CSC 869

Data Mining

Term Project

Extracting the Ratings of user reviews based on text review data

(Ratings From Reviews)

Project Report

Submitted By:

AnshulVyas

Spring 2015

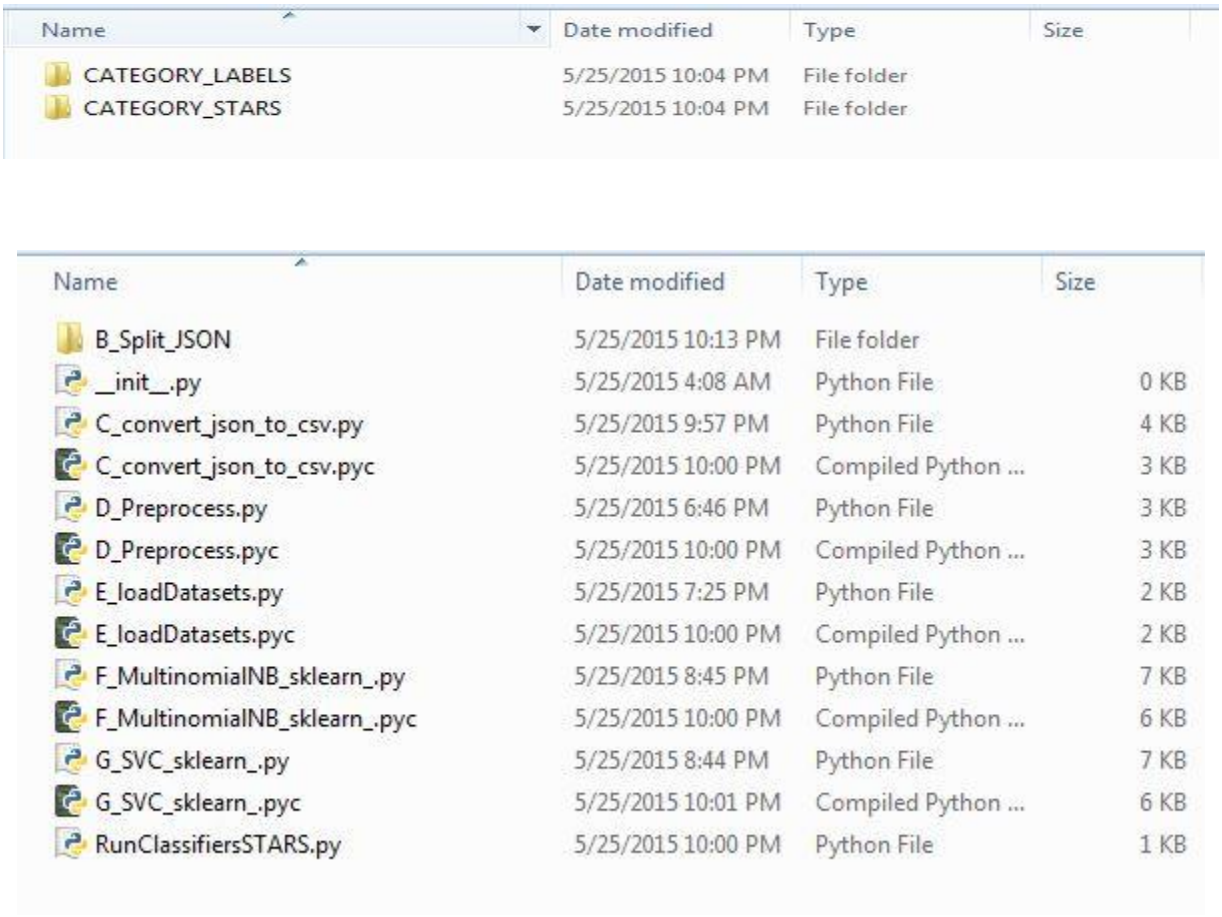
SAN FRANCISCO STATE UNIVERSITY

SAN FRANCISCO, CA

Contents

PART – I Instructions on compiling and running the program(s):

The folder structure for my project is shown in the screenshots. Make sure the folder structure is the same as the output and input files generated will be as according to this structure. The highlighted words are the folders where the script will be found and executed.



