CSCI 5408 Data Management, Warehousing and Analytics

Project Report

Integrated Review System – *Analysis of Restaurant Reviews*

Submitted by:

Hemanth Kurra (B00784050)

Srisaichand Singamaneni (B00792835)

Contents

ABSTRACT

The statistics show that there are 115,394 number of food and service businesses across Canada and on an average, Canadians make 17 million restaurant visits on a typical day (Summary - Canadian Industry Statistics, 2017). One of the easy ways to choose a restaurant in a new place is reviews or ratings. There could be hundreds of reviews on each restaurant. Therefore, the developed Integrated Review System will do sentiment analysis on the restaurant reviews and provide a consolidated review of the restaurant. This Review system will save a lot of time for the customer and helps the business to understand the emotions of the customers easily.

PROBLEM STATEMENT

In general, most people rely on online reviews for choosing restaurant. But end up in reading all the reviews or just see the star ratings. There are many websites, applications and forums to help the customers to fetch information and know about the restaurant. But certain times customer might be in a position where, confused to choose the website or application for reviewing. So, the developed application could fill the gap between customers and review process.

INTRODUCTION

The Integrated Review System is a one stop solution to the people, who rely on the restaurant reviews for the choosing the restaurant. It is aimed to consolidate the reviews from different sources, analyse the reviews and provide the integrated review of the restaurant. The system uses the Yelp reviews dataset from Kaggle website to train the model. The captured reviews from different sources will be loaded into the trained model for predictive analysis and the consolidated review would be provided to the customers and business. Also, the reviews of restaurant are classified into three categories positive, negative, and neutral.

VALUE PROPOSITION

The developed Integrated Review System would help the customers and as well as the businesses. By getting the data from multiple sources and using it for predictive analysis, draws a line out from other review platforms available in the market. Based on the classified category review, the customers and businesses are benefited in terms of time on review analysis. This can be easily integrated into a graphical user interface in future, so that everyone can have easy access to the review system.

IMPLEMANTATION

Step1:

Requirement Analysis; Researched different data sources for selecting the dataset and required information to kick start the project. In this step, we choose Yelp review dataset for our model training and Google API's for streaming the latest reviews. Later, we discussed and researched on certain tools to implement the project. The tools will be discussed under tools and technologies section.

Step2:

Data Pre-processing; In this step we started looking into data cleaning. The extracted Yelp review dataset is a roughly three and half GB. The first challenge here was, we found it difficult to analyse the data manually due to its size. Therefore, we break down the dataset into number of small chucks.

By analysing the dataset, we found that there is lot of noisy data such as null values, new line characters etc. Finally, we removed the unnecessary columns and cleaned the data using python script.

Step3:

Sentiment Analysis; In this step we performed sentiment analysis on the pre-processed dataset by using Vader sentiment analysis library. The library was imported from GitHub repository, it is lexicon and rule-based sentiment analysis tool that was specially designed for analysing the emotions in social media and released under MIT open source license (Cjhutto, 2018). Finally, we derived the review text, sentiment of the text i.e., either positive, negative or neutral and the sentiment score.

Step4:

Model Training; In this step we started towards our model training. The outcome data of step3 was used for the model training. We used spark Machine Learning libraries for feature extraction and classification. Regex Tokenizer and Count vectors libraries are used for feature extraction and logistic regression algorithm was used for the classification.

```
2018-08-05 16:08:19 INFO  Executor:54 - Running task 0.0 in stage 50.0 (TID 446)
2018-08-05 16:08:19 INFO  Executor:54 - Finished task 0.0 in stage 50.0 (TID 446). 705 bytes result sent to driver
2018-08-05 16:08:19 INFO  TaskSetManager:54 - Finished task 0.0 in stage 50.0 (TID 446) in 3 ms on localhost (executor driver) (2/2)
2018-08-05 16:08:19 INFO  TaskSchedulerImpl:54 - Removed TaskSet 50.0, whose tasks have all completed, from pool
2018-08-05 16:08:19 INFO  DAGScheduler:54 - ResultStage 50 (aggregate at AreaUnderCurve.scala:45) finished in 0.029 s
2018-08-05 16:08:19 INFO  DAGScheduler:54 - Job 37 finished: aggregate at AreaUnderCurve.scala:45, took 0.045776 s
2018-08-05 16:08:19 INFO  MapPartitionsRDD:54 - Removing RDD 107 from persistence list
2018-08-05 16:08:19 INFO  BlockManager:54 - Removing RDD 107
Accuracy: 0.652818
2018-08-05 16:08:19 INFO  deprecation:1173 - mapred.output.dir is deprecated. Instead, use mapreduce.output.fileoutputformat.outputdir
2018-08-05 16:08:19 INFO  FileOutputCommitter:108 - File Output Committer Algorithm version is 1
2018-08-05 16:08:19 INFO  SparkContext:54 - Starting job: runJob at SparkHadoopWriter.scala:78
2018-08-05 16:08:19 INFO  DAGScheduler:54 - Got job 38 (runJob at SparkHadoopWriter.scala:78) with 1 output partitions
2018-08-05 16:08:19 INFO  DAGScheduler:54 - Final stage: ResultStage 51 (runJob at SparkHadoopWriter.scala:78)
2018-08-05 16:08:19 INFO  DAGScheduler:54 - Parents of final stage: List()
2018-08-05 16:08:19 INFO  DAGScheduler:54 - Missing parents: List()
2018-08-05 16:08:19 INFO  DAGScheduler:54 - Submitting ResultStage 51 (MapPartitionsRDD[116] at saveAsTextFile at ReadWrite.scala:283), which has no missing paren
2018-08-05 16:08:19 INFO  MemoryStore:54 - Block broadcast_84 stored as values in memory (estimated size 70.6 KB, free 408.0 MB)
2018-08-05 16:08:19 INFO  MemoryStore:54 - Block broadcast_84_piece0 stored as bytes in memory (estimated size 25.0 KB, free 408.0 MB)
2018-08-05 16:08:19 INFO  BlockManagerInfo:54 - Added broadcast_84_piece0 in memory on ip-172-31-5-30.ca-central-1.compute.internal:35150 (size: 25.0 KB, free: 41
```

Figure 1 : Model Accuracy

Step5:

Reviews streaming; In this step, we enrolled for Google Maps API. It is a cluster of different level of requests like Google Places Search API, Google Places Details API, Google Place Photos API and many more. To extract reviews from this API, we need to make two API request calls. Initially, we must make a request call for Google Place Search API, wherein the result will be a Nested JavaScript Object Notation (JSON). We passed through the JSON response to extract place ID of a restaurant and used it as an input to Google Place Details API. This result in another Nested JSON where we have multiple parameters like reviews, place description, and many more. Finally, we converted the output to Extensible Mark-up Language (XML) format for ease of parsing through the data and converted the output file into Comma Separated Values (CSV).

Step6:

Test Data Prediction; In this step the test data of reviews which was streamed from the step5 was loaded into the trained model using pipeline library form the Spark libraries. The predictions were derived and labelled as Sentiment Text, and Sentiment of the reviews as positive, negative or neutral.

