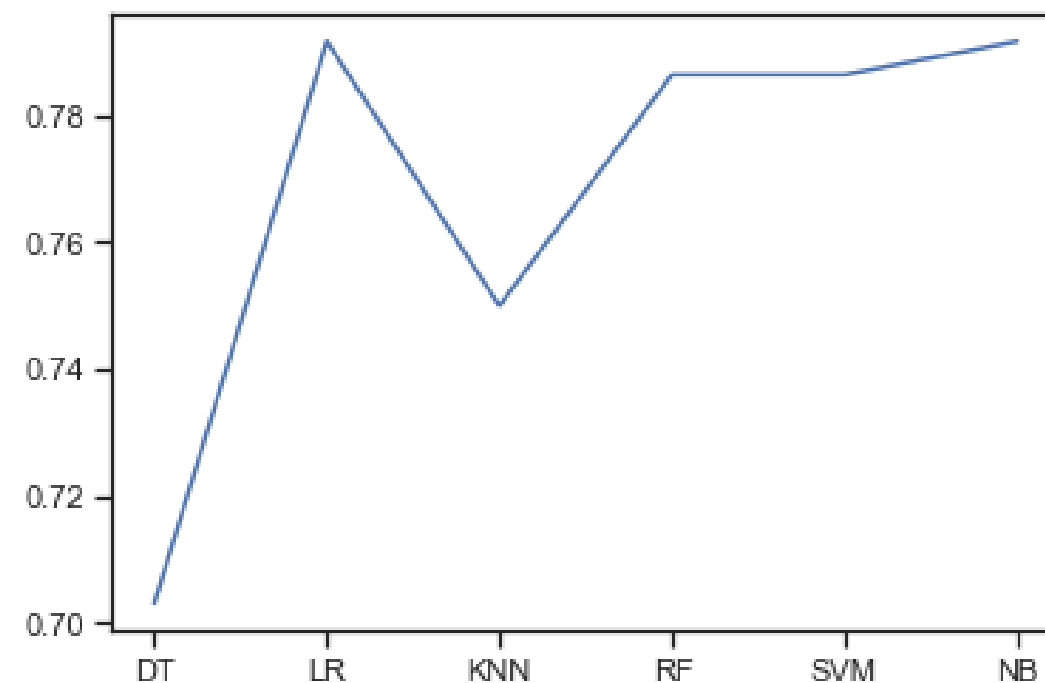
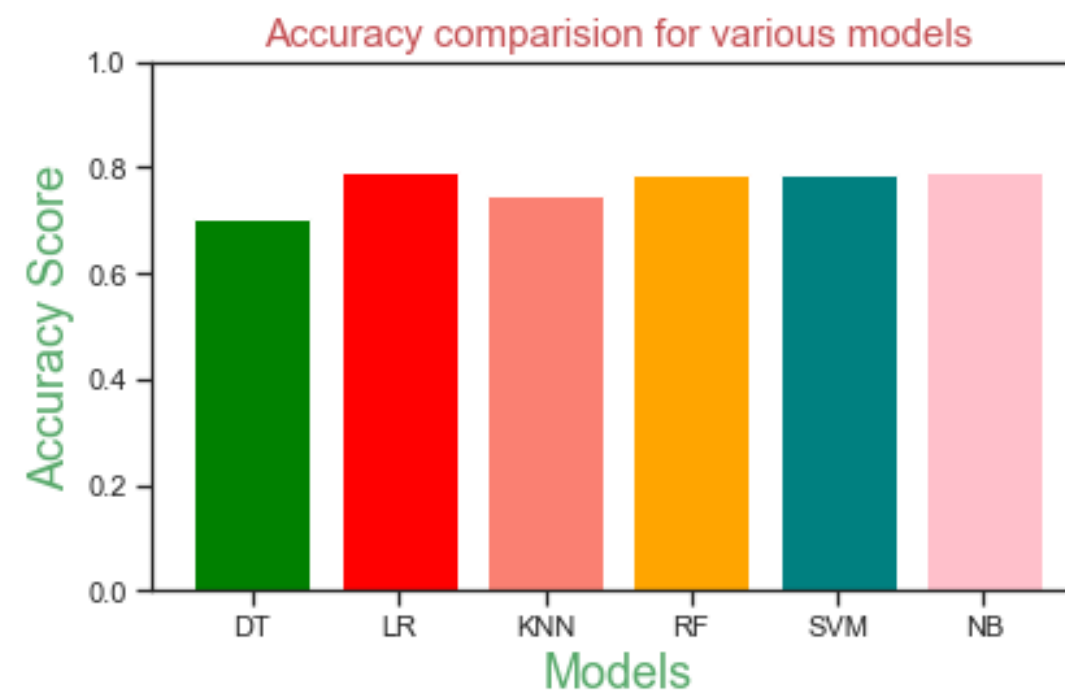
```
model_svc = SVC(gamma='scale')  
model_svc.fit(X_train,y_train)
```

EVALUATION

Accuracy

Accuracy Score

DT	0.703125
LR	0.791667
KNN	0.750000
RF	0.786458
SVM	0.786458
NB	0.791667



After training the models, we have to do testing the model using testing data.