Name	Date modified	Type	Size
CATEGORY_LABELS	5/25/2015 10:04 PM	File folder	
CATEGORY_STARS	5/25/2015 10:04 PM	File folder	

Name	Date modified	Type	Size
B_Split_JSON	5/25/2015 10:13 PM	File folder	
__init__.py	5/25/2015 4:08 AM	Python File	0 KB
C_convert_json_to_csv.py	5/25/2015 9:57 PM	Python File	4 KB
C_convert_json_to_csv.pyc	5/25/2015 10:00 PM	Compiled Python ...	3 KB
D_Preprocess.py	5/25/2015 6:46 PM	Python File	3 KB
D_Preprocess.pyc	5/25/2015 10:00 PM	Compiled Python ...	3 KB
E_loadDatasets.py	5/25/2015 7:25 PM	Python File	2 KB
E_loadDatasets.pyc	5/25/2015 10:00 PM	Compiled Python ...	2 KB
F_MultinomialNB_sklearn_.py	5/25/2015 8:45 PM	Python File	7 KB
F_MultinomialNB_sklearn_.pyc	5/25/2015 10:00 PM	Compiled Python ...	6 KB
G_SVC_sklearn_.py	5/25/2015 8:44 PM	Python File	7 KB
G_SVC_sklearn_.pyc	5/25/2015 10:01 PM	Compiled Python ...	6 KB
RunClassifiersSTARS.py	5/25/2015 10:00 PM	Python File	1 KB

Figure 1 Structure of Project Folder

Initially my code was scattered everywhere and it was difficult to manage. After giving the demo, I realized that I need to consolidate my evaluations of each classifier and feature sets.

If you will follow these steps, you will be able to run the scripts:

1. "A_OnlyRestaurants.py":
   This step is an individual script to extract the restaurants reviews from all other reviews in the dataset. . The folder "A_OnlyRestaurants" can be found on Google Drive, which contains the required files needed to run the script. You can access the folder here: https://drive.google.com/folderview?id=0B6rp7NwWox3YfnM4QkdENGpFVkZkZlY5alFCWkNOSkhJUzRreFBueWZ5b0hSa1FiQzlYSWc&usp=sharing
   Input Files: 'yelp_academic_dataset_business.json' &yelp_academic_dataset_review.json
   Output Files: 'restaurants.json' & 'restaurants_review.json
   USAGE: python A_Restaurants.py

2. "MainAnalysis/B_SplitJSON":
   This step is completed beforehand as it only requires the Linux *"split"* command to split the retaurants_review.json into multiple small files. As we will be working on these files only, I didn't include the rest of the split files because I did not perform analysis on them. The file "70000_restaurant_ab.json" is also available on Google Drive, here:
   Files: *"10000_restaurant_ab.json" & "70000_restaurant_ab.json"*

3. "MainAnalysis/CATEGORY_STARS/RunClassifiersSTARS.py":
   After the second step, we can run and compare both of our classifiers with their respective feature sets by executing this script in the command line. This script is linked to all other scripts present in the folder "MainAnalysis".
   USAGE: python RunClassifiersSTARS.py

4. "MainAnalysis/CATEGORY_LABEL/RunClassifiersLABEL.py":
   Similarly, we can run and compare both of our classifiers with their respective feature sets by executing this class categorized script in the command line. This script is linked to all other scripts present in the folder "MainAnalysis".
   USAGE: python RunClassifiersLABELS.py

   The A_OnlyRestaurants folder is meant to run separately. If you want to run the script, it can be done individually.

   Rest of the files and the datasets are available in the .zip folder submitted with this report.

PART – IIDescription of Main Strategies:

Firstly, the Yelp! datasetI used for this project was in the form of nested JSON. The review dataset consists of 1.6M user reviews of all the businesses in major cities of US, Canada, UK, Germany, for example, Phoenix, Los Angeles, Pittsburgh, Edinburgh, and Karlsruhe.

The flow chart describes the main strategies implemented by me, and the detailed description is after that.

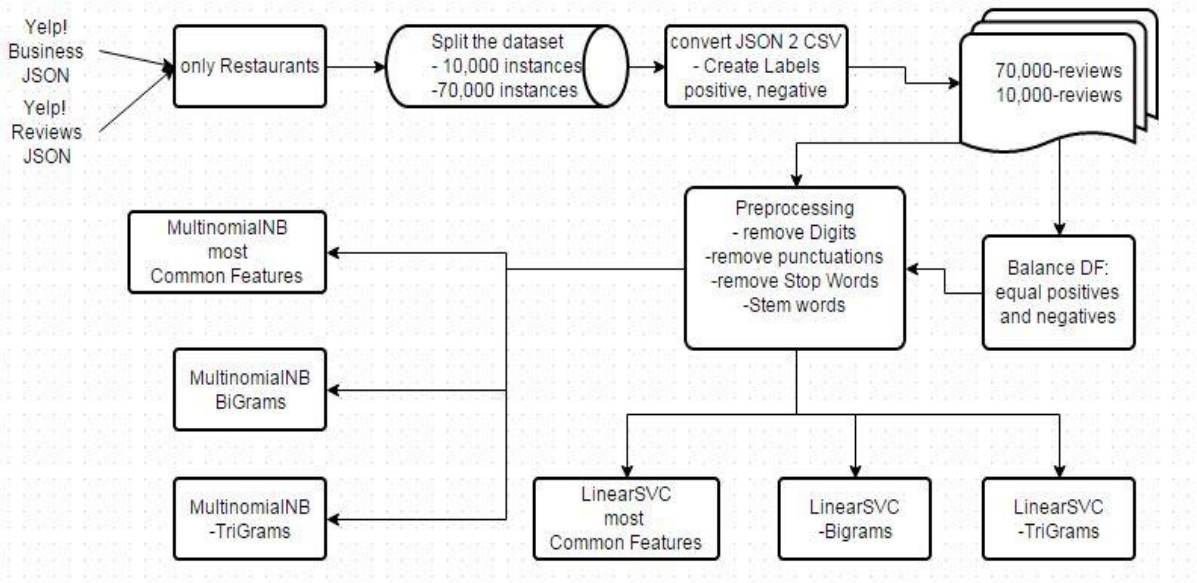


Figure 2Flow Chart for the main strategies implemented

For the analysis, I decided to work on reviews where business category was "Restaurants". This gave me an opportunity to consider all the restaurants there are in the dataset. Unaware of the size of the dataset, I made an inner join of the ['business_id'] between the Business Dataset and the review Dataset, and was able to extract the reviews that consist of only restaurants as their businesses. The size of file was around 800MB.

So, my next task was to split the reviews file into multiple smaller datasets which I can use for analysis. I used the Linux "split" command for this. I made two subsets of the big dataset. First one was of 10000 instances and another of 70000 instances. I discarded rest of them and included 1 dataset from both sizes.

```
NANSH@NANSH-PC /cygdrive/e/PyCharm_Projects/Demo/1onlyRestaurants
$ split -l 10000 yelp_academic_dataset_review.json 10000_restaurant
```

Now, I had 2 datasets, each of 10000 instances and 70000 instances. My next task was to convert the nested JSON datasets to CSV for which i wrote a Python script using the JSON library. I also intuitionally removed some attributes because my scope of initial analysis was for only to work on the Star ratings and the review text. I added another attribute as LABEL where I divided the star categories as:

“5”, “4”, “3” – positive
“2”, “1” – negative
The next thing was data visualization, so that I could get to know my data better. I made histograms for reviews that were in each star category.

Figure 3 Data visualization of two different size datasets

After seeing this, I knew I made an error while labeling the data. I relabeled the dataset as:

“5”, “4” – positive
“3” - neutral
“2”, “1” – negative
So that the dataset may now be balanced.Now it was time to do some preprocessing. I wrote a script where I defined several methods for cleaning the text reviews of all the digits and punctuations and mainly the Stop Words. I also used a Porter Stemming algorithm readily available in the NLTK to stem the words.

Next I implemented the Naïve Bayes Classifier which is readily available in NLTK dataset and ran it on different datasets. The results were similar to the sk-learn’sMultinomialNB I implemented next. Hence, it didn’t make sense to include the same analysis twice.

After that, I came across the text mining book online which was based on sklearn. Intrigued to try the new methodology, I shifted my classifier from NLTK to sk-learn’s Multinomial Naïve Bayes. I made 3 feature sets using the Count Vectorizer function

- Most common features in the dataset
- BiGrams
- TriGrams

I ran all my datasets on the MultinomialDB with feature set of 400 most common words in the 10000 instances dataset and then on other feature sets as well. The results of it I will discuss in the next section.

Similar procedures were followed using the Linear SVC model of sklearn. I ran these feature sets to see the difference between the two classifiers. The multinomial Naïve Bayes performed much better than SVC model implemented.

Initial analysis included both 10000 and 70000 instances. The results only included the accuracy of the classifiers on the training datasets. But as I explored more, I came to realize that the 70000 instances dataset had words from other languages as well. As the reviews were from Germany and Canada also, most of the reviews contained German or French words. This was a major setback for me. Due to time constraints, I decided to focus my analysis only on the 10000 instances dataset.

Although I have shown the results of 70000 instances, I have removed the dataset and the scripts associated with it.

I have done the exact same steps but keeping the class labels as – "positive", "neutral" "negative". As there were only 3 classes, the accuracy, precision and recall results were better than the 5 class labels.