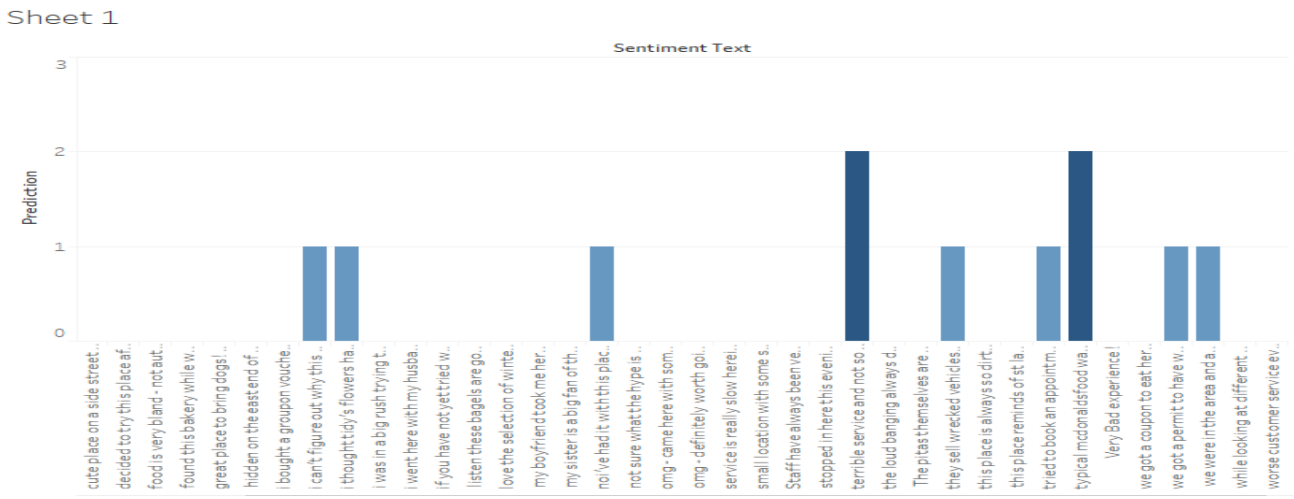

Figure 2: Prediction classification

Step7:

Consolidated Review; In this step, we calculated the average of the predictions and provided the final integrated review of the restaurant.

WORKFLOW

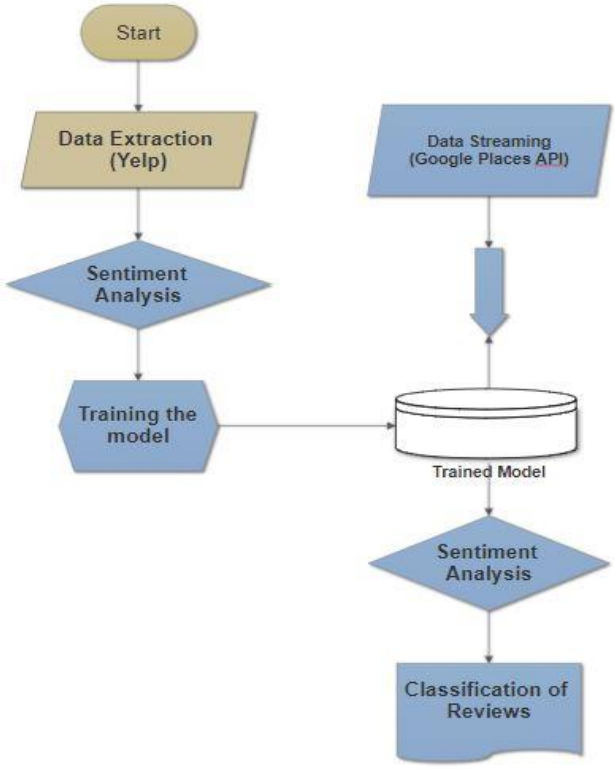

Figure 3:The work flow structure

DATA SOURCES

For this project, we used only one open dataset i.e., Yelp reviews, that is extracted from the Kaggle website. Also, Google reviews of individual restaurants are streamed using the google maps platform, google places and search API's.

TOOLS, ALGORITHM & PROGRAMMING LANGUAGES

To implement all these operations, we used "Python" as our programming language. Python is well-known for its libraries and ease of accessing API's. For this project, Google Places and Search API's are used with python by importing their respective packages. Python also has extensive library support like Pandas which is used for data manipulation and analysis.

To perform Data Analytics and Management on the Dataset we used "Apache Spark". Apache Spark provides quick and effective processing on data, either storing or streaming using the immutable Resilient Distributed Dataset (RDDs). Spark Machine Learning library is a machine learning framework used for classification, regression and clustering the dataset.

Python and Spark are installed on amazon EC2 instance. Alongside these tools, Vader Sentiment library was used for sentiment analysis and postman tool was used for API requests.

WORK BREAKDOWN

The entire work is break down into four sprints.

Sprint1:

Sprint-one consists of requirement analysis and data pre-processing. This was a challenging sprint, with gathering all the requirements and finalising the dataset. All decisions were made in this sprint. Stared with R-Studio for the data management and analysis. But unfortunately, we need to roll back in this step, because of the file size limitations. Later, in this sprint we completed all the data pre-processing work. However, the sprint was delayed than the estimated timeline.