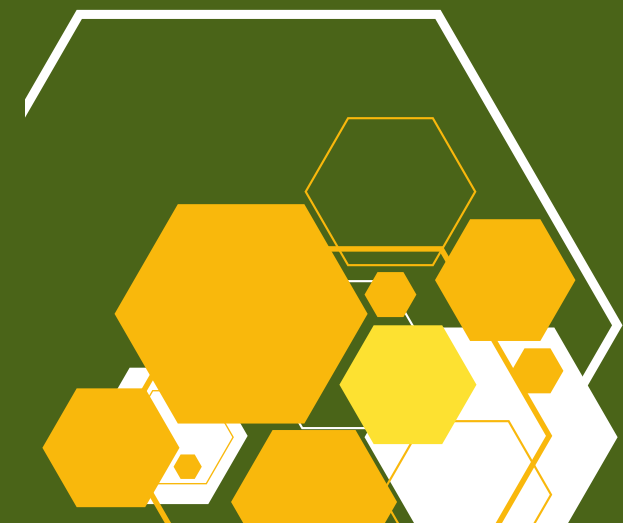
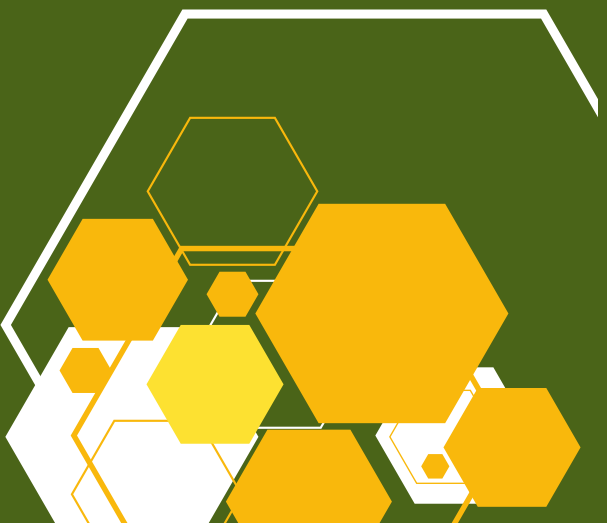
We can get that the best accuracy equals high value.

Naïve Bayes and Logistic Regression are the highest accuracy.



EVALUATION

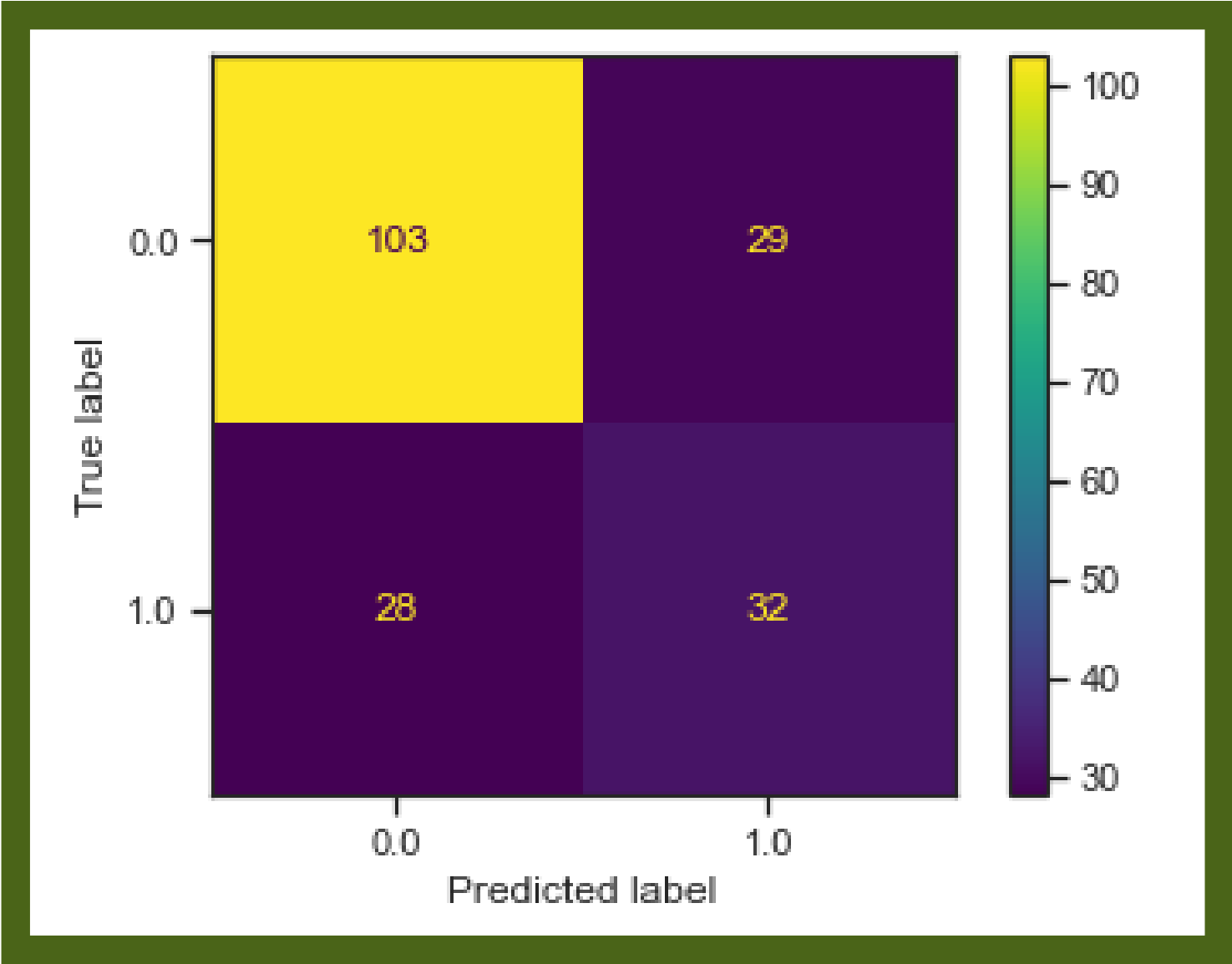
Confusion Matrix and Classification Report



