Mahi Prashant Nakhate

10

In [33]:
```
import pandas as pd
import numpy as np
```

In [34]:
```
df = pd.read_csv('gender.csv')
df.head()
```

Out[34]:

	long_hair	forehead_width_cm	forehead_height_cm	nose_wide	nose_long	lips_thin	distance_nose_to_lip_long	gender
0	1	11.8	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

In [35]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5001 entries, 0 to 5000
Data columns (total 8 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   long_hair                  5001 non-null   int64  
 1   forehead_width_cm          5001 non-null   float64
 2   forehead_height_cm         5001 non-null   float64
 3   nose_wide                  5001 non-null   int64  
 4   nose_long                  5001 non-null   int64  
 5   lips_thin                  5001 non-null   int64  
 6   distance_nose_to_lip_long  5001 non-null   int64  
 7   gender                     5001 non-null   object 
dtypes: float64(2), int64(5), object(1)
memory usage: 312.7+ KB
```

In [36]:
```
df.shape
```

Out[36]:
```
(5001, 8)
```

In [37]:
```
df.describe()
```

Out[37]:

	long_hair	forehead_width_cm	forehead_height_cm	nose_wide	nose_long	lips_thin	distance_nose_to_lip_long
count	5001.000000	5001.000000	5001.000000	5001.000000	5001.000000	5001.000000	5001.000000
mean	0.869626	13.181484	5.946311	0.493901	0.507898	0.493101	0.498900
std	0.336748	1.107128	0.541268	0.500013	0.499988	0.500002	0.500049
min	0.000000	11.400000	5.100000	0.000000	0.000000	0.000000	0.000000
25%	1.000000	12.200000	5.500000	0.000000	0.000000	0.000000	0.000000
50%	1.000000	13.100000	5.900000	0.000000	1.000000	0.000000	0.000000
75%	1.000000	14.000000	6.400000	1.000000	1.000000	1.000000	1.000000
max	1.000000	15.500000	7.100000	1.000000	1.000000	1.000000	1.000000

In [38]:
```
df.corr()
```

Out[38]:

	long_hair	forehead_width_cm	forehead_height_cm	nose_wide	nose_long	lips_thin	distance_nose_to_lip_long
long_hair	1.000000	-0.006530	-0.017233	0.001216	0.014432	0.011287	-0.025794
forehead_width_cm	-0.006530	1.000000	0.088596	0.251648	0.257368	0.258564	0.251328
forehead_height_cm	-0.017233	0.088596	1.000000	0.211655	0.194120	0.205441	0.215292
nose_wide	0.001216	0.251648	0.211655	1.000000	0.565192	0.557615	0.569303
nose_long	0.014432	0.257368	0.194120	0.565192	1.000000	0.561229	0.559794
lips_thin	0.011287	0.258564	0.205441	0.557615	0.561229	1.000000	0.565312
distance_nose_to_lip_long	-0.025794	0.251328	0.215292	0.569303	0.559794	0.565312	1.000000

In [39]:
```
df.gender = [1 if i == "male" else 0 for i in df.gender]
```

In [40]:
```
df.gender
```

Out[40]:
```
0       0
1       0
2       0
3       0
4       0
       ..
4996    0
4997    0
4998    0
4999    0
5000    0
Name: gender, Length: 5001, dtype: int64
```

In [41]:
```
df.head()
```

Out[41]:

	long_hair	forehead_width_cm	forehead_height_cm	nose_wide	nose_long	lips_thin	distance_nose_to_lip_long	gender
0	1	11.8	6.1	1	0	1	1	0
1	0	14.0	5.4	0	0	1	0	0
2	0	11.8	6.3	1	1	1	1	0
3	0	14.4	6.1	0	1	1	1	0
4	1	13.5	5.9	0	0	0	0	0

In [42]:
```
#n x_data
x_df = df.drop(["gender"],axis = 1)

