

Assignment No. 2

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.neighbors import NearestCentroid
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn import datasets

dataset=pd.read_csv("/content/assignment2MLdataset - Sheet1.csv")
X=dataset.iloc[:, :-1].values
y=dataset.iloc[:, -1].values
print(X,y)

[[2 4]
 [4 6]
 [4 4]
 [4 2]
 [6 4]
 [6 2]] ['negative' 'negative' 'positive' 'negative' 'negative' 'positive']

classifier=KNeighborsClassifier(n_neighbors=3)
classifier.fit(X,y)

KNeighborsClassifier(n_neighbors=3)

# predicting given input point with general KNN
X_test=np.array([6,6])
y_pred=classifier.predict([X_test])
print("general KNN: ",y_pred)

general KNN:  ['negative']

weighted_classifier=KNeighborsClassifier(n_neighbors=3, weights="distance")
weighted_classifier.fit(X,y)

KNeighborsClassifier(n_neighbors=3, weights='distance')

# predicting given input point with weighted KNN
# X_test=np.array([6,6])
y_pred=weighted_classifier.predict([X_test])
print("weighted KNN: ",y_pred)

weighted KNN:  ['negative']

ncentroid=NearestCentroid()
ncentroid.fit(X,y)

NearestCentroid()

#predicting given input point with nearest centroid
```

```
n_pred=ncentroid.predict([X_test])
print("Nearest Centroid: ", n_pred)
```

```
Nearest Centroid: ['negative']
```

```
# Passing whole dataset to general KNN
y_pred_g=classifier.predict(X)
print("general KNN: ",y_pred_g)
```

```
general KNN: ['negative' 'negative' 'negative' 'positive' 'positive' 'negative']
```

```
# Passing whole dataset to weighted KNN
y_pred_w=weighted_classifier.predict(X)
print("weighted KNN: ",y_pred_w)
print("actual output: " ,y)
```

```
weighted KNN: ['negative' 'negative' 'positive' 'negative' 'negative' 'positive']
actual output: ['negative' 'negative' 'positive' 'negative' 'negative' 'positive']
```

```
#passing whole dataset to nearest centroid
y_pred_n=ncentroid.predict(X)
print("Nearest Centroid: ", y_pred_n)
```

```
Nearest Centroid: ['negative' 'negative' 'negative' 'positive' 'positive' 'positive']
```

```
print("Accuracy score of general KNN is: ", accuracy_score(y,y_pred_g))
print("Accuracy score of weighted KNN is: ", accuracy_score(y,y_pred_w))
```

```
Accuracy score of general KNN is: 0.3333333333333333
Accuracy score of weighted KNN is: 1.0
```

```
print("Confusion matrix for general KNN is: ", confusion_matrix(y,y_pred_g))
print("Confusion matrix for weighted KNN is: ", confusion_matrix(y,y_pred_w))
```

```
Confusion matrix for general KNN is: [[2 2]
[2 0]]
```

```
Confusion matrix for weighted KNN is: [[4 0]
[0 2]]
```