[**INFO7250 Engineering Big-Data Systems - Fall 2016**](https://blackboard.neu.edu/webapps/blackboard/execute/launcher?type=Course&id=_2476165_1&url=)

**Yelp: Predicting User Ratings for New Business**

Section - 01

**Team Members** – Kunal Singh Deora, Manasi Laddha, Swarna Satishkumar Dommeti, Bhavna Menghrajani, Sarthak Agarwal

**Professor –** Dino Konstantopoulos

December 12, 2016

Table of Contents

[**1.** **Introduction** 3](#_Toc469323380)

[**1.1** **Idea** 3](#_Toc469323381)

[**1.2** **Challenge** 3](#_Toc469323382)

[**1.3** **Approach** 3](#_Toc469323383)

[**2.** **Dataset Description** 4](#_Toc469323384)

[**3.** **Data Cleaning & Feature Engineering** 6](#_Toc469323385)

[**3.1** **Dataset Preparation** 6](#_Toc469323386)

[**3.2** **Feature Generation using Sentiment Analysis** 9](#_Toc469323387)

[**4.** **Machine Learning** 12](#_Toc469323388)

[**4.1.1** **Data Splitting** 12](#_Toc469323389)

[**4.1.2** **Algorithm** 12](#_Toc469323390)

[**4.1.3** **Finding new business** 14](#_Toc469323391)

[**4.1.4** **Prediction** 15](#_Toc469323392)

[**5.** **Application** 18](#_Toc469323393)

[**6.** **Challenges faced** 20](#_Toc469323394)

[**7.** **Tools & Technologies Used** 21](#_Toc469323395)

[**8.** **Conclusion** 22](#_Toc469323396)

[**9.** **Future scope** 23](#_Toc469323397)

[**10.** **Work Distribution** 24](#_Toc469323398)

[**11.** **References** 25](#_Toc469323399)

Chapter 1

1. **Introduction**

## **Idea**

The idea is to build a predictive system that predicts the degree of likeness of new food joints for users based on users past historical data on a global geographical scale.

e.g. If user travels to California from Boston, the system will predict which restaurants he will like in his five-mile radius and how will he rate the restaurant based on his/her previous pattern on Yelp.

## **Challenge**

Out of more than 100 features across five different data feeds, choose the appropriate features or create a new feature to get a high accuracy for each user.

## **Approach**

A high level diagram/flow of the project:

Data Cleaning & Feature Eng.

Machine

Learning

App

Mongo

DB

Sentiment Analysis

Hadoop - MapReduce

Yelp

Chapter 2

1. **Dataset Description**

The Yelp dataset consists of five data feeds but we primarily work on the below mentioned two feeds.

* Business – Information of all the businesses in Yelp.
* Review – Reviews of all the business.

The data is provided by Yelp on their website which consists of *85,539* businesses and *2,685,066* reviews. The file sizes were 75MB and 2.2GB respectively.

The data is provided as JSON files and their structure is as follows:

**BUSINESS**:



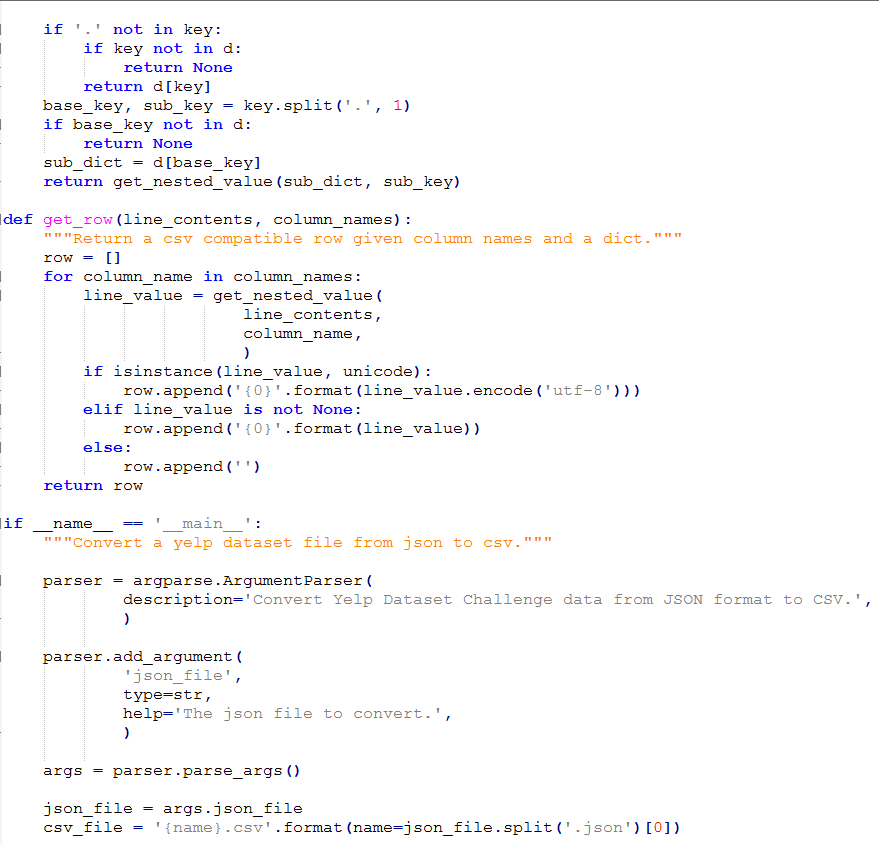
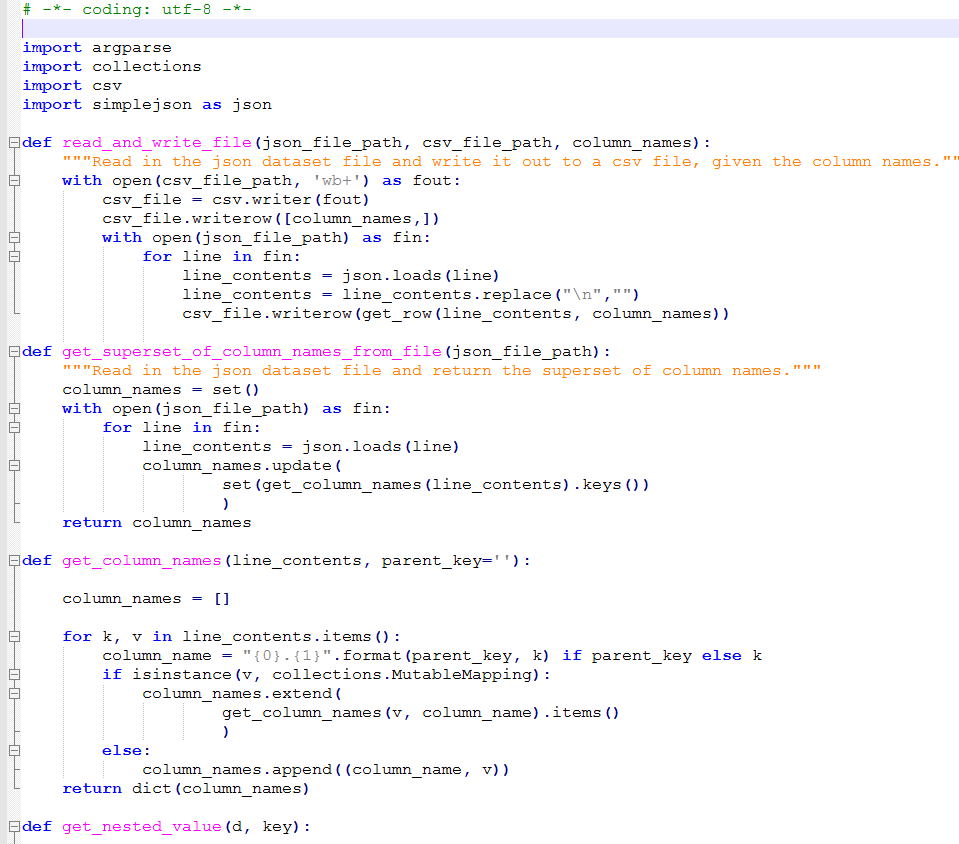
**REVIEW**:



Chapter 3

1. **Data Cleaning & Feature Engineering**

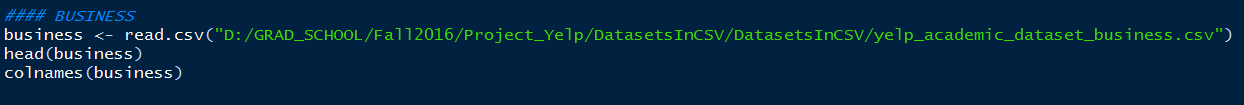
## **Dataset Preparation**

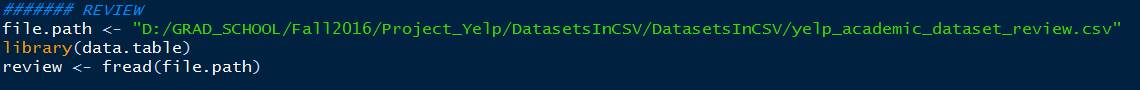
* + 1. Convert JSON to CSV We used Python to convert JSON to CSV and the code is as follows 
    2. Used R to analyze the data and prepare it for Machine Learning input.

