



Experiment – 2.3

Student Name: Ananya Singla

Branch: BE-CSE

Semester: 5th

Subject Name: Machine Learning Lab

Subject Code: 20CSP-317

UID: 20BCS9957

Section/Group: 604-B

1. Aim/Overview of the practical:

Import any data set and implement K Neighbors Classifier Algorithm to check accuracy in Jupiter Notebook.

2. Software/Hardware Requirements:

- Jupyter Notebook/Google Collab
- Python
- pandas Library
- Standard Dataset

3. Code & Output:

```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

```
# Importing the dataset
dataset = pd.read_csv(r'C:\Users\ANANYA\Social_Network_Ads.csv')
```

```
x= dataset.iloc[:,[2,3]].values
y=dataset.iloc[:, -1].values
dataset.head()
```

```
In [1]: # Importing the Libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

In [2]: # Importing the dataset
dataset = pd.read_csv(r'C:\Users\ANANYA\Social_Network_Ads.csv')

x= dataset.iloc[:,[2,3]].values
y=dataset.iloc[:, -1].values
dataset.head()
```

```
Out[2]:
```

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0

```
from sklearn.model_selection import train_test_split
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
```

```
from sklearn.preprocessing import StandardScaler
```

```
sc=StandardScaler()
```

```
x_train=sc.fit_transform(x_train)
```

```
x_test=sc.transform(x_test)
```

```
# Training the K-NN model on the Training set
```

```
from sklearn.neighbors import KNeighborsClassifier
```

```
classifier = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
```

```
classifier.fit(x_train, y_train)
```

```
#Predicting the Test set results
```

```
y_pred = classifier.predict(x_test)
```

Making the Confusion Matrix

```
from sklearn.metrics import confusion_matrix, accuracy_score
```

```
cm = confusion_matrix(y_test, y_pred)
```

```
ac = accuracy_score(y_test, y_pred)
```

```
cm
```

```
In [3]: from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)

from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
x_train=sc.fit_transform(x_train)
x_test=sc.transform(x_test)
```

```
In [5]: # Training the K-NN model on the Training set
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
classifier.fit(x_train, y_train)

#Predicting the Test set results
y_pred = classifier.predict(x_test)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
ac = accuracy_score(y_test, y_pred)
cm
```

```
Out[5]: array([[55,  3],
               [ 1, 21]], dtype=int64)
```

Learning outcomes (What I have learnt):

1. Learnt about K-NN algorithm.
2. Learnt about confusion matrix.