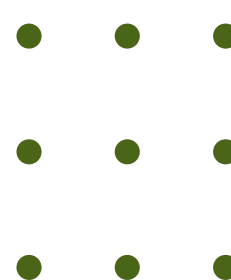
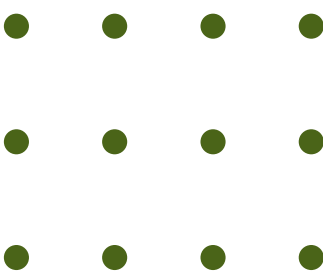
Decision Tree

Confusion Matrix and Classification Report

- People who are truly negative and detected negative are 100 people.
- People who are truly negative but detected positive are 29 people.
- People who are truly positive and detected positive are 32 people.
- People who are truly positive but detected negative are 28 people.



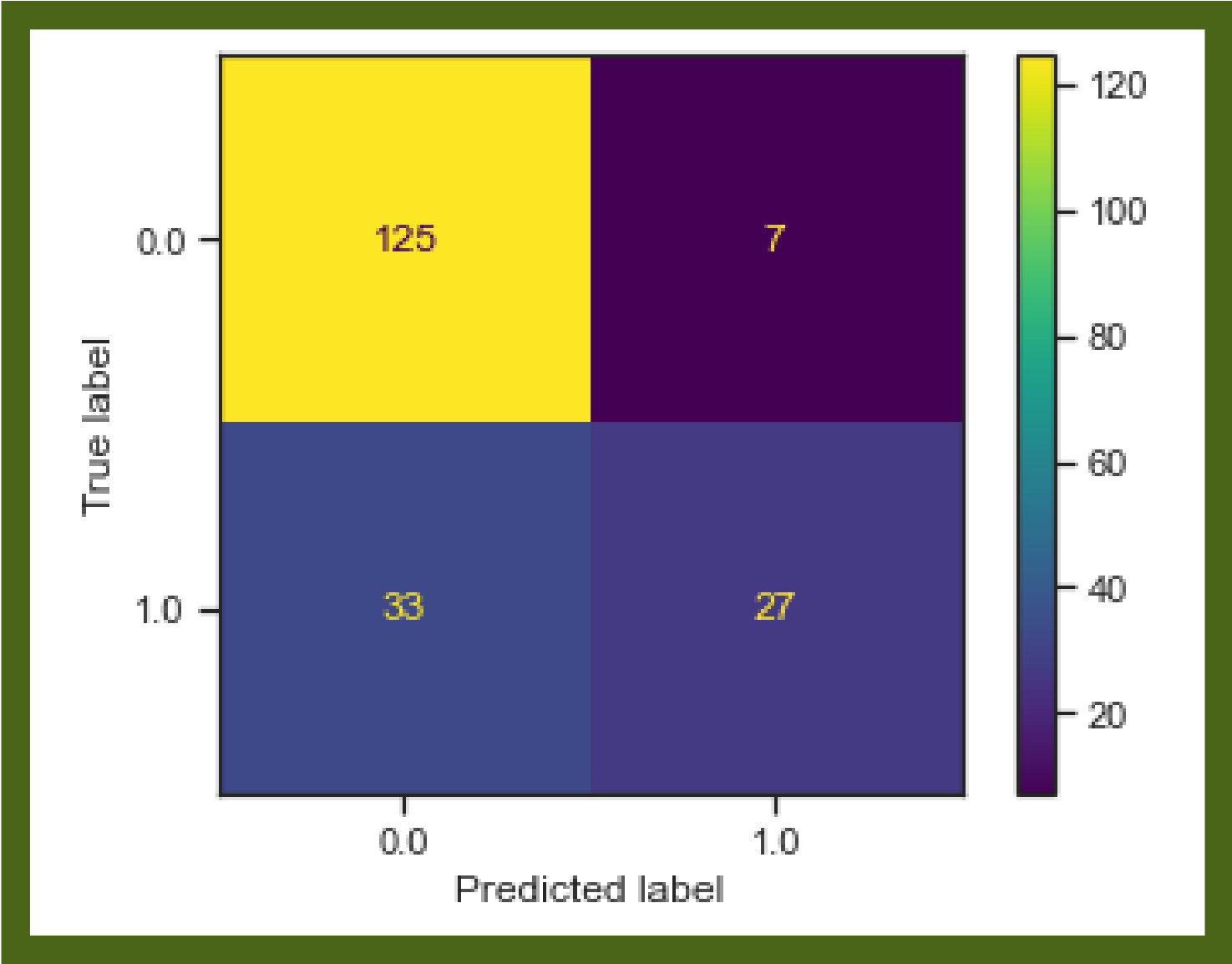
	precision	recall	f1-score	support
0.0	0.79	0.78	0.78	132
1.0	0.52	0.53	0.53	60
accuracy			0.70	192
macro avg	0.66	0.66	0.66	192
weighted avg	0.70	0.70	0.70	192



