

Gender Classification Model by Blessing Nwokolo.

In [11]:

```
# get the data ready
%matplotlib inline
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
gender = pd.read_csv("DATA/gender.csv")
gender.head()
```

Out[11]:

	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 [12]:

```
len(gender)
```

Out[12]:

5001

In [13]:

```
gender.duplicated().sum()
```

Out[13]:

1768

In [14]:

```
gender.drop_duplicates(inplace=True)
```

In [15]:

```
len(gender)
```

Out[15]:

3233

In [16]:

```
gender.isna().sum()
```

Out[16]:

```
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
```

In [17]:

```
#create x (feature matrix)
x = gender.drop("gender", axis=1)

#create y (labels)
y = gender["gender"]
```

In [18]:

```
y.head()
```

Out[18]:

```
0    Male
1    Female
2    Male
3    Male
4    Female
Name: gender, dtype: object
```

In [19]:

```
x.head()
```

Out[19]:

	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

In [20]:

```
#v3.Split data into train and test
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2)
```

In [21]:

```
#Importtion of my model
from sklearn.ensemble import RandomForestClassifier

#Setup random seed
np.random.seed(42)

#make the data
x = gender.drop("gender", axis=1)
y = gender["gender"]

#ssplit the data
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2)

#instantiation of my medel
gd = RandomForestClassifier()
gd.fit(x_train, y_train)

#evaluate the model

gd.score(x_test, y_test)
```

Out[21]:

0.9536321483771252

In [33]:

```
np.array(y_test);
```

In [34]:

```
gd.predict(x_test);
```

In [24]:

```
# comparing predictions to truth labels to evaluate the model
y_preds = gd.predict(x_test)
np.mean(y_preds == y_test)
```

Out[24]:

0.9536321483771252

In [25]:

```
from sklearn.metrics import accuracy_score
accuracy_score( y_test, y_preds)
```

Out[25]:

0.9536321483771252

In []: