Human Gender Classification Using Machine Learning

Taghreed Alamri

14-10-2021

—

Data Science

—

Dr.Essam Al Dawood

1.Objective
The goal of this project is to classify whether a person is male or female considering the facial features (such as nose width, Forehead length, etc.) of that person, by applying different machine learning models and comparing their performance.

2.Exploring the Dataset
The data I will use is the Gender Classification dataset from Kaggle. The dataset is created considering real scenarios. It has 5001 samples that consists of 8 columns (7 features/predictors and 1 label/target column).

- **long_hair**: indicates whether this person has a long hair (1) or not (0).
- **forehead_width_cm**: width of the forehead from right to left given in cm.
- **forehead_height_cm**: height of the forehead in cm from where the hair grows to the eyebrows.
- **nose_wide**: whether the nose is wide or not. 1 represents wide and 0 not.
- **nose_long**: whether the nose is long or not. 1 represents long and 0 not.
- **lips_thin**: whether this person has a thin lip or not. 1 represents thin and 0 not.
  distance_nose_to_lip_long: is the distance from nose to lip is long? 1 represents yes and 0 not.
- **Gender**: either Meal or Female, it is the target column with 2 classes Male and Female.

2.1Data Types
long_hair, nose_wide, nose_long, lips_thin and distance_nose_to_lip_long columns are all of type integer.
forehead_width_cm and forehead_heigh_cm are of type float.
gender is of type object(string).

2.2 Detecting Missing Values

```
df.info()
```

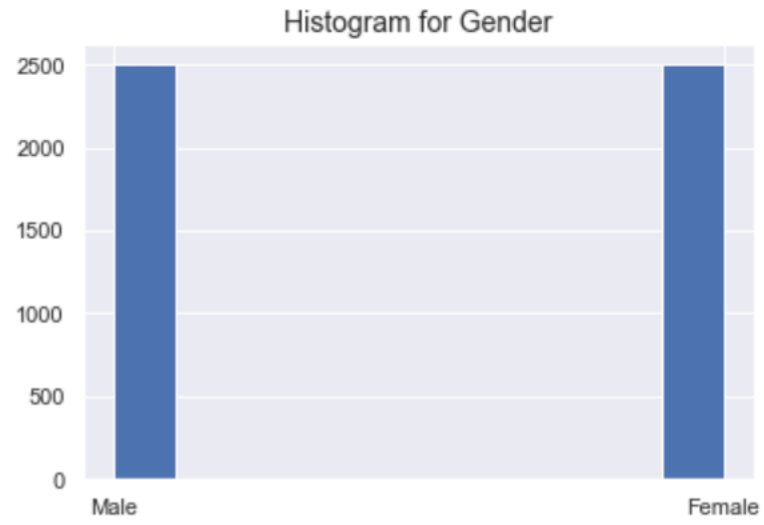
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5001 entries, 0 to 5000
Data columns (total 8 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   long_hair                  5001 non-null   int64
 1   forehead_width_cm          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
```

We can see that there are no missing values for all the columns (if we had missing values we can handle it by dropping the rows with missing values using dropna() or replacing it with value(mean or median) using fillna())

2.3 Dataset Balance

```
plt.figure()
plt.hist(df.gender);
plt.title("Histogram for Gender", fontsize =14)
df.gender.value_counts()
```

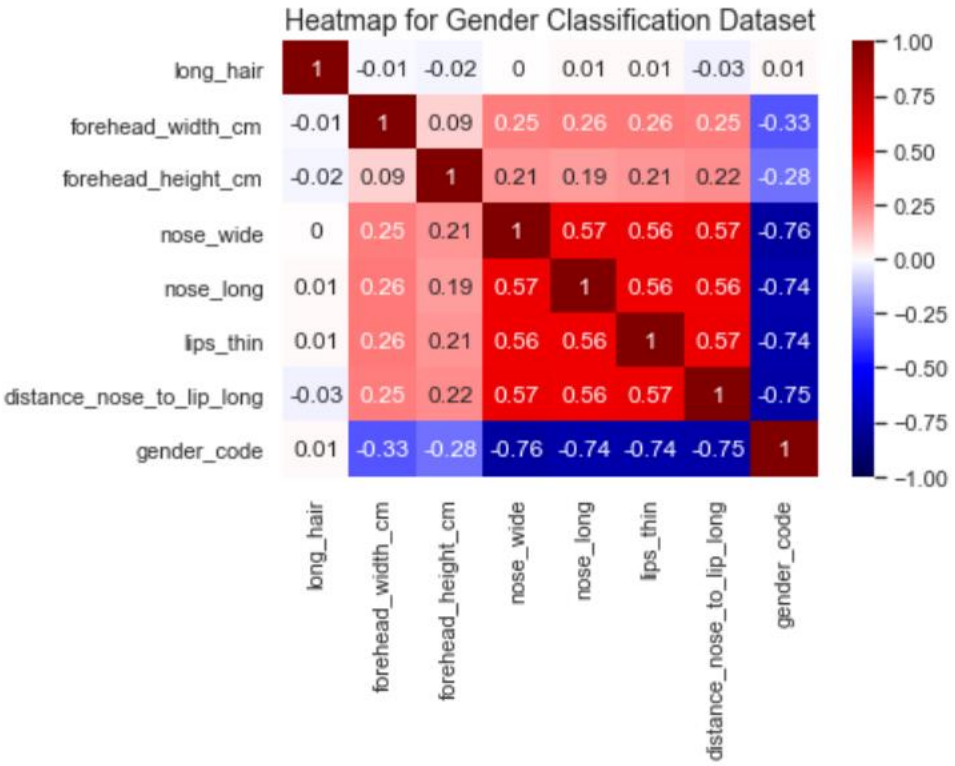
```
Female    2501
Male      2500
Name: gender, dtype: int64
```



As we can see, the dataset consists of 2501 Male samples and 2500 Female samples, so the dataset is balanced.

"gender" is categorical feature, so we change it to numerical with male= 0 and Female= 1 and we save it in a new column called "gender_code".

2.4 Correlation



We can see that there is a strong negative correlation between gender and nose width, nose length, lips thin and distance from nose to lips. Also, there is a low positive correlation between the features of the nose and lips.

2.5 Detecting Outliers

As we can see from the boxplot there is no outliers (if we had outliers we can handle it by dropping the rows with outliers using dropna() or replacing it with value(mean or median) using fillna()).