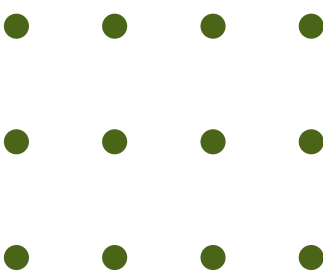
Logistic Regression

Confusion Matrix and Classification Report

- People who are truly negative and detected negative are 125 people.
- People who are truly negative but detected positive are 7 people.
- People who are truly positive and detected positive are 27 people.
- People who are truly positive but detected negative are 33 people.



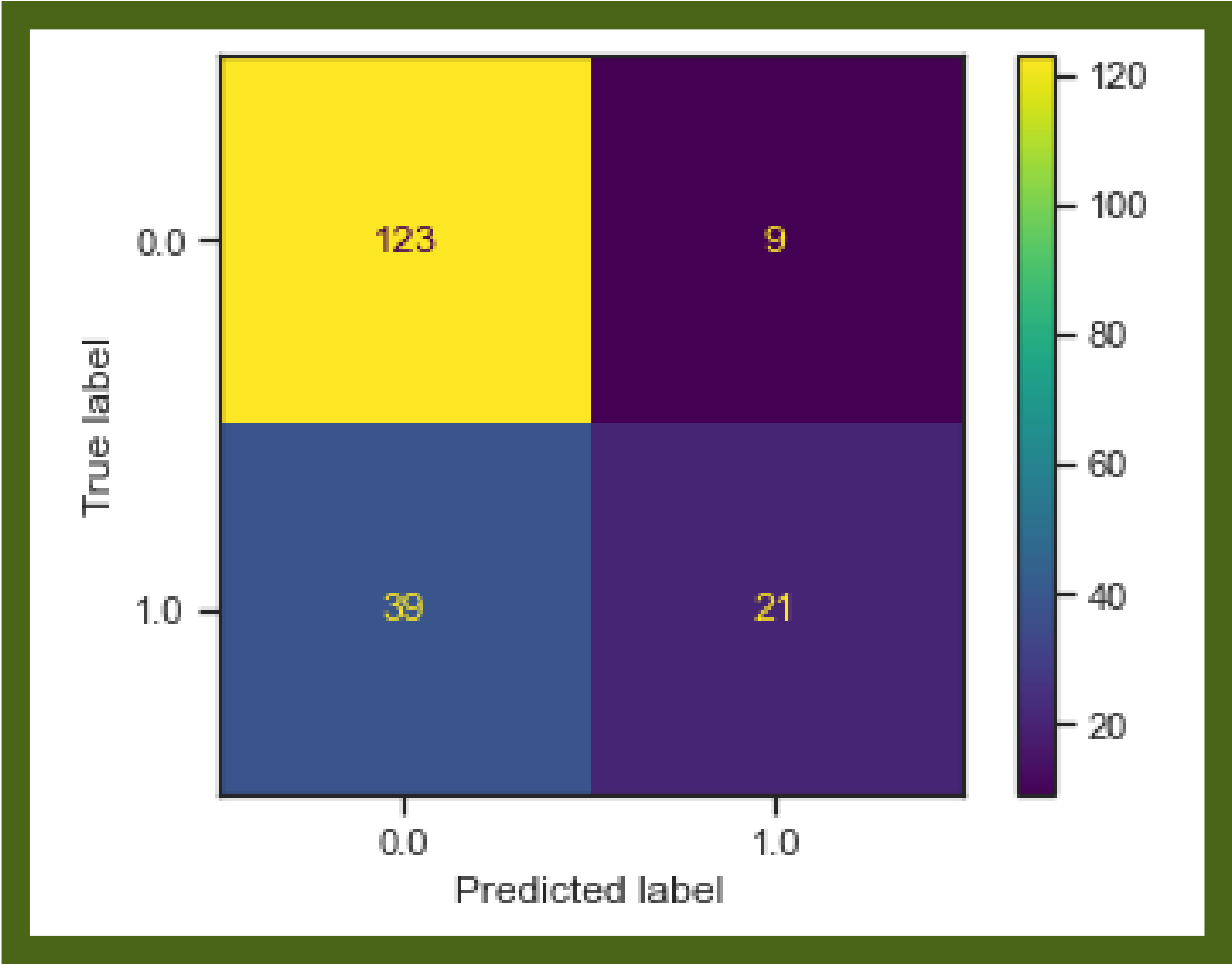
	precision	recall	f1-score	support
0.0	0.79	0.95	0.86	132
1.0	0.79	0.45	0.57	60
accuracy			0.79	192
macro avg	0.79	0.70	0.72	192
weighted avg	0.79	0.79	0.77	192



