

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
% matplotlib inline
```

```
df = pd.read_csv("/content/drive/MyDrive/Datasets/diabetes.csv")
df.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigree
0	6	148	72	35	0	33.6	
1	1	85	66	29	0	26.6	
2	8	183	64	0	0	23.3	
3	1	89	66	23	94	28.1	
4	0	137	40	35	168	43.1	

```
y=df['Outcome']
X=df.drop(['Outcome'],axis=1)
```

```
df.shape
```

```
(768, 9)
```

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25,random_state=30)
```

LOGISTIC REGRESSION

```
from sklearn.linear_model import LogisticRegression
reg=LogisticRegression()
reg.fit(X_train,y_train)
y_pred=reg.predict(X_test)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:818: Converge
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

Increase the number of iterations (max_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,
```

```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, y_pred)
```

```
array([[113, 17],
       [ 24, 38]])
```

```
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	0.82	0.87	0.85	130
1	0.69	0.61	0.65	62
accuracy			0.79	192
macro avg	0.76	0.74	0.75	192
weighted avg	0.78	0.79	0.78	192

DECISION TREE

```
from sklearn import tree
clf = tree.DecisionTreeClassifier()
clf.fit(X_train, y_train)
y_pred=clf.predict(X_test)
```

```
confusion_matrix(y_test, y_pred)
```

```
array([[103, 27],
       [ 26, 36]])
```

```
print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	0.80	0.79	0.80	130
1	0.57	0.58	0.58	62
accuracy			0.72	192
macro avg	0.68	0.69	0.69	192
weighted avg	0.73	0.72	0.72	192

SVM

```
from sklearn import svm
clf = svm.SVC(kernel='linear')
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
```

```
confusion_matrix(y_test, y_pred)
```

```
array([[116, 14],
```

```
[ 25,  37]])
```

```
print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	0.82	0.89	0.86	130
1	0.73	0.60	0.65	62
accuracy			0.80	192
macro avg	0.77	0.74	0.76	192
weighted avg	0.79	0.80	0.79	192

NAIVE BAYES

```
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb.fit(X_train, y_train)
y_pred = gnb.predict(X_test)
```

```
confusion_matrix(y_test, y_pred)
```

```
array([[114,  16],
       [ 24,  38]])
```

```
print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	0.83	0.88	0.85	130
1	0.70	0.61	0.66	62
accuracy			0.79	192
macro avg	0.76	0.74	0.75	192
weighted avg	0.79	0.79	0.79	192