Human Gender Classification Using Machine Learning

Taghreed Alamri

14-10-2021

—

Data Science

—

Dr.Essam Al Dawood

Abstract

Gender classification is to determine a person's gender, e.g., male or female, based on his or her biometric cues. Face is one of the most important biometric traits and face-based approaches are the most popular for gender discrimination.
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, I used dataset from Kaggle that was created considering real scenarios and applied three classification models, k-nearest neighbors (KNN), Logistic Regression and Support Vector Machine (SVM) and compared their performance and concluded that KNN yield best result.

Design

This project originates from a "Predict Gender" task on Kaggle for data created considering real scenarios. Gender classification is essentially a two-class classification problem (result is often a binary value, e.g., 0 or 1, representing either male or female). Classifying gender accurately can boost the performance of many other applications including face recognition and smart human-computer interface.

Data

The data I used is the Gender Classification dataset from Kaggle. 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.

- 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).

the dataset contains 2501 Male samples and 2500 Female samples, so the dataset is balanced.

Algorithms

Feature Engineering

- I converted gender from categorical to numerical with male= 0 and Female= 1 and saved the result in a new column called "gender_code".
- I applied StandardScaler() to scale the features and found it gave better accuracy for KNN and Logistic Regression (For SVM I didn't apply it because I found that it decreases the accuracy).
- I split the dataset records into 60/40 train vs. test(holdout). I tried different ratios and found that this ratio gave best accuracy.

KNN Model

To get optimal performance, I used GridCV to Tune Hyperparameters of my KNN model with c v=5 and found that the best parameters are {'n_neighbors': 94, 'weights': 'uniform'}
The paramgrid I used is:

'n_neighbors'	['rbf', 'sigmoid', 'linear']
weights	['uniform', 'distance']

SVM Model

To get optimal performance, I used GridCV to Tune Hyperparameters of my SVM model with cv=5 and found that the best parameters are {'kernel': 'rbf', 'C': 10, 'gamma': 0.1}
The paramgrid I used is:

Kernel	['rbf', 'sigmoid', 'linear']
C	[0.1, 1, 10, 100, 1000]
gamma	[1, 0.1, 0.01, 0.001, 0.0001]

Logistic Regression *Model*

To get optimal performance, I used GridCV to Tune Hyperparameters of my LR model with cv =5 and found that the best parameters are {'C': 10, 'penalty': 'l1'}
The paramgrid I used is:

C	[0.1, 1, 10, 100]
penalty	['l1', 'l2']
solver	'liblinear'

Prediction for test data/ holdout

Metric	*KNN*	*SVM*	*LR*
Accuracy	*97.15%*	*96.95%*	*96.45%*
F1	*97%*	*97%*	*96%*
Precision	*97%*	*97%*	*96%*
Recall	*97%*	*97%*	*96%*

I found that KNN gave best performance with 97.15% accuracy. Then, SVM with 96.95% and the worst one is LR with 96.45% accuracy.

Tools

The main tools I used in addition to Python and Jupyter Notebook are:
• Numpy and Pandas for data manipulation.
• Scikit-learn for modeling.
• Matplotlib and Seaborn for plotting and visualization.

Communication

I uploaded the dataset and all project documents (proposal, MVP and writeup) and code on my GitHub public repo named "Gender Classification Project".