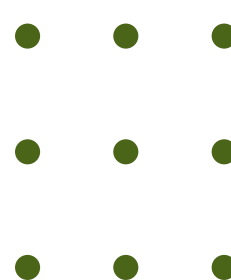
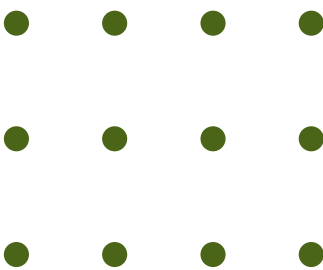
K-Nearest Neighbor (KNN)

Confusion Matrix and Classification Report

- People who are truly negative and detected negative are 123 people.
- People who are truly negative but detected positive are 9 people.
- People who are truly positive and detected positive are 21 people.
- People who are truly positive but detected negative are 39 people.



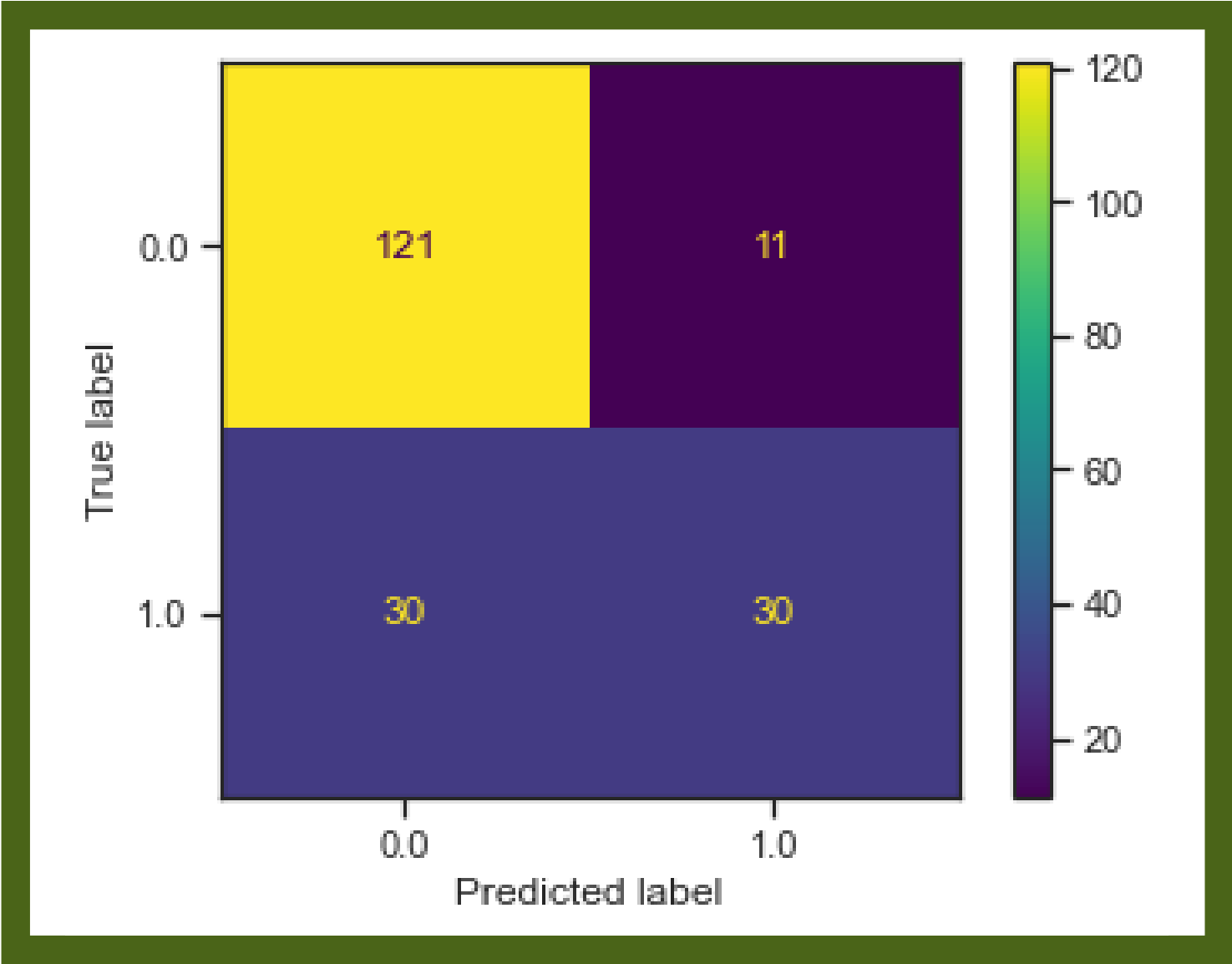
	precision	recall	f1-score	support
0.0	0.76	0.93	0.84	132
1.0	0.70	0.35	0.47	60
accuracy			0.75	192
macro avg	0.73	0.64	0.65	192
weighted avg	0.74	0.75	0.72	192



