Part 1. Dataset Assessment

The dataset consists of 14640 observations and has 15 features including the target variable, airline\_sentiment. Of those features, airline\_sentiment\_gold, negativereason\_gold, and tweet\_coord have above 90% of null values, so it is better to neglect those columns when building a prediction model. The tweet\_location and user\_timezone features have more than 30% of null values. Even with the data that is not null, tweet\_location has a lot of invalid data like ‘this place called NYC’, or ‘Somewhere celebrating life,’ which cannot be used for prediction models. Similarly, user\_timezone has too much variety but very skewed data with most data being in the US & Canada timezone. Those features are also neglected when choosing the predictor variables. Also, nagativereason and negativereason\_confidence are also neglected to keep the dataset large enough because they have around 30% of null values.

Table

Description automatically generated

When using tweet\_created variable, it required some feature engineering because it is a object type variable and the predictive models cannot function with object or string type variables. I divided the tweet\_created column into date and time columns and removed anything other than the numeric variables. Date column now have a format of [YYYYMMDD] and time column now have a format of [HHMMSS]. The airline column is made into dummy variables and have value of 1 if they are in the category of the airline.

Part 2. Action List of Steps

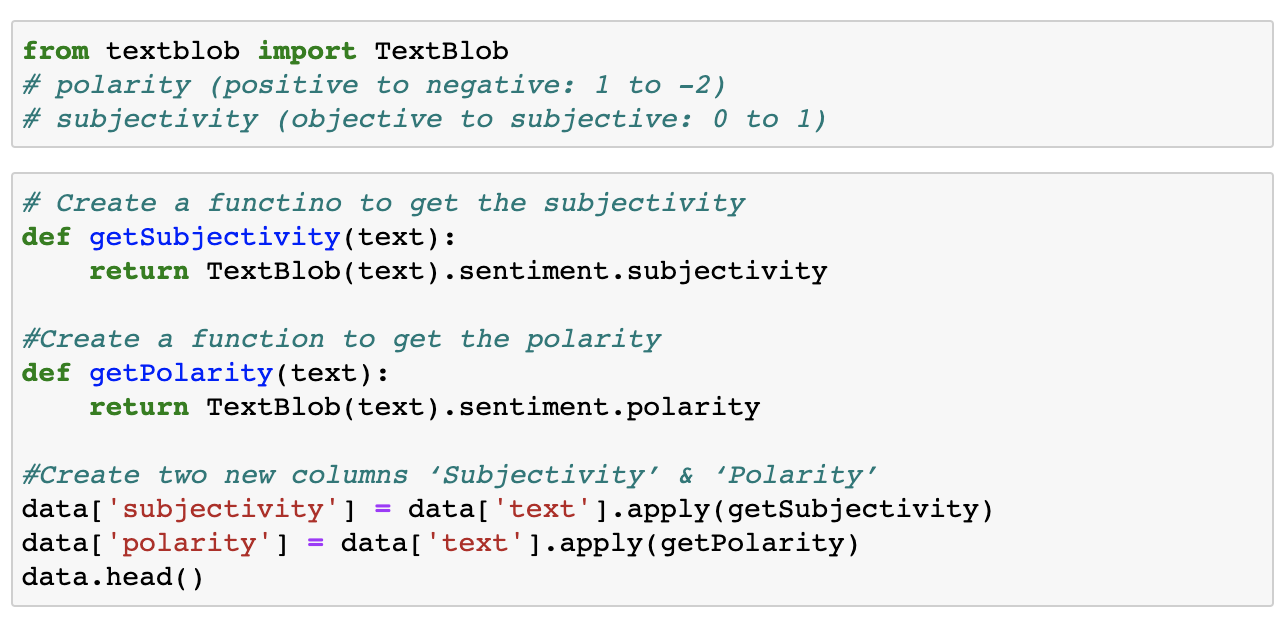
Data Preparation. Using TextBlob package, subjectivity and polarity scores are evaluated through sentiment analysis process. Polarity shows how negative or positive a text is, and it is represented by continuous numerical value between 1 and -1, 1 being positive and -1 being negative. Subjectivity shows how objective or subjective a text is, and it is represented by continuous numerical value between 0 and 1, 0 being objective and 1 being subjective. Then I split the dataset into 66% train set and 33% test set.

Building Model. I built logistic regression model, decision tree, and random forest without any adjustment and fit models into train set. Then I evaluated accuracy score. To use grid search, I put various range of parameters into GridSearchCV for each model and fit the model into train set. When I got which parameter scores the best, I created another model for each and evaluated the score.

Evaluating Model. To evaluate model performance, I used the score() function, confusion matrix method, and classification\_report() function to see the precision score, recall score, and f1-score.

Code:

TextBlob:



Models with no adjustments:

Graphical user interface, text, application

Description automatically generated

GridSearch and HyperParameter:

Graphical user interface, text, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Feature Importance:

Graphical user interface, text

Description automatically generated