

ML SUBMISSION 10

IMPLEMENTATION OF KNN FROM SCRATCH

J042 AVANI NARVEKAR

IMPORTS

```
In [1]: import numpy as np
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
```

```
In [2]: iris = load_iris()
```

```
In [3]: X = iris.data
y = iris.target
cols = iris.data
```

```
In [4]: X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.2)
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

```
Out[4]: ((120, 4), (30, 4), (120,), (30,))
```

KNN FUNCTION

```
In [5]: ## euclidean distance
def euclidean_dist(p1,p2):
    return np.sqrt(np.sum((p1-p2)**2))
```

```
In [6]: from scipy.stats import mode
from numpy.random import randint
```

```
In [7]: # kNN
def K_NN(x,y,inp,k):
    labels=[]
    for i in inp:
        dist = []
        for j in range(len(x)):
            d = euclidean_dist(np.array(x[j,:]),i)
            dist.append(d)
        dist = np.array(dist)
        dist_upd = np.argsort(dist)[:k]
        l = y[dist_upd]
        lab = mode(l)
        lab = lab.mode[0]
```

```
        labels.append(lab)
    return labels
```

KNN from sklearn

```
In [8]: from sklearn.neighbors import KNeighborsClassifier as KNN
        knnn = KNN(n_neighbors = 5)
        knnn.fit(X_train, y_train)
```

Out[8]: KNeighborsClassifier()

Predictions

```
In [9]: y_pred1 = K_NN(X_train, y_train, X_test, 5)
        print(accuracy_score(y_test, y_pred1))
```

0.9333333333333333

```
In [10]: y_pred2 = knnn.predict(X_test)
         print(accuracy_score(y_test, y_pred2))
```

0.9333333333333333