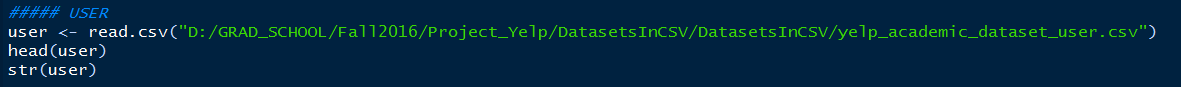
Following steps were performed:

* Filter the data related to food and similar categories.
* Filter the data for the users that have reviewed more than hundred businesses.
* Extract the relevant features out of 95 attributes in the Business dataset which will be used in our Machine learning algorithm.
* Rename the attributes to more meaningful names.
* Build and structure the data for our input to the Machine Learning Algorithm

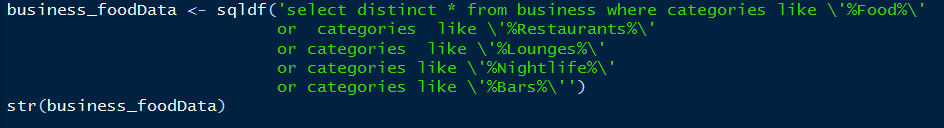
**Load the Business, Review and User CSVs into dataframe**



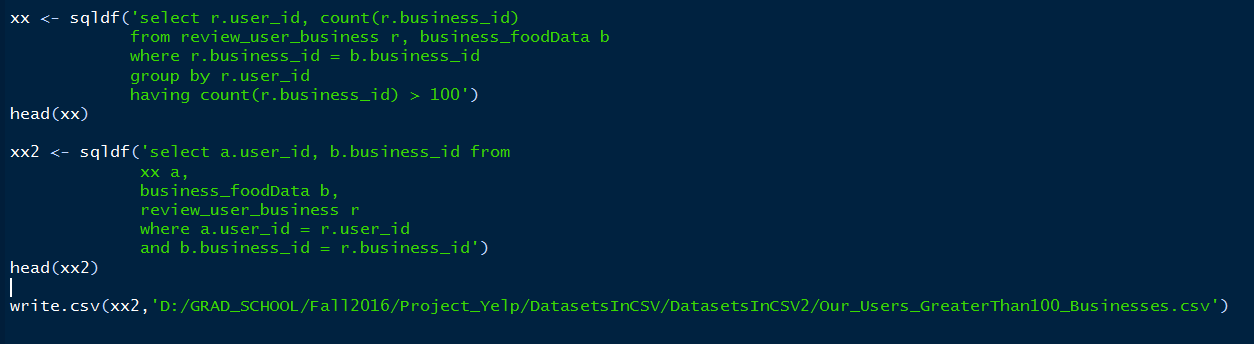




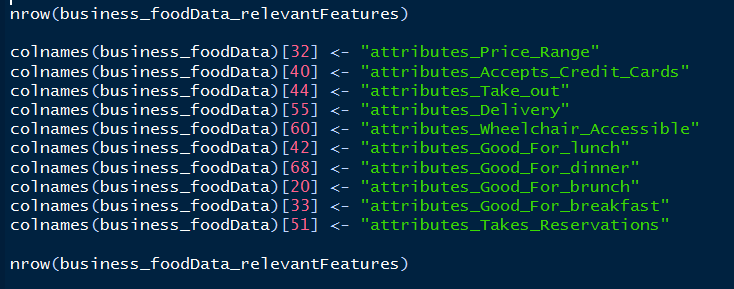
**Filter the data related to food and similar categories**



