

Tutorial No.4

Implement following algorithms using Python on suitable data sets. i. K-Nearest neighbour ii. Support Vector Machines.

```
[256]: import pandas as pd
import numpy as np
from sklearn import linear_model
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_iris
from sklearn.metrics import accuracy_score
```

```
[257]: df=pd.read_csv("gender_classification_v3.csv");
```

```
[258]: df.isnull().sum()
```

```
[258]: long_hair      0
forehead_width_cm  0
forehead_height_cm 0
nose_wide          0
nose_long          0
lips_thin          0
distance_nose_to_lip_long 0
gender            0
dtype: int64
```

```
[259]: df
```

```
distance_nose_to_lip_long 0
gender                    0
dtype: int64
```

```
[259]: df
```

```
[259]:
```

	long_hair	forehead_width_cm	forehead_height_cm	nose_wide	nose_long	lips_thin	distance_nose_to_lip_long	gender
0	1	11.5	6.1	1	0	1	1	Male
1	0	14.0	5.4	0	0	1	0	Female
2	0	11.8	6.3	1	1	1	1	Male
3	0	14.4	6.1	0	1	1	1	Male
4	1	13.5	5.9	0	0	0	0	Female
...
4996	1	13.5	5.1	0	0	0	0	Female
4997	1	11.9	5.4	0	0	0	0	Female
4998	1	12.9	5.7	0	0	0	0	Female
4999	1	13.2	6.2	0	0	0	0	Female
5000	1	15.4	5.4	1	1	1	1	Male

5001 rows x 8 columns

```
[260]: X = df[['long_hair', 'forehead_width_cm', 'forehead_height_cm', 'nose_wide', 'nose_long', 'lips_thin', 'distance_nose_to_lip_long']]
y = df['gender']
```

```
[250]: X = df[['long_hair', 'forehead_width_cm', 'forehead_height_cm', 'nose_wide', 'nose_long', 'lips_thin', 'distance_nose_to_lip_long']]
      y = df['gender']
```

```
[251]: df.isnull().sum()
```

```
[251]: long_hair      0
      forehead_width_cm  0
      forehead_height_cm  0
      nose_wide     0
      nose_long     0
      lips_thin     0
      distance_nose_to_lip_long  0
      gender      0
      dtype: int64
```

```
[252]: y
```

```
[252]: 0      Male
      1      Female
      2      Male
      3      Male
      4      Female
      ...
      4596   Female
      4597   Female
      4598   Female
      4599   Female
      5000    Male
      Name: gender, Length: 5001, dtype: object
```

```
[ ]: #
```

```
[ ]: #
```

```
[256]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
      #
      #
```

```
[257]: len(X_train)
```

```
[257]: 4000
```

```
[258]: len(y_train)
```

```
[258]: 4000
```

```
[259]: knn = KNeighborsClassifier(n_neighbors=3)
```

```
[270]: knn.fit(X_train, y_train)
```

```
[270]: <KNeighborsClassifier
      KNeighborsClassifier(n_neighbors=3)
```

```
[271]: y_pred = knn.predict(X_test)
```

```
[272]: accuracy = accuracy_score(y_test, y_pred)
      print("KNN Accuracy:", accuracy)
      KNN Accuracy: 0.97002507002507
```

```
[273]: ee=[1,15,4,6.5,1,1,1,1]
      knn.predict(ee)
```

```
[227]: accuracy = accuracy_score(y_test, y_pred)
       print("KNN Accuracy: ", accuracy)

       KNN Accuracy: 0.8780298370029837

[228]: m = [3, 10, 4, 6, 9, 1, 1, 1]
       knn.predict(m)

       C:\Users\CDPP\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\base.py:493: UserWarning: X does not have valid feature names, but KNeighborsClassifier was fitted with feature names
         warnings.warn(

[229]: array(['Male'], dtype=object)

[230]: from sklearn.svm import SVC
       from sklearn.model_selection import train_test_split
       from sklearn.metrics import accuracy_score

[231]: svm = SVC(kernel='linear')

[232]: svm.fit(X_train, y_train)

[233]: 
       SVC(kernel='linear')

[234]: y_pred = svm.predict(X_test)

[235]: accuracy = accuracy_score(y_test, y_pred)
       print("SVM Accuracy: ", accuracy)

       SVM Accuracy: 0.961038961038961
```