The sample output of a file is:

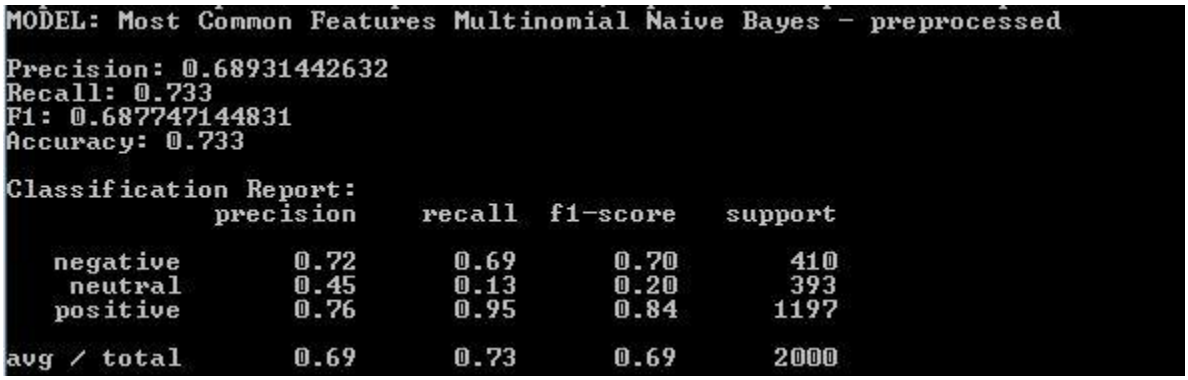


Figure 4 Sample output of Multinomial NB Classifier with most common features as feature set

Considering two datasets for the classification:

- 10000 – Preprocessed but not balanced
- 10000 – Preprocessed and balanced

I have tried to compare and analyze the results of the classifiers on these two datasets. The evaluations of these classifiers are briefly discussed in the next section.

PART – III Evaluation Results and discussions:

Here is a table for comparison between the two Classifiers – Multinomial Naïve Bayes and SVC.

FEATURE SETS ->			Most Common Features		Bi Grams		Tri Grams	
CATEGORY ->			Stars	Labels	Stars	Labels	Stars	Labels
CLASSIFIERS		Precision	0.495	0.68	0.457	0.68	0.379	0.585
	10000-preprocessed	Recall	0.5	0.733	0.45	0.702	0.389	0.576
		Accuracy	0.5	0.733	0.45	0.702	0.385	0.638
Multinomial Naïve Bayes		Precision	0.471	0.652	0.46	0.618	0.351	0.492
	10000-balanced	Recall	0.476	0.647	0.38	0.612	0.351	0.466
		Accuracy	0.476	0.644	0.42	0.609	0.351	0.46

Figure 5 Evaluation of Multinomial NB

FEATURE SETS ->			Most Common Features		Bi Grams		Tri Grams	
CATEGORY ->			Stars	Labels	Stars	Labels	Stars	Labels
CLASSIFIERS		Precision	0.475	0.691	0.424	0.623	0.34	0.513
	10000-preprocessed	Recall	0.471	0.676	0.427	0.625	0.36	0.6
Linear SVC		Accuracy	0.472	0.711	0.427	0.664	0.367	0.6
		Precision	0.438	0.611	0.4	0.561	0.339	0.48
	10000-balanced	Recall	0.43	0.622	0.37	0.564	0.334	0.452
		Accuracy	0.43	0.622	0.396	0.564	0.334	0.452

Figure 6 Evaluation of SVC

This table shows the comparison between two different datasets – preprocessed Dataset and balanced Dataset.

Analyzing the different classes, we can infer the following:

- The accuracy of label category is much better than the Star category, because there are 3 classes to classify as compared to 5.
- The performance of Multinomial Naïve Bayes is better than the Linear SVC.
- Balancing the dataset decreases the accuracy of the classifier.

The graphs of the two datasets for Naïve Bayes and SVC are as follows:

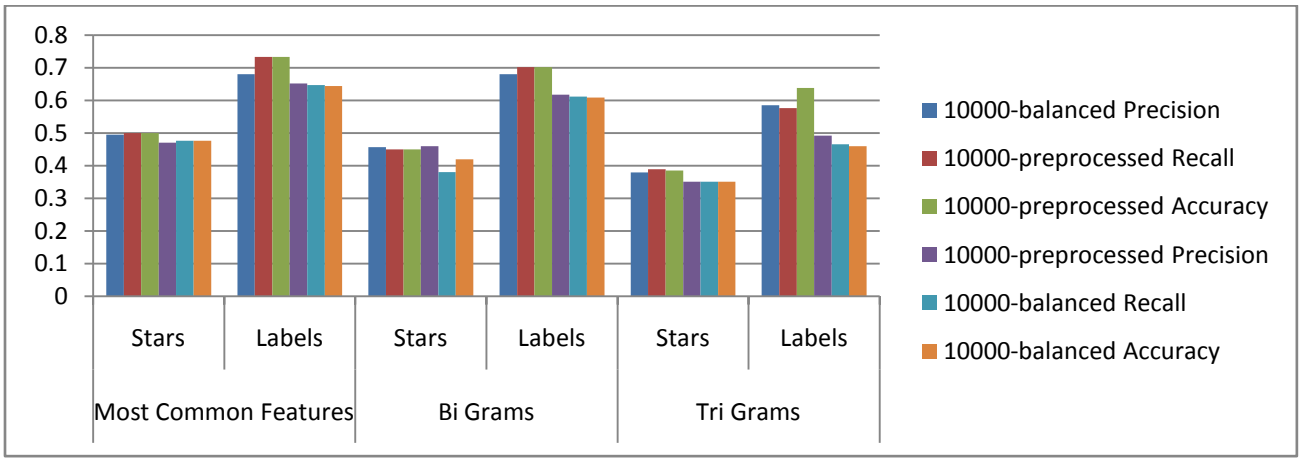


Figure 7 Multinomial Naive Bayesian (2 Datasets)

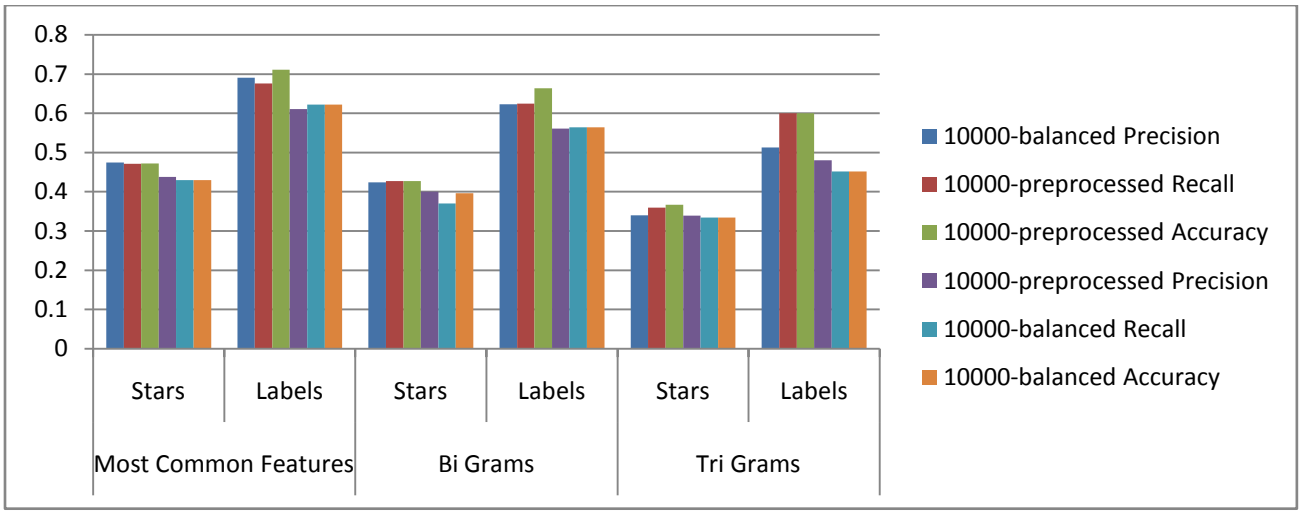


Figure 8 SVC (2 Datasets)

PART – IVConclusion and Future Work:

In conclusion, I would like to quote Ralph Waldo Emerson, who has said, "Life is a journey, not a destination." I couldn't find it more relevant than in Data Mining analysis. The analysis shown in the report is the result of various approaches which I have tried to implement.I have used Multinomial Naïve Bayes and SVC as the two classifiers for the analysis. The results showed that MultinmialNB has better performance than SVC. There are still many possibilities to explore and work on. Here are some of the things I would like to move towards:

Future work:

- Due to time constraints, I could not compare reviews between different cities, it would have been interesting to draw some conclusions based on reviews from different cities.
- Implement more classifiers such as Random Forest and also Adaboost. Or use ensemble approach.
- Implement the learning curve and confusion matrix.
- Correlating different datasets such as Business, Users, Tips with the Review dataset.