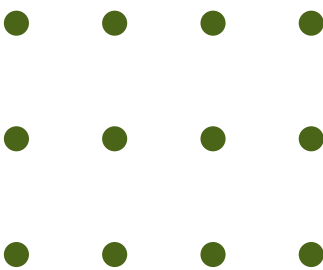
Random Forest

Confusion Matrix and Classification Report

- People who are truly negative and detected negative are 121 people.
- People who are truly negative but detected positive are 11 people.
- People who are truly positive and detected positive are 30 people.
- People who are truly positive but detected negative are 30 people.



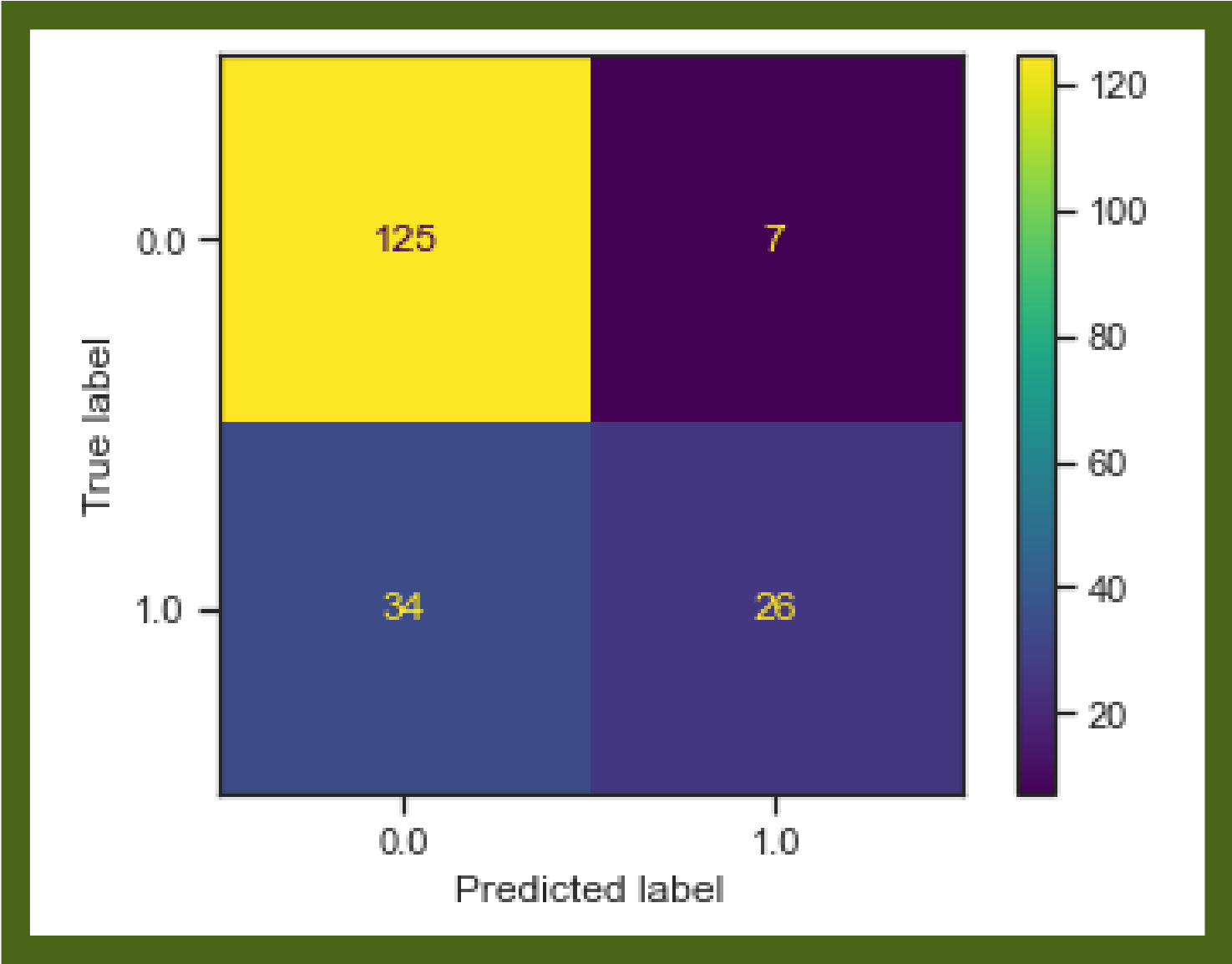
	precision	recall	f1-score	support
0.0	0.80	0.92	0.86	132
1.0	0.73	0.50	0.59	60
accuracy			0.79	192
macro avg	0.77	0.71	0.72	192
weighted avg	0.78	0.79	0.77	192