Sprint2:

Sprint-two consists of sentiment analysis on the pre-processed data from the sprint one. This was a small sprint and we manged to achieve this sprint in the estimated timeline. By using the lexicon and rule-based Vader sentiment analysis library, we performed the sentiment analysis on the extracted open dataset.

Sprint3:

Sprint-three consists of Model Training, using the training dataset that was derived from the sprint two. This sprint was also been challenging and hard time in achieving the accuracy of the model using the logistic regression classification machine learning algorithm. As mentioned earlier the dataset was a large file, we break down the dataset into 20,000 rows of small chunks. We tried with different combinations of chunks and observed the accuracy fluctuations around 60 to 65 percent. Finally, we achieved approximately 65 percent accuracy model.

Sprint4:

Sprint-four consists of test data streaming and performance evaluation. In this sprint, we encountered problems with Google API's. When we tried to fetch the latest reviews of the restaurant, we could not get the reviews by using google search API. Therefore, we used, google places API and the manual work around was given by hardcoding the place ID into the request to the google search

API. Later the python script was given to pipeline the process. Also, the conclusion was drawn by loading the test data into trained model and the average ratings were calculated from the predictions of the model. This sprint took more time than any other sprint with finalizing the conclusions.

Finally, we achieved the target in four sprints and the estimated timeline. However, we missed the individual sprint timelines and required to adjust the timelines in between the sprints.

LIMITATIONS

As part of this project there are few limitations. The free version of Google API's can fetch only five reviews per request and each key can do two requests per day.

The dataset used to train model is from Kaggle, and it is not updated on a regular basis, we tried requesting the Yelp to provide recent data to train model. But, it is available to premium users wherein we must pay for the data. Hence, we decided to use the dataset from Kaggle and left the updating part for future purposes.

FUTURE WORK

There can be lot of improvements to enhance the performance and the accuracy of the system. The following are some of the future scope of the project

- With Google API premium access, large data could be streamed.
- Different classification algorithms could be used to improve the accuracy of the model.
- Comparison algorithm can be developed to compare the ratings and reviews.
- Importantly, there can be a possibility of developing a front-end application for users and business.

  Example: Input the restaurant name, search different sources and extract data, and load the test data and predict the scores and emotions.

- Visualizations could be made to help the customers and business by providing the statistics.
- Data can be streamed in real time using Online Transaction Processing (OLTP) from Yelp and train the model on a regular schedule to improve the accuracy of the Model.

CRITICAL REVIEW

We believe, the developed Integrated Review System achieved the purpose of the aimed project. However, there are necessities of research and improvements in some parts such as exploring different classification algorithms and getting the latest datasets to improve the accuracy of the model. Also, some work needs to be done on fetching the test data from different sources.

ROLES

Hemanth kurra acted as Data Engineer and gathered the required data and organized and maintained proper architectures for future use, which includes cleaning and labelling of data. Also, handled model training and performed end calculations in the predictive analysis.

Srisaichand Singamaneni acted as a Data Scientist and handled all the process related to streaming, loading data, and performing predictive analysis. Also, classified end results from the predictive analysis and drawn conclusions.

We worked collectively and adjusted the missed timelines in different sprints to achieve the result.

GitHub URL

https://github.com/singamanenisrisai/Integrated-Review-System---CSCI-5408-Project.git

REFERENCES

[1]. Extract, transform, load. (2018, May 24). Retrieved from https://en.wikipedia.org/wiki/Extract,_transform,_load

[2]. Google APIs Explorer. (n.d.). Retrieved from https://developers.google.com/apis-explorer/#p/

[3]. Kaggle: Your Home for Data Science. (n.d.). Retrieved from https://www.kaggle.com/

[4]. Yelp Dataset. (n.d.). Retrieved from https://www.yelp.com/dataset

[5]. C. (2018, April 22). Cjhutto/vaderSentiment. Retrieved from https://github.com/cjhutto/vaderSentiment

[6]. Summary - Canadian Industry Statistics. (2017, May 31). Retrieved from https://www.ic.gc.ca/app/scr/app/cis/summary-sommaire/72;jsessionid=00016sZQOr0kEBege6Dgbmxw9p-:D197QKJL8

[7]. Documentation | Apache Spark. (n.d.). Retrieved from https://spark.apache.org/documentation.html