**Filter the data for the users that have reviewed more than hundred businesses**



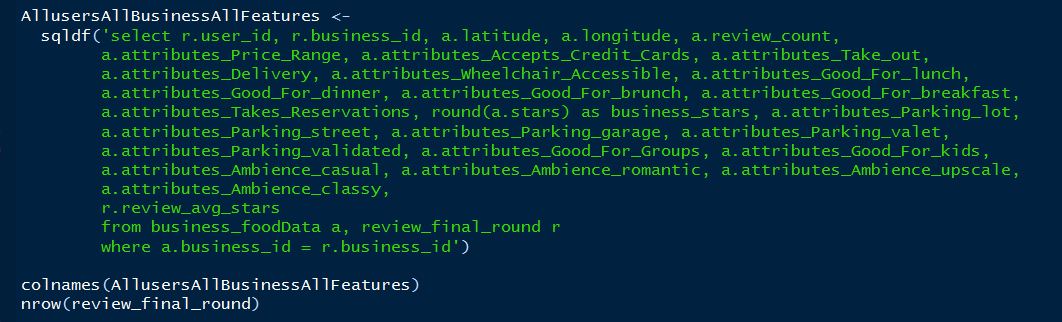
**Rename the attributes to more meaningful names.**



**Build and structure the data for our input to the Machine Learning Algorithm.**

We structured the data based on the input that our ML algorithm needs. The input to the algorithm will be something like this:

Users\_Id, Business\_id, ----Relevant Features---, Label



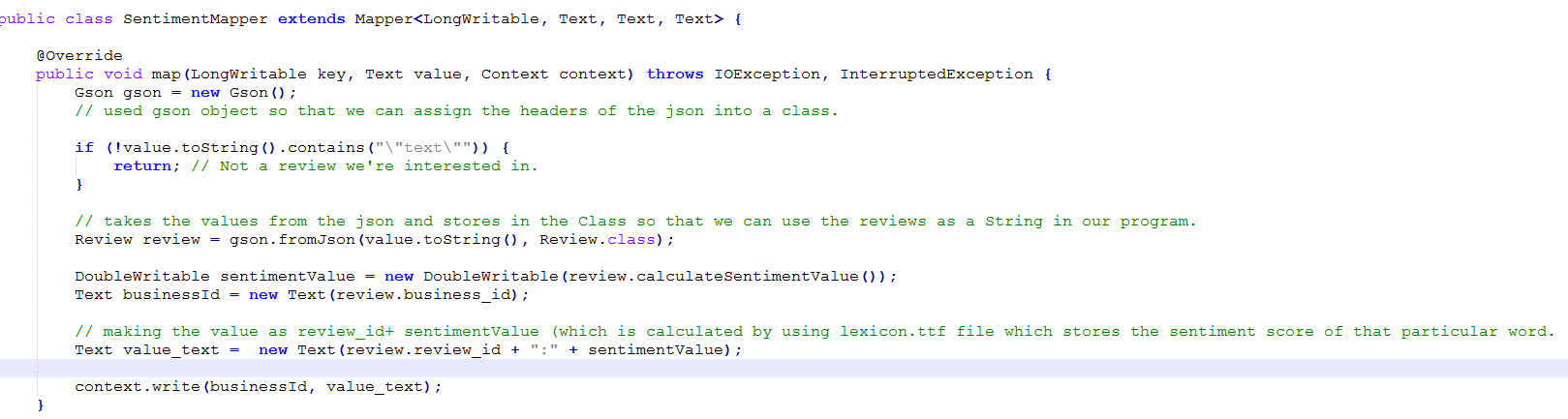
## **Feature Generation using Sentiment Analysis**

We generated a new feature using sentiment analysis of Reviews in Yelp dataset. This feature is a sentiment score of a business based on the sentiments of all the reviews that the business has reviewed. According to us this feature is a true representation of what user thinks about the business.