Support Vector Machine

Confusion Matrix and Classification Report

- People who are truly negative and detected negative are 125 people.
- People who are truly negative but detected positive are 7 people.
- People who are truly positive and detected positive are 26 people.
- People who are truly positive but detected negative are 34 people.

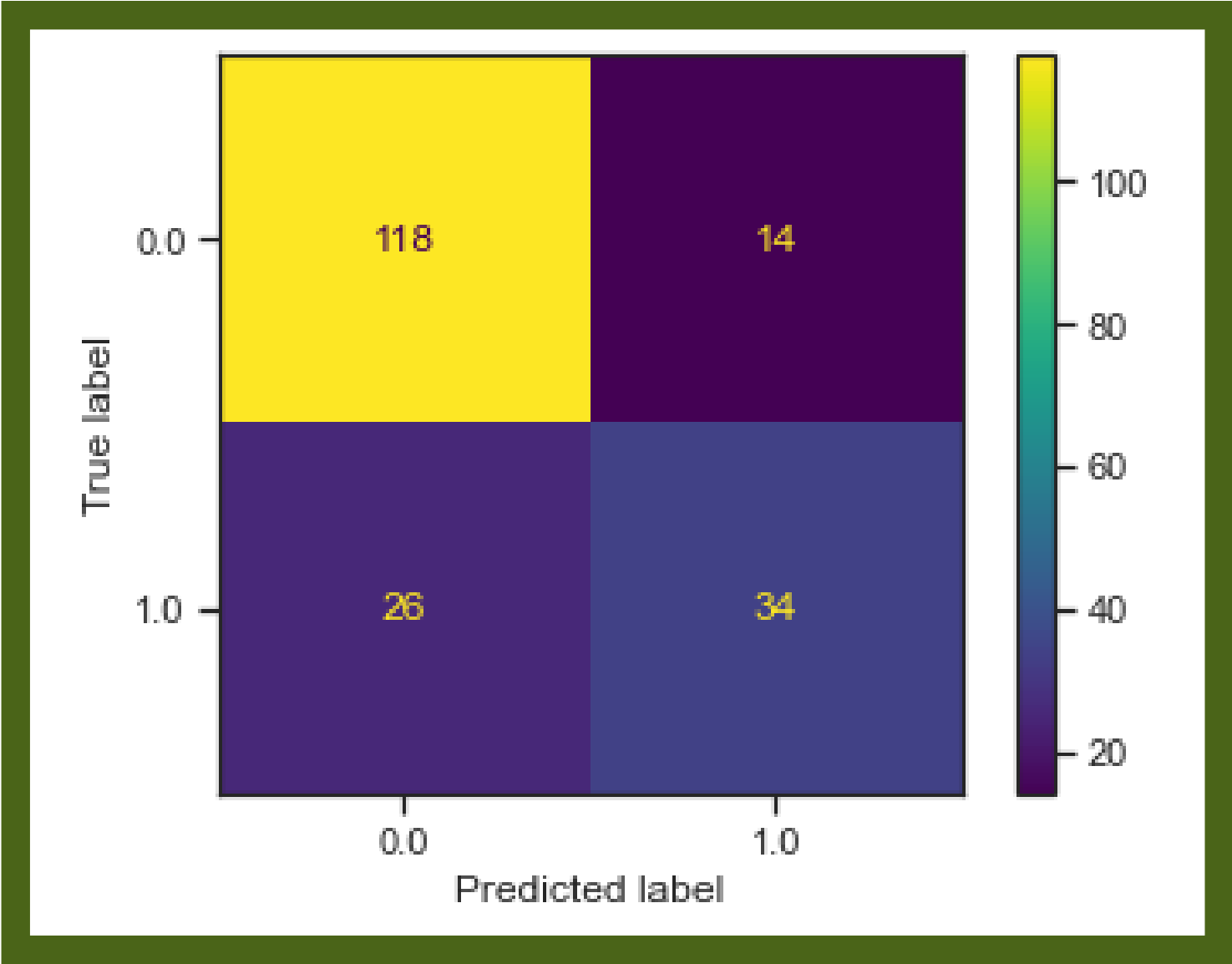


	precision	recall	f1-score	support
0.0	0.79	0.95	0.86	132
1.0	0.79	0.43	0.56	60
accuracy			0.79	192
macro avg	0.79	0.69	0.71	192
weighted avg	0.79	0.79	0.77	192

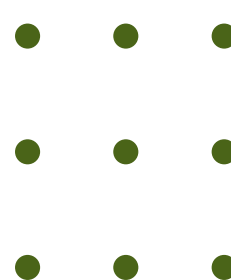
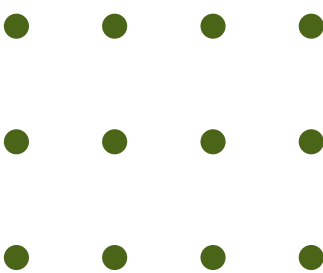
Naïve Bayes

Confusion Matrix and Classification Report

- People who are truly negative and detected negative are 118 people.
