Natural Language Processing (NLP)
Mini Project

Help Zomato Predict Rating from the Review

Objective: Using NLP and machine learning, make a model to predict the rating in a review based on the content of the text review. This will help identify cases with a mismatch.

Problem Statement:
Zomato is India's largest platform for discovering restaurants and ordering food. It operates in India as well as a few cities internationally. Bangalore is one of the biggest customers and restaurant bases for Zomato with 4 to 5 million users using the platform each month.

Users on the platform can also post reviews of restaurants and provide a rating accompanying the review. The content in the reviews should ideally reflect the rating provided by the customer. In many cases, there is a mismatch, owing to multiple reasons, where the rating does not match the customer review. The reviews and rating match is very important as it builds customer trust on the platform and helps the user get an accurate picture of the restaurant.

You, as a data scientist, need to enable the identification and cleanup of such cases to ensure the ratings reflect the reviews and that the reviews seem trustworthy to the customer. You will need to use NLP techniques in conjunction with machine learning models to predict the rating from the review text.

Domain: Hospitality and internet

Analysis to be done: Perform specific data cleanup, build a rating prediction model using the Random Forest technique and NLP.

Content:
rating: the rating given by the customer
review_text: the text in the review

Steps to perform:

Perform clean up on the data; tweak the stop words (negative terms are important). Follow up with a Random Forest Regressor to predict the star rating given by the customers.

Tasks:

1. Load the data using read_csv function from pandas package
2. Null values in the review text?
    a. Remove the records where the review text is null
3. Perform cleanup on the data
    a. Normalize the casing

b. Remove extra line breaks from the text
c. Remove stop words
   i. Note: Terms like ‘no’, ‘not’, ‘don’, ‘won’ are important, don’t remove
d. Remove punctuation

4. Separation into train and test sets
   a. Use train-test method to divide your data into 2 sets: train and test
   b. Use a 70-30 split
5. Use TF-IDF values for the terms as features to get into a vector space model
   a. Import TF-IDF vectorizer from sklearn
   b. Instantiate with a maximum of 5000 terms in your vocabulary
   c. Fit and apply on the train set
   d. Apply on the test set
6. Model building: Random Forest Regressor
   a. Instantiate RandomForestRegressor from sklearn (set random seed)
   b. Fit on the train data
   c. Make predictions for the train set
7. Model evaluation
   a. Report the root mean square error
8. Hyperparameter tuning
   a. Import GridSearch
   b. Provide the parameter grid to choose:
      i. max_features – ‘auto’, ‘sqrt’, ‘log2’
      ii. max_depth – 10, 15, 20, 25
9. Find the parameters with the best mean square error in cross-validation
   a. Choose the appropriate scoring as the metric for scoring
   b. Choose stratified 5 fold cross-validation scheme
   c. Fit on the train set
10. What are the best parameters?
11. Predict and evaluate using the best estimator
    a. Use the best estimator from the grid search to make predictions on the test set
    b. What is the root mean squared error on the test set?
12. Can you identify mismatch cases?
    a. Make a rule based on the predicted value and the actual value that identifies mismatch cases (e.g. difference in actual and predicted being more than a cutoff)
    b. How many such cases do you see?
    c. Are all these mismatch cases genuine?