Ishan Gupta - K-Nearest Neighbors - 19BCE7467

```
[7] import numpy as np
    import matplotlib.pyplot as plt
    import pandas as pd
    dataset = pd.read_csv('/content/diabetes.csv')
    print(dataset)
```

```
     Pregnancies  Glucose  ...  Age  Outcome
0              6      148  ...   50        1
1              1       85  ...   31        0
2              8      183  ...   32        1
3              1       89  ...   21        0
4              0      137  ...   33        1
..           ...      ...  ...  ...      ...
763           10      101  ...   63        0
764            2      122  ...   27        0
765            5      121  ...   30        0
766            1      126  ...   47        1
767            1       93  ...   23        0

[768 rows x 9 columns]
```

```
[8] X = dataset.iloc[:, :-1].values
    y = dataset.iloc[:, -1].values
```

```
[9] # Splitting the dataset into the Training set and Test set
    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 = 0)
```

```
[10] from sklearn.preprocessing import StandardScaler
     sc = StandardScaler()
     X_train = sc.fit_transform(X_train)
     X_test = sc.transform(X_test)
```

```
[11] # Training the K-NN model on the Training set
     from sklearn.neighbors import KNeighborsClassifier
     classifier = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
     classifier.fit(X_train, y_train)
```

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=5, p=2,
                     weights='uniform')
```

```
[12] # Making the Confusion Matrix
     from sklearn.metrics import confusion_matrix, accuracy_score
     y_pred = classifier.predict(X_test)
     cm = confusion_matrix(y_test, y_pred)
     print(cm)
     accuracy_score(y_test, y_pred)
```

```
[[114  16]
 [ 22  40]]
0.8020833333333334
```

Ishan Gupta - Weighted K-Nearest Neighbors - 19BCE7467

```
[13] import numpy as np
     import matplotlib.pyplot as plt
     import pandas as pd
     dataset = pd.read_csv('/content/diabetes.csv')
     print(dataset)
```

```
     Pregnancies  Glucose  ...  Age  Outcome
0              6      148  ...   50        1
1              1       85  ...   31        0
2              8      183  ...   32        1
3              1       89  ...   21        0
4              0      137  ...   33        1
..           ...      ...  ...  ...      ...
763           10      101  ...   63        0
764            2      122  ...   27        0
765            5      121  ...   30        0
766            1      126  ...   47        1
767            1       93  ...   23        0

[768 rows x 9 columns]
```

```
[14] X = dataset.iloc[:, :-1].values
     y = dataset.iloc[:, -1].values
```

```
[15] # Splitting the dataset into the Training set and Test set
     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 = 0)
```


```
[15] # Splitting the dataset into the Training set and Test set
     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 = 0)
```


```
[16] from sklearn.preprocessing import StandardScaler
     sc = StandardScaler()
     X_train = sc.fit_transform(X_train)
     X_test = sc.transform(X_test)
```

```
[18] # Training the K-NN model on the Training set
     from sklearn.neighbors import KNeighborsClassifier
     classifier = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2, weights='distance')
     classifier.fit(X_train, y_train)
```

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=5, p=2,
                     weights='distance')
```

```
[19] # Making the Confusion Matrix
     from sklearn.metrics import confusion_matrix, accuracy_score
     y_pred = classifier.predict(X_test)
     cm = confusion_matrix(y_test, y_pred)
     print(cm)
     accuracy_score(y_test, y_pred)
```

```
[[113  17]
 [ 22  40]]
0.796875
```