

WEEK-11

K-Nearest Neighbor Classifier(KNN)

```
import pandas as pd
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
iris_data=pd.read_csv("Iris.csv")
```

```
iris_data.head()
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

```
iris_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 150 entries, 0 to 149
```

```
Data columns (total 5 columns):
```

```
# Column Non-Null Count Dtype
```

```
--- ----
```

```
0 sepal_length 150 non-null float64
```

```
1 sepal_width 150 non-null float64
```

```
2 petal_length 150 non-null float64
```

```
3 petal_width 150 non-null float64
```

```
4 species 150 non-null object
```

```
dtypes: float64(4), object(1)
```

```
memory usage: 6.0+ KB
```

```
iris_data.describe()
```

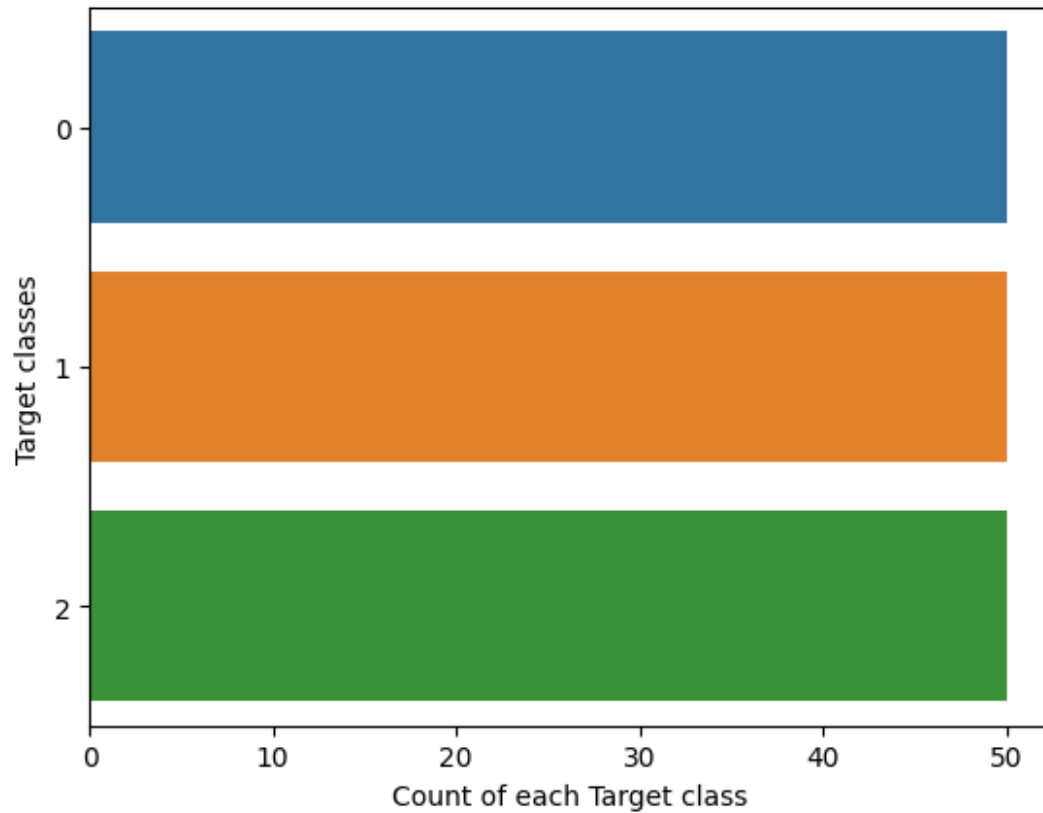
	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

```
iris_data.shape
```

```
(150, 5)
```

```
iris_data['species'].replace({'Iris-setosa':0,'Iris-versicolor':1, 'Iris-virginica':2},inplace=True)
```

```
sns.countplot(y=iris_data['species'],data=iris_data)
plt.xlabel("Count of each Target class")
plt.ylabel("Target classes")
plt.show()
```



```
X = iris_data.drop(['species'],axis=1)
Y = iris_data.species
X.head()
```

	sepal_length	sepal_width	petal_length	petal_width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

```
Y.head()
```

```
0 0
1 0
2 0
3 0
4 0
```

```
Name: species, dtype: int64
```

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
X=scaler.fit_transform(X)

from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.3, random_state=20, stratify=y)

from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(5)
knn.fit(X_train,Y_train)
knn.score(X_test,Y_test)

0.9777777777777777
```

Sample Example

```
X = [[0], [1], [2], [3]]
y = [0, 0, 1, 1]
```

```
from sklearn.neighbors import KNeighborsClassifier
neigh = KNeighborsClassifier(n_neighbors=3)
neigh.fit(X, y)
```

```
KNeighborsClassifier(n_neighbors=3)
```

```
print(neigh.predict([[1.1]]))
```

```
print(neigh.predict_proba([[0.9]]))
```

```
[0]
[[0.66666667 0.33333333]]
```