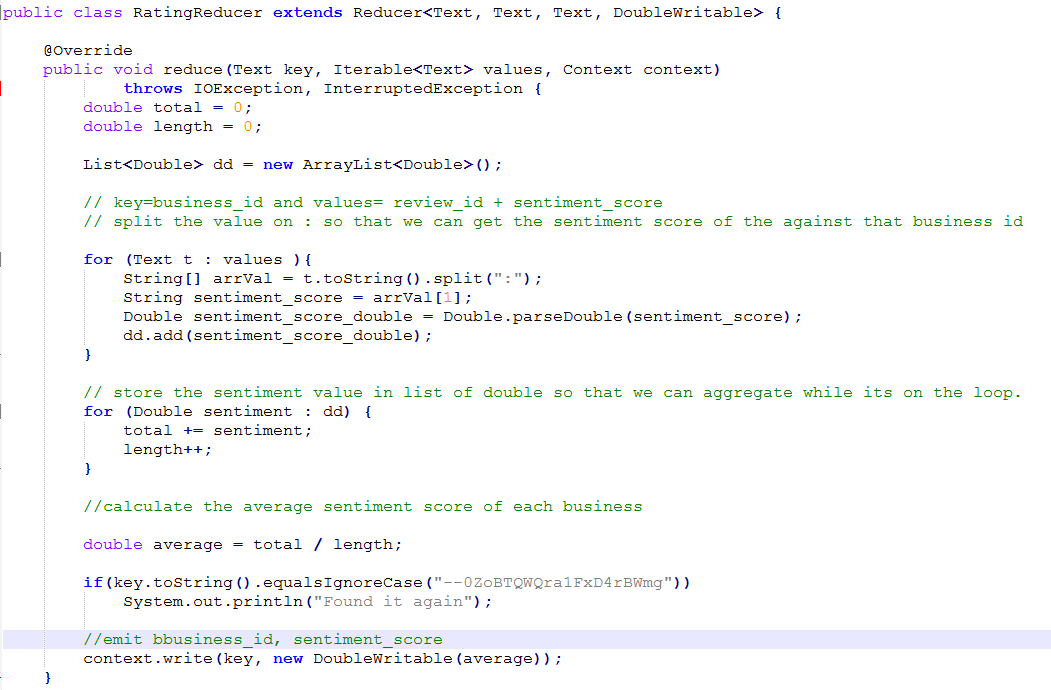
We wrote the code in R and tried performing sentiment analysis, but since the review file is approximately 2.2 GB and we have to process each word of each review, R was going out of memory. Hence we implemented it using Hadoop Map Reduce and ran the code in Distributed mode.

Our MapReduce algorithm is as follows:

1. The input to our Mapper is the reviews file in JSON format and used Google’s GSON library to read the file and assign it to the class.
2. In the Mapper we calculate the sentiment score of all the reviews and output the Business\_id as key and a combination of Review\_id and sentiment score as value.

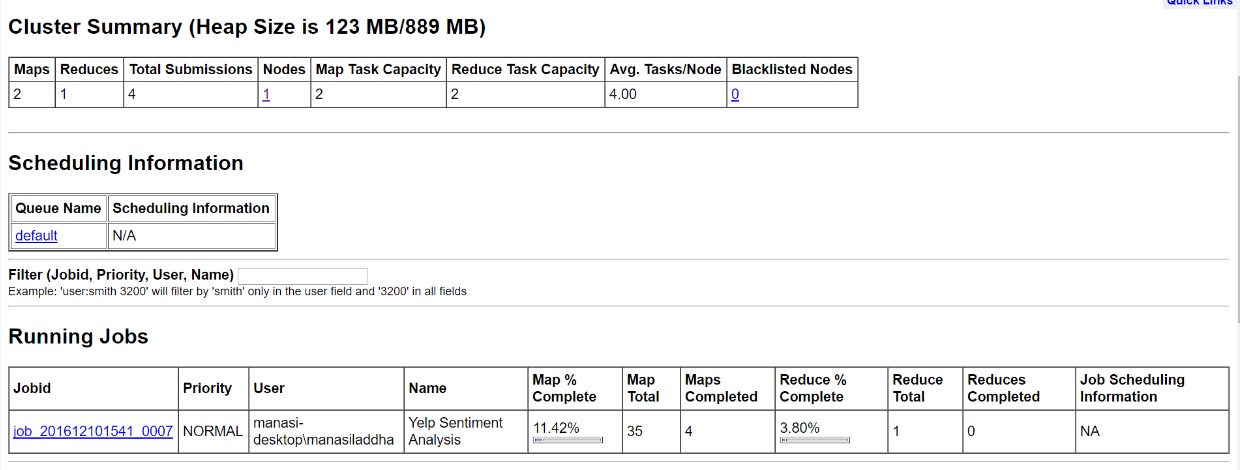


1. In the Reducer, we split the value and extracts the sentiment score and calculate the average sentiment score for each business.
2. The output of the reducer is the Business\_id and its sentiment score.

