

K-Nearest Neighbor in Machine Learning

```
In [ ]: # Importing Libraries
import pandas as pd
import numpy as np
import seaborn as sns
import sklearn
```

```
In [ ]: df = sns.load_dataset("iris")
df.head()
```

```
Out[ ]:
```

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

```
In [ ]: df.nunique()
```

```
Out[ ]: sepal_length    35
sepal_width         23
petal_length        43
petal_width         22
species              3
dtype: int64
```

Splitting the data

```
In [ ]: X= df[["sepal_length", "petal_length"]]
y = df["species"]
```

Importing KNeighborsClassifier and fitting it with X and y

```
In [ ]: from sklearn.neighbors import KNeighborsClassifier
model = KNeighborsClassifier(n_neighbors=5).fit(X,y)
```

Predicting outcomes

```
In [ ]: model.predict([[3.1, 1.2]])
```

c:\Users\Salman\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but KNeighborsClassifier was fitted with feature names

warnings.warn(

```
Out[ ]: array(['setosa'], dtype=object)
```

```
In [ ]: model.predict(X)
```

Metrics for evaluation and dividing data into 80/20

```
In [ ]: from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
X_train,X_test,y_train,y_test = train_test_split(X,y, test_size=0.2, random_state=0)
model = KNeighborsClassifier().fit(X_train, y_train)
```

```
In [ ]: predicted_values = model.predict(X_test)
predicted_values
```

```
Out[ ]: array(['virginica', 'versicolor', 'setosa', 'virginica', 'setosa',
               'virginica', 'setosa', 'versicolor', 'versicolor', 'versicolor',
               'virginica', 'versicolor', 'versicolor', 'versicolor',
               'versicolor', 'setosa', 'versicolor', 'versicolor', 'setosa',
               'setosa', 'virginica', 'versicolor', 'setosa', 'setosa',
               'virginica', 'setosa', 'setosa', 'versicolor', 'versicolor',
               'setosa'], dtype=object)
```

```
In [ ]: from sklearn.metrics import accuracy_score
score = accuracy_score(y_test, predicted_values)
score
```

Out[]: 1.0