# y_data
y_df = df.gender.values
```

In [43]:
```
x_df
```

Out[43]:

	long_hair	forehead_width_cm	forehead_height_cm	nose_wide	nose_long	lips_thin	distance_nose_to_lip_long
0	1	11.8	6.1	1	0	1	1
1	0	14.0	5.4	0	0	1	0
2	0	11.8	6.3	1	1	1	1
3	0	14.4	6.1	0	1	1	1
4	1	13.5	5.9	0	0	0	0
...	...	...	...	...	...	...	...
4996	1	13.6	5.1	0	0	0	0
4997	1	11.9	5.4	0	0	0	0
4998	1	12.9	5.7	0	0	0	0
4999	1	13.2	6.2	0	0	0	0
5000	1	15.4	5.4	1	1	1	1

5001 rows × 7 columns

In [44]:
```
y_df
```

Out[44]:
```
array([0, 0, 0, ..., 0, 0, 0], dtype=int64)
```

In [45]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x_df, y_df, test_size = 0.3, random_state = 1)
```

In [46]:
```
from sklearn.naive_bayes import GaussianNB

gnb = GaussianNB()

gnb.fit(x_train, y_train)

print("print Train for accuracy of NBC algo:", gnb.score(x_train, y_train))
print("print Test for accuracy of NBC algo:", gnb.score(x_test, y_test))
```

```
print Train for accuracy of NBC algo: 1.0
print Test for accuracy of NBC algo: 1.0
```

In [47]:

```
from sklearn.naive_bayes import MultinomialNB
mnb = MultinomialNB()

mnb.fit(x_train, y_train)
print("print Train for accuracy of MNB algo:",mnb.score(x_train, y_train))
print("print Test for accuracy of MNB algo:",mnb.score(x_test, y_test))
```

```
print Train for accuracy of MNB algo: 1.0
print Test for accuracy of MNB algo: 1.0
```

In [48]:

```
from sklearn.naive_bayes import BernoulliNB
bnb = BernoulliNB()

bnb.fit(x_train, y_train)

print("print Train accuracy for BNB algo:", bnb.score(x_train, y_train))
print("print Test accuracy for BNB algo:", bnb.score(x_test, y_test))
```

```
print Train accuracy for BNB algo: 1.0
print Test accuracy for BNB algo: 1.0
```

In [49]:

```
long_hair = int(input("Enter person's long hair:"))
forehead_width_cm = float(input("Enter person's forehead_width_cm:"))
forehead_height_cm = float(input("enter person's forehead_height_cm:"))
nose_wide = int(input("enter person's nose_wide:"))
nose_long = int(input("enter person's nose_long:"))
lips_thin = int(input("enter person's lips_thin:"))
distance_nose_to_lip_long = int(input("enter person's distance_nose_to_lip_long:"))
person = [[long_hair, forehead_width_cm, forehead_height_cm, nose_wide, nose_long, lips_thin, distance_nose_to_lip_long
result = gnb.predict(person)
print(result)

if result == 1:
    print("person may be Male")
else:
    print("person may not be Male")
```

```
Enter person's long hair:1
Enter person's forehead_width_cm:12
enter person's forehead_height_cm:1
enter person's nose_wide:2
enter person's nose_long:1
enter person's lips_thin:4
enter person's distance_nose_to_lip_long:1
[0]
person may not be Male

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but
GaussianNB was fitted with feature names
  warnings.warn(
```

In [50]:

```
long_hair = int(input("Enter person's long hair:"))
forehead_width_cm = float(input("Enter person's forehead_width_cm:"))
forehead_height_cm = float(input("enter person's forehead_height_cm:"))
nose_wide = int(input("enter person's nose_wide:"))
nose_long = int(input("enter person's nose_long:"))
lips_thin = int(input("enter person's lips_thin:"))
distance_nose_to_lip_long = int(input("enter person's distance_nose_to_lip_long:"))
person = [[long_hair, forehead_width_cm, forehead_height_cm, nose_wide, nose_long, lips_thin, distance_nose_to_lip_long
result = mnb.predict(person)
print(result)

if result == 1:
    print("person may be Male")
else:
    print("person may not be Male")
```

```
Enter person's long hair:1
Enter person's forehead_width_cm:2
enter person's forehead_height_cm:1
enter person's nose_wide:12
enter person's nose_long:10
enter person's lips_thin:4
enter person's distance_nose_to_lip_long:2
[0]
person may not be Male

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but
MultinomialNB was fitted with feature names
  warnings.warn(
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: