Task_6_Report

Overview

In this task, I explored both text representation and classification methods based on restaurant review and health inspection results dataset. This task has 2 parts. In the first part, I uses unigram and bigram, two text representation methods, to classification and get the most frequent words from dataset. Classification algorithms application is the second part. I apply 7 different classification algorithms and compare their performance using F1-score. A visualization is created for comparison

Toolkit

Python 3.9 and Jupyter are the environments I adopted in this Task. To preprocess text data, I adopt nltk and string packages. In Text representation part, Count Vectorizer is adopted to make unigram and bigram tasks. In classification part, I use algorithms from sklearn and xgboost packages. Besides, f1_score in sklearn.metrics is used for comparison. I uses train_test_split in sklearn.model_selection model to help create classification models.

Data Preprocess

I use nltk package in this task to preprocess data. To detective punctuations, String package is used. After extracting all review data from dataset. Tokenize function was used for tokenizing. Then, I use stopwords in nltk.corous for removing all common stopwords from data. Finally, to uniform text and removing affixes, I use porterstemmer from nltk.stem.

Features from Text

I extract features from cuisine types and zip code in additional files.

For cuisine types, I mapped each cuisine to a column. 1 in each column means this restaurant serves this type, 0 for otherwise. Most of the restaurants serves more than 1 cuisines, and one restaurant can have several cuisine types with 1 in my data.

Zip code is an interesting data. Despite being made up of numbers, zip codes cannot be compared with each other solely based on their number. Like cuisines, I mapped each zip code to one unique column.

Text Representation

In this part of tasks, I use text I get after preprocessing in previous step and then compute the count of each unigram and bigram terms. Based on the counts, I extract the top terms in terms of their mutual information value.

The bigram results:

```
'american new': 0,
'tex mex': 28,
'breakfast brunch': 3,
'asian fusion': 2,
'tapas bars': 26,
'dim sum': 7,
'american traditional': 1,
'sushi bars': 25,
'modern european': 19,
'fast food': 8,
'gluten free': 12,
'soul food': 24,
'himalayan nepalese': 13,
'food stands': 11,
```

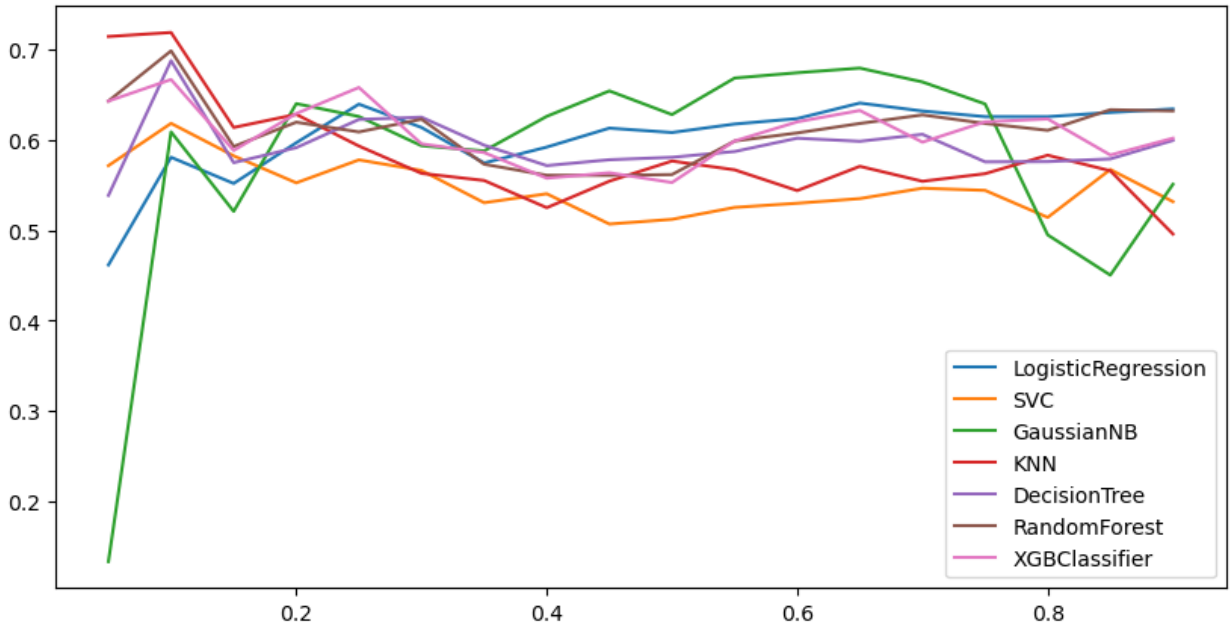
```
 'latin american': 16,
 'tapas small': 27,
 'small plates': 23,
 'hot dogs': 14,
 'live raw': 17,
 'raw food': 22,
 'hot pot': 15,
 'middle eastern': 18,
 'cajun creole': 4,
 'chicken wings': 5,
 'fish chips': 9,
 'food court': 10,
 'persian iranian': 20,
 'comfort food': 6,
 'puerto rican': 21
```

