

Name:- Muhammad Huzaifa Waseem (2303-KHI-DEG-021)
Pair Partner 1:- Muhammad Faizan Rafique (2303.005.KHI.DEG)
Pair Partner 2:- Syed Muhammad Hammad Irshad(2303.KHI.DEG.032)

UNIT 3.2:

Assignment

K-NN Algorithm:-

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets
from sklearn.metrics import f1_score, mean_squared_error, recall_score
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

from sklearn.neighbors import KNeighborsClassifier

wine = datasets.load_wine()
wine_x = wine.data[:, 2:]
wine_y = wine.target
wine_y

array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2])

x_train, x_test, y_train, y_test = train_test_split(wine_x, wine_y, test_size=0.3)
scaler = StandardScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)

model = KNeighborsClassifier()

model.fit(x_train, y_train)

▼ KNeighborsClassifier
KNeighborsClassifier()

y_pred = model.predict(x_test)
f1_score(y_test, y_pred, average="micro")

0.9444444444444444
```

Voting Classifier:-

Hard Voting:-

Voting classifier

```
from sklearn.ensemble import VotingClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn import datasets
from sklearn.metrics import f1_score, mean_squared_error, recall_score
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

wine = datasets.load_wine()
wine_x = wine.data[:, 2:]
wine_y = wine.target

x_train, x_test, y_train, y_test = train_test_split(wine_x, wine_y, test_size=0.3)

scaler = StandardScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)

dt_model = DecisionTreeClassifier()
svm_model = SVC()
log_model = LogisticRegression()

voting_model = VotingClassifier(
    estimators=[('dt', dt_model),
                ('svm', svm_model),
                ('lr', log_model)],
    voting='hard')
voting_model.fit(x_train, y_train)
for model in (dt_model, svm_model, log_model, voting_model):
    model.fit(x_train, y_train)
    y_pred = model.predict(x_test)
    print(model.__class__.__name__, f1_score(y_test, y_pred, average="micro"))

DecisionTreeClassifier 0.9814814814814815
SVC 0.9814814814814815
LogisticRegression 0.9814814814814815
VotingClassifier 0.9814814814814815
```

Soft Voting:-

Voting classifier

```
from sklearn.ensemble import VotingClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn import datasets
from sklearn.metrics import f1_score, mean_squared_error, recall_score
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

wine = datasets.load_wine()
wine_x = wine.data[:, 2:]
wine_y = wine.target

x_train, x_test, y_train, y_test = train_test_split(wine_x, wine_y, test_size=0.3)

scaler = StandardScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)

dt_model = DecisionTreeClassifier()
log_model = LogisticRegression()

voting_model = VotingClassifier(
    estimators=[('dt', dt_model),
                ('lr', log_model)],
    voting='soft')
voting_model.fit(x_train, y_train)
for model in (dt_model, log_model, voting_model):
    model.fit(x_train, y_train)
    y_pred = model.predict(x_test)
    print(model.__class__.__name__, f1_score(y_test, y_pred, average="micro"))

DecisionTreeClassifier 0.9814814814814815
LogisticRegression 0.9814814814814815
VotingClassifier 0.9814814814814815
```