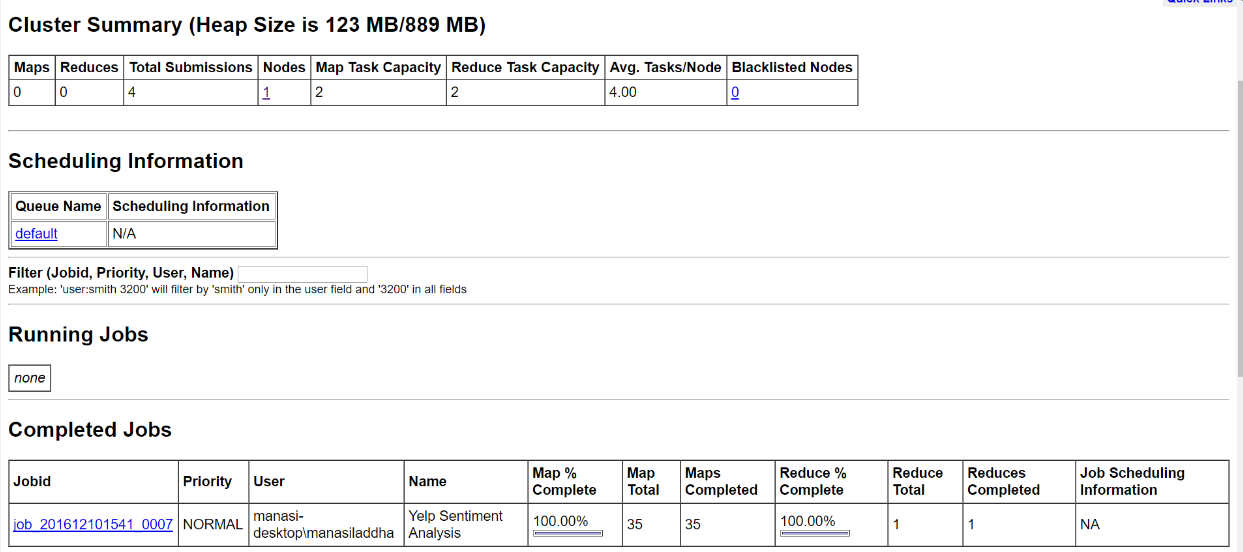


Since the reviews data was of 1.9 GB, we ran the map reduce job on fully distributed system. It took around 10 minutes to complete the job. Below are the screenshot of the job-tracker sandbox.

JOBTRACKER SAND – Fully Distributed Mode



JOBTRACKER SAND – Fully Distributed Mode



Chapter 4

1. **Machine Learning**
   * 1. **Data Splitting**

Using python, scikit-learn library, we split the data into training and test data using *train\_test\_split* function of *sklearn.model\_selection* library. We set the function to shuffle the data and then split it into 80 percent training and 20 percent test data.