The unigram results:

```
'reviews': 90,
 'cuisine': 31,
 'location': 69,
 'review_num': 89,
 'avg_rating': 5,
 'vietnamese': 121,
 'sandwiches': 95,
 'restaurants': 88,
 'american': 2,
 'new': 79,
 'mexican': 73,
 'tex': 113,
 'mex': 72,
 'breakfast': 11,
 'brunch': 13,
 'french': 46,
 'creperies': 29,
 'vegetarian': 119,
 'ethiopian': 38,
 'vegan': 118,
 'thai': 114,
 'barbeque': 6,
 'chinese': 23,
 'japanese': 62,
 'delis': 32,
 'seafood': 98,
 'buffets': 14,
 'burgers': 15,
 'southern': 104,
 'asian': 3,
 'fusion': 47,
 'tapas': 112,
 'bars': 7,
 'irish': 60,
 'caribbean': 20,
 'dim': 33,
 'sum': 108,
 'cantonese': 19,
 'italian': 61,
 'cafes': 16,
 'traditional': 115,
 'steakhouses': 107,
 'sushi': 109,
 'modern': 75,
 'european': 39,
 'fast': 40,
 'food': 44,
 'greek': 51,
 'pizza': 82,
 'cheesesteaks': 21,
 'szechuan': 110,
 'mediterranean': 7
1,
 'gluten': 50,
 'free': 45,
 'pakistani': 80,
 'indian': 57,
 'hawaiian': 54,
 'soul': 102,
 'korean': 63,
 'moroccan': 77,
 'himalayan': 55,
 'nepalese': 78,
 'stands': 106,
 'salvadoran': 94,
 'latin': 66,
 'small': 101,
 'plates': 83,
 'hot': 56,
 'dogs': 35,
 'live': 68,
 'raw': 87,
 'pot': 85,
 'middle': 74,
 'eastern': 36,
 'african': 1,
 'taiwanese': 111,
 'halal': 53,
 'soup': 103,
 'diners': 34,
 'turkish': 117,
 'spanish': 105,
 'haitian': 52,
 'cuban': 30,
 'basque': 8,
 'russian': 92,
 'salad': 93,
 'cajun': 17,
 'creole': 28,
 'chicken': 22,
 'wings': 122,
 'scandinavian': 96,
 'fish': 42,
 'chips': 24,
 'german': 49,
 'brazilian': 10,
 'malaysian': 70,
 'trinidadian': 116,
 'gastropubs': 48,
 'indonesian': 58,
 'shanghainese': 10
0,
 'laotian': 65,
 'lebanese': 67,
 'afghan': 0,
 'belgian': 9,
 'court': 27,
 'cambodian': 18,
 'mongolian': 76,
 'australian': 4,
 'filipino': 41,
 'british': 12,
 'kosher': 64,
 'persian': 81,
 'iranian': 59,
 'venezuelan': 120,
 'colombian': 25,
 'fondue': 43,
 'polish': 84,
 'senegalese': 99,
 'comfort': 26,
 'puerto': 86,
 'rican': 91,
 'egyptian': 37,
 'scottish': 97
```

Classification Algorithm

In this part, I adopt 7 different classification algorithms. All algorithms I use, including LogisticRegression, SVC, GaussianNB, KNN, DecisionTree, RandomForest, and XGB. I use all additional features including cuisine types, location, number of reviews, and the average rating to generate classification model. Cuisine types and location are preprocessed as mentioned in Data Preprocess part. After getting f1-scores of each algorithm, I visualize them in a plot. X-axis shows the test size set when train the model. Y-axis shows the scores of each algorithm.



As can be seen, GaussianNB has a better performance in this task when we set test size between 0.4 and 0.6. Besides, KNN have the best performance in all selected algorithms when test size is set as 0.1.

Conclusion

In this task, I trained text representation method and classification methods using python. In text representation part, preprocessing the review text and generating phrases and n-grams would help increase the accuracy of classification and topic extraction. In classification part, different algorithms may have different performance. KNN performs better than other algorithms in this task.