K-Nearest Neighbor algorithm

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
plt.style.use('ggplot')
```

```
df = pd.read_csv("diabetes.csv")
df.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Diab
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	

```
df.shape
```

```
(768, 9)
```

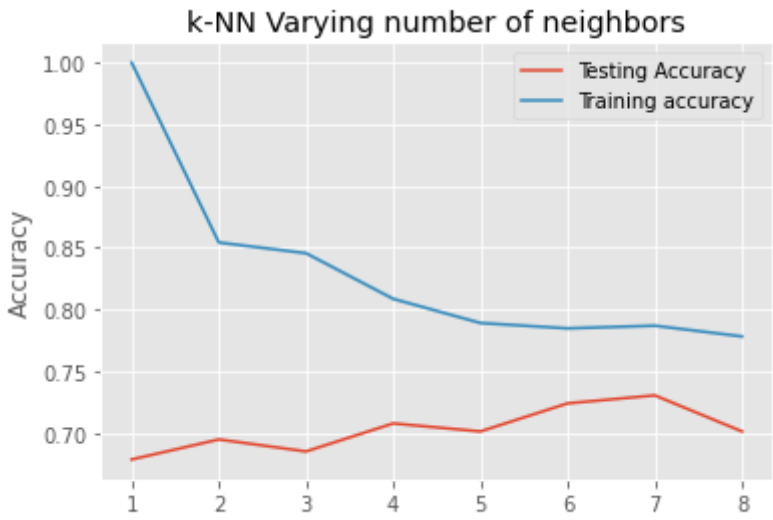
```
X = df.drop('Outcome',axis=1).values
y = df['Outcome'].values
```

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.4,random_state=42, strati
```

```
from sklearn.neighbors import KNeighborsClassifier

neighbors = np.arange(1,9)
train_accuracy = np.empty(len(neighbors))
test_accuracy = np.empty(len(neighbors))

for i,k in enumerate(neighbors):
    knn = KNeighborsClassifier(n_neighbors=k)

    knn.fit(X_train, y_train)

    train_accuracy[i] = knn.score(X_train, y_train)

    test_accuracy[i] = knn.score(X_test, y_test)
```

```
plt.title('k-NN Varying number of neighbors')
```

```
plt.plot(neighbors, test_accuracy, label='Testing Accuracy')
plt.plot(neighbors, train_accuracy, label='Training accuracy')
plt.legend()
plt.xlabel('Number of neighbors')
plt.ylabel('Accuracy')
plt.show()
```




```
knn = KNeighborsClassifier(n_neighbors=7)
```

```
knn.fit(X_train,y_train)
```

```
KNeighborsClassifier(n_neighbors=7)
```

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=7, p=2,
           weights='uniform')
```

```
KNeighborsClassifier(n_jobs=1, n_neighbors=7)
```

```
knn.score(X_test,y_test)
```

```
0.7305194805194806
```

```
from sklearn.metrics import confusion_matrix
```

```
y_pred = knn.predict(X_test)
```

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

```
array([[165,  36],
       [ 47,  60]])
```

```
pd.crosstab(y_test, y_pred, rownames=['True'], colnames=['Predicted'], margins=True)
```

Predicted	0	1	All
True			
0	165	36	201
1	47	60	107
All	212	96	308

```
from sklearn.metrics import classification_report
```

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

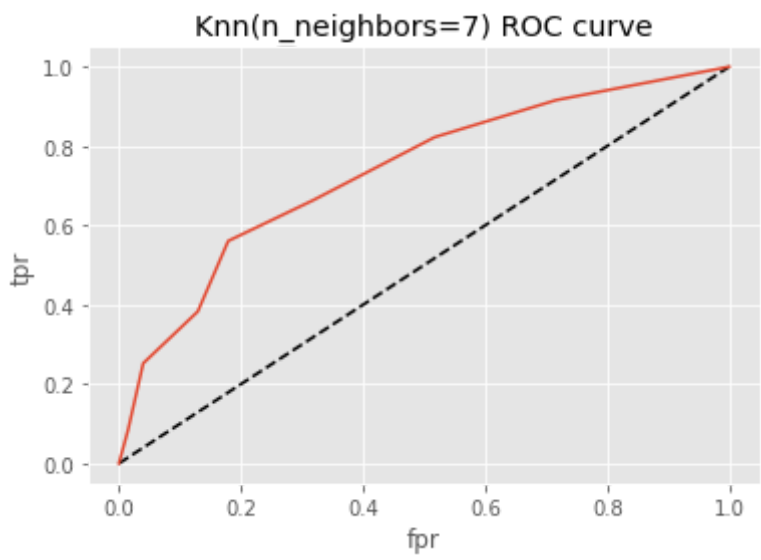
```
              precision    recall  f1-score   support

           0       0.78      0.82      0.80       201
           1       0.62      0.56      0.59       107

    accuracy                           0.73       308
   macro avg       0.70      0.69      0.70       308
weighted avg       0.73      0.73      0.73       308
```

```
y_pred_proba = knn.predict_proba(X_test)[:,1]
```

```
from sklearn.metrics import roc_curve
```

```
fpr, tpr, thresholds = roc_curve(y_test, y_pred_proba)
```

```
plt.plot([0,1],[0,1],'k--')
plt.plot(fpr,tpr, label='Knn')
plt.xlabel('fpr')
plt.ylabel('tpr')
plt.title('Knn(n_neighbors=7) ROC curve')
plt.show()
```



```
from sklearn.metrics import roc_auc_score
roc_auc_score(y_test,y_pred_proba)
```

```
0.7345050448691124
```