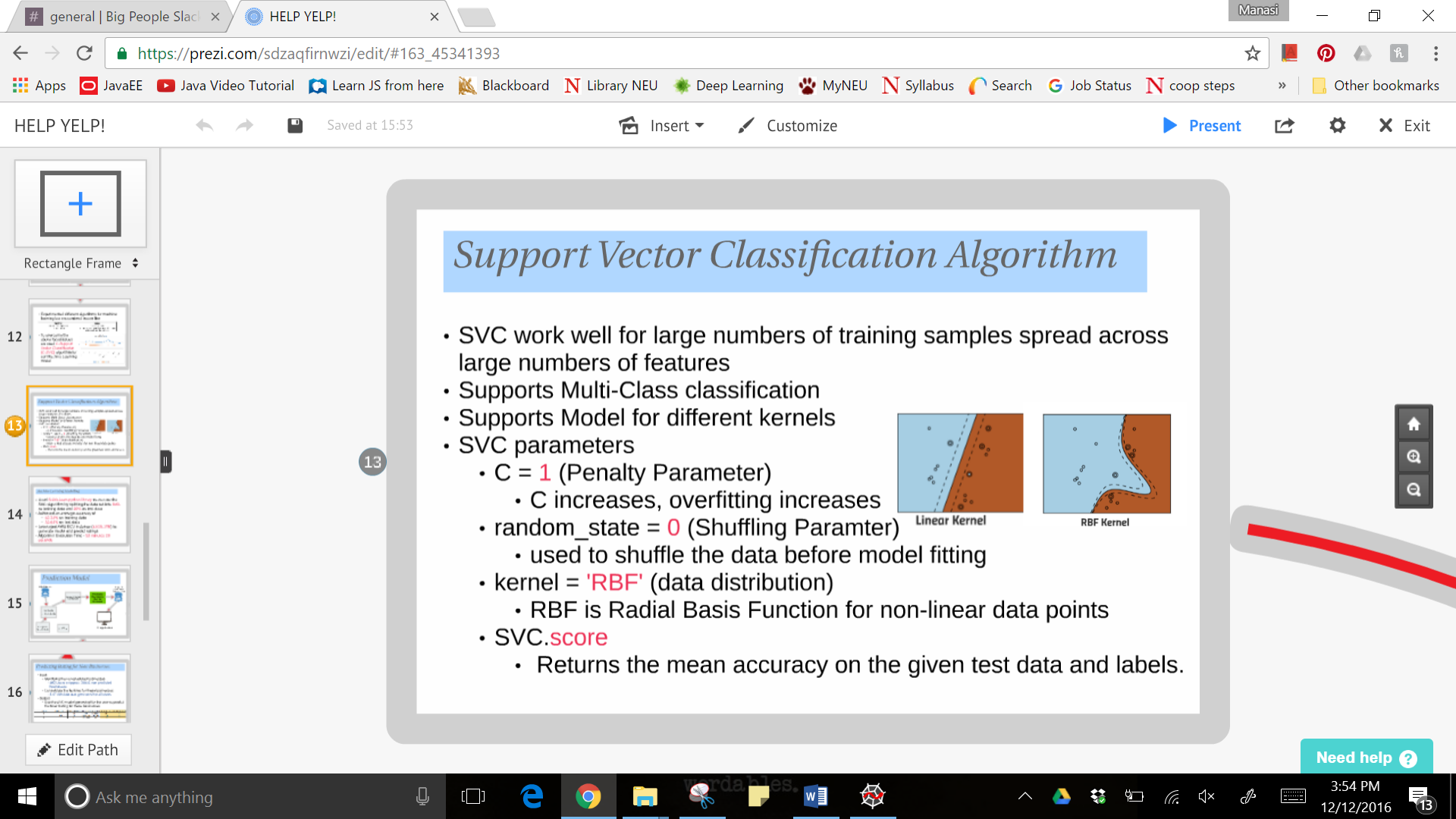
* + 1. **Algorithm**

We implemented the two models using scikit libraries: support vector machines (SVM) and random forest classifiers. These algorithms were implemented with the defaults from scikit, which can be found in their user guide. However, we investigated the types of kernels for SVM and the regularization parameters for SVM.

1. **Random Forest:**

While implementing Random forest we used the scikit implementation and the results were over fitted. i.e. the difference between the training and testing accuracy was huge.

1. **SVC (Support Vector Classification):**



Algorithm Overview

We experimented with SVM using scikit’s SVM with linear, polynomial and Gaussian kernels. We also performed a parameter sweep and the results were as follows:

|  |  |  |
| --- | --- | --- |
| **Training Accuracy** | **Testing Accuracy** | **Parameters** |
| 0.623397036 | 0.536425359 | c = 1, kernel = “rbf” |
| 0.714505147 | 0.433934534 | c =2, kernel = “rbf” |
| 0.600952494 | 0.427557664 | c = 1, kernel = “linear” |
| 0.775047514 | 0.413936467 | c =1, kernel = “rbf”, no preprocessing |
| 0.617386536 | 0.426587355 | c = 1, kernel = “rbf”, randomState = 0 |
| 0.577037933 | 0.423913976 | Linear SVC, pentaly = l2 |
| 0.545430948 | 0.413995582 | Linear SVC, pentaly = l2, loss = “hinge” |

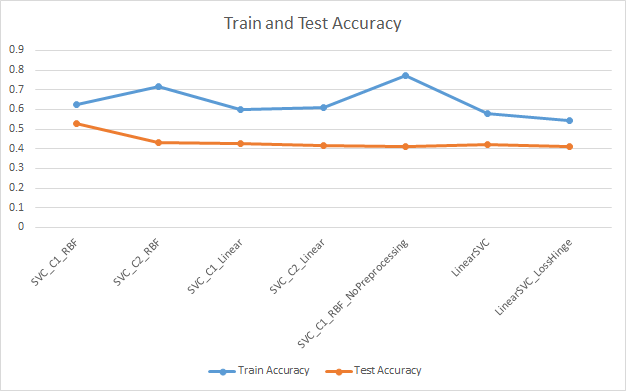


