

▼ Assignment 5

KNN algorithm on diabetes dataset

```
1 import pandas as pd
2 import numpy as np
3 import seaborn as sns
4 import matplotlib.pyplot as plt
5 %matplotlib inline
6 import warnings
7 warnings.filterwarnings('ignore')
8 from sklearn.model_selection import train_test_split
9 from sklearn.svm import SVC
10 from sklearn import metrics
```

```
1 df=pd.read_csv('diabetes.csv')
```

```
1 df.columns
```

```
[> Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
        'BMI', 'Pedigree', 'Age', 'Outcome'],
        dtype='object')
```

Check for null values. If present remove null values from the dataset

```
1 df.isnull().sum()
```

```
Pregnancies    0
Glucose         0
BloodPressure   0
SkinThickness   0
Insulin         0
BMI             0
Pedigree        0
Age            0
Outcome         0
dtype: int64
```

```
1
```

Outcome is the label/target, other columns are features

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

```
1 from sklearn.preprocessing import scale
2 X = scale(X)
3 # split into train and test
4 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 42)
```

```

1 from sklearn.neighbors import KNeighborsClassifier
2 knn = KNeighborsClassifier(n_neighbors=7)
3
4 knn.fit(X_train, y_train)
5 y_pred = knn.predict(X_test)

1 print("Confusion matrix: ")
2 cs = metrics.confusion_matrix(y_test,y_pred)
3 print(cs)

Confusion matrix:
[[123  28]
 [ 37  43]]

1 print("Accuracy ",metrics.accuracy_score(y_test,y_pred))

Accuracy  0.7186147186147186

```

Classification error rate: proportion of instances misclassified over the whole set of instances. Error rate is calculated as the total number of two incorrect predictions (FN + FP) divided by the total number of a dataset (examples in the dataset).

Also $\text{error_rate} = 1 - \text{accuracy}$

```

1 total_misclassified = cs[0,1] + cs[1,0]
2 print(total_misclassified)
3 total_examples = cs[0,0]+cs[0,1]+cs[1,0]+cs[1,1]
4 print(total_examples)
5 print("Error rate",total_misclassified/total_examples)
6 print("Error rate ",1-metrics.accuracy_score(y_test,y_pred))

65
231
Error rate 0.2813852813852814
Error rate  0.2813852813852814

1 print("Precision score",metrics.precision_score(y_test,y_pred))

Precision score 0.6056338028169014

1 print("Recall score ",metrics.recall_score(y_test,y_pred))

Recall score  0.5375

```

```

1 print("Classification report ",metrics.classification_report(y_test,y_pred))

```

Classification report			precision	recall	f1-score	support
0	0.77	0.81	0.79	151		
1	0.61	0.54	0.57	80		
accuracy			0.72	231		
macro avg	0.69	0.68	0.68	231		
weighted avg	0.71	0.72	0.71	231		

Colab paid products - [Cancel contracts here](#)

