

Importing the Packages

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.metrics import accuracy_score
```

```
In [2]: df = pd.read_csv('diabetes.csv') #Dataframe
```

Dataset

```
In [3]: df.head(7)
```

```
Out[3]:
```

	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
5	5	116	74	0	0	25.6	0.201	30	0
6	3	78	50	32	88	31.0	0.248	26	1

Correlation Matrix

```
In [4]: corrMatrix = df.corr()
df.corr()
```

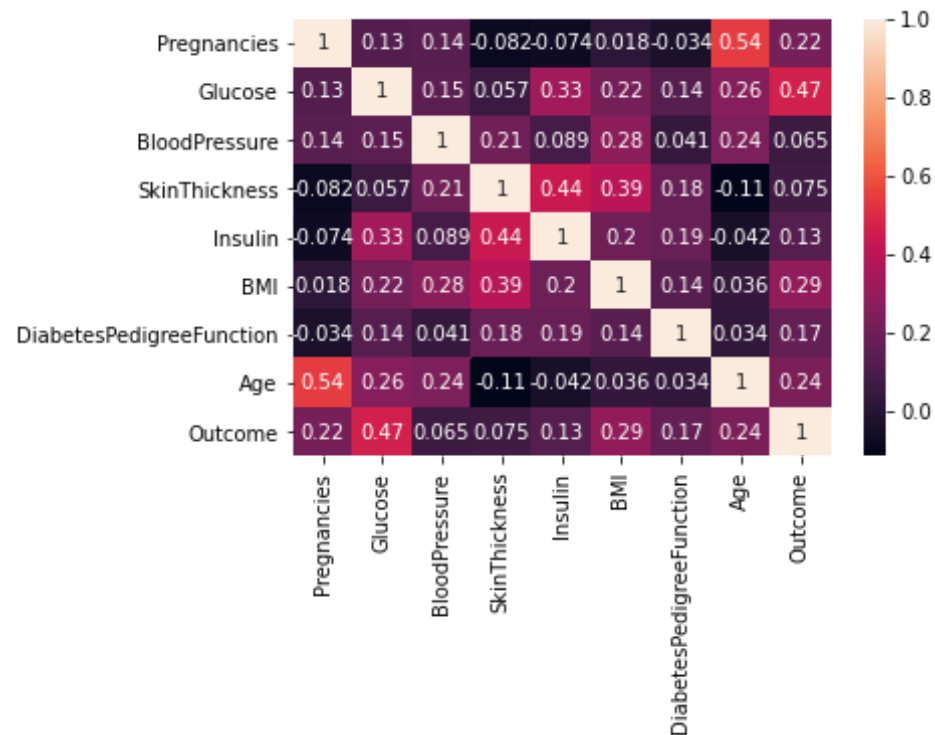
```
Out[4]:
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1.000000	0.129459	0.141282	-0.081672	-0.073535	0.017683	-0.033523	0.544341	0.221896

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Glucose	0.129459	1.000000	0.152590	0.057328	0.331357	0.221071	0.137337	0.263514	0.466581
BloodPressure	0.141282	0.152590	1.000000	0.207371	0.088933	0.281805	0.041265	0.239528	0.065068
SkinThickness	-0.081672	0.057328	0.207371	1.000000	0.436783	0.392573	0.183928	-0.113970	0.074752
Insulin	-0.073535	0.331357	0.088933	0.436783	1.000000	0.197859	0.185071	-0.042163	0.130548
BMI	0.017683	0.221071	0.281805	0.392573	0.197859	1.000000	0.140647	0.036242	0.292695
DiabetesPedigreeFunction	-0.033523	0.137337	0.041265	0.183928	0.185071	0.140647	1.000000	0.033561	0.173844
Age	0.544341	0.263514	0.239528	-0.113970	-0.042163	0.036242	0.033561	1.000000	0.238356
Outcome	0.221898	0.466581	0.065068	0.074752	0.130548	0.292695	0.173844	0.238356	1.000000

Visual Representation of Confusion Matrix

```
In [5]: sns.heatmap(corrMatrix, annot=True)
plt.show()
```



```
In [6]: ### Features and Output
```

```
In [7]: X = df.iloc[:,0:8].values
        y = df.iloc[:,8].values
```

Test/Train Splitting

```
In [8]: from sklearn.model_selection import train_test_split
        X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.20)
```

Creating Instance of RFC

```
In [9]: from sklearn.ensemble import RandomForestClassifier
        clf1 = RandomForestClassifier(
            n_estimators=20,
            criterion='entropy',
```

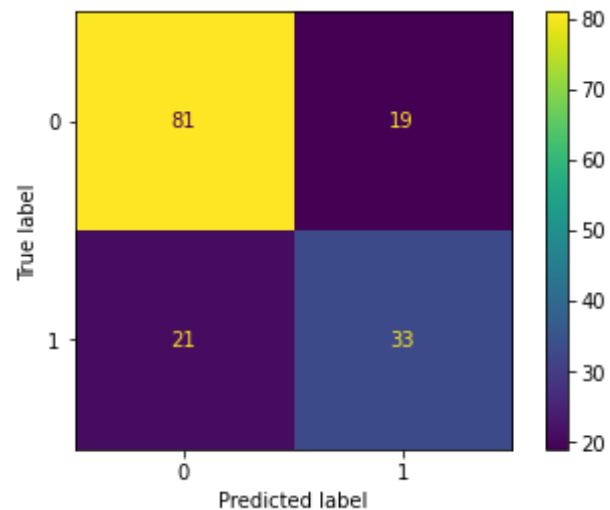
```
max_depth=10,  
random_state=42)  
clf1.fit(X=X_train, y=y_train)
```

```
Out[9]: RandomForestClassifier(criterion='entropy', max_depth=10, n_estimators=20,  
random_state=42)
```

```
In [10]: y_pred = clf1.predict(X=X_test)
```

Confusion Matrix

```
In [11]: from sklearn.metrics import plot_confusion_matrix  
plot_confusion_matrix(clf1, X_test, y_test)  
plt.show()
```



Accuracy Score

```
In [12]: print(f'{round(accuracy_score(y_test, y_pred)*100,2)}%')
```

74.03%

```
In [ ]:
```