- People who are truly negative but detected positive are 14 people.
- People who are truly positive and detected positive are 34 people.
- People who are truly positive but detected negative are 26 people.



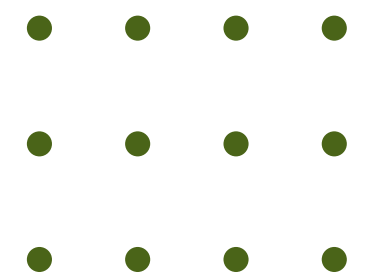
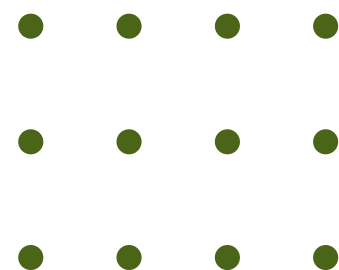
	precision	recall	f1-score	support
0.0	0.82	0.89	0.86	132
1.0	0.71	0.57	0.63	60
accuracy			0.79	192
macro avg	0.76	0.73	0.74	192
weighted avg	0.78	0.79	0.78	192

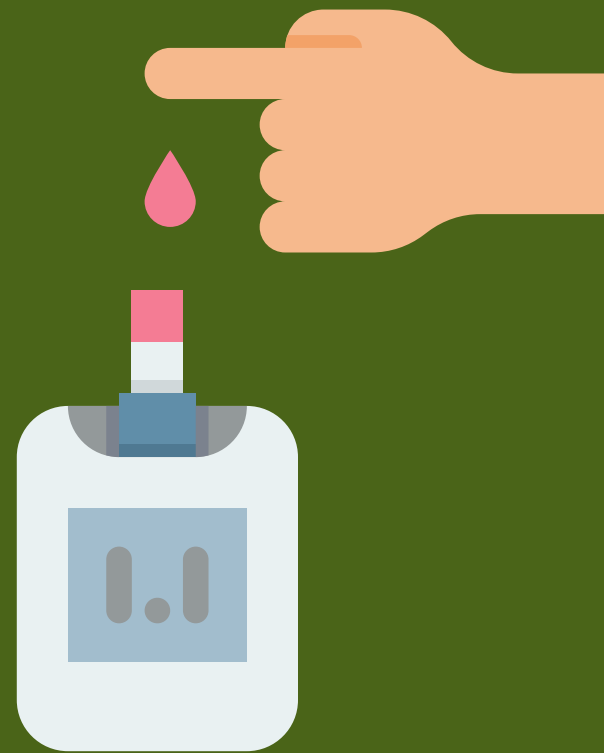


CONCLUSION

Naïve Bayes model is the best model for this dataset.

- Naïve Bayes model tend to have high accuracy, high precision, high recall, and high f1-score than other models.





THANK YOU

For more information

https://github.com/Yohannes-Alexander/machine_learning_project/tree/main/Project_Diabetes

