Machine Learning Assignment

Karn Arora (J078)

Exp 10:- K-Nearest Neighbour from Scratch

Importing Libraries

```
import numpy as np
import pandas as pd
from sklearn import datasets
from collections import Counter
```

```
iris = datasets.load_iris()
Species = iris.target
data = pd.DataFrame(np.c_[iris.data, Species.reshape((Species.shape[0],1))], columns = iri
data.head()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	Species
0	5.1	3.5	1.4	0.2	0.0
[illegible]	[illegible]	3.0	1.4	0.2	0.0
[illegible]	[illegible]	3.2	1.3	0.2	0.0
3	4.6	3.1	1.5	0.2	0.0
4	5.0	3.6	1.4	0.2	0.0

Saved successfully! ×

```
data['Species'].value_counts()
```

```
2.0    50
1.0    50
0.0    50
Name: Species, dtype: int64
```

Using K-Nearest Neighbour function

```
from sklearn.model_selection import train_test_split
train, test = train_test_split(data, test_size = 0.2, random_state = 0)
```

```
class knn():
    def __init__(self,X, Y, k_neighbors):
        self.k_neighbors = k_neighbors
        self.X_train = X
        self.Y_train = Y
        self.target = set(Y)

 # calculating euclidean distance
    def euclidean_distance(self,row1,row2):
        distance = 0.0
        for i in range(len(row1)):
            distance += (row1[i]-row2[i])**2
        return np.sqrt(distance)

    def sort_distance(self,r):
        return r[2]

    def get_neighbors(self,row):
        dist = []
        for row_index in range(len(self.X_train)):
            d = self.euclidean_distance(self.X_train.iloc[row_index,:], row)
            dist.append((self.X_train.iloc[row_index,:],self.Y_train.iloc[row_index],d))
        dist.sort(key = self.sort_distance)

        neighbors = []
        for i in range(self.k_neighbors):
            neighbors.append(dist[i][1])
        return neighbors

    def predict(self,row):
        neigh = self.get_neighbors(row)
        neighbors = Counter(neigh)
        count = 0
```

Saved successfully!

```
            if neighbors[i]>count:
                count = neighbors[i]
                pred = i
        return pred
```

```
Y = train['Species']
X = train.drop('Species',axis = 1)
clf = knn(X, Y, 5)
X.loc[0,:]
```

```
sepal length (cm)    5.1
sepal width (cm)     3.5
petal length (cm)    1.4
petal width (cm)     0.2
Name: 0, dtype: float64
```

```
predictions = []
Y_test = test['Species']
X_test = test.drop('Species',axis = 1)
for row in range(len(X_test)):
    pred = clf.predict(X_test.iloc[row,:])
    predictions.append(pred)
```

```
from sklearn.metrics import accuracy_score
accuracy_score(Y_test,predictions)
```

```
0.9666666666666667
```

Using K-Nearest Neighbour in Scikit Learn

```
from sklearn.neighbors import KNeighborsClassifier
neigh = KNeighborsClassifier(n_neighbors=3)
neigh.fit(X,Y)
pred1=neigh.predict(X_test)
accuracy_score(Y_test,pred1)
```

```
0.9666666666666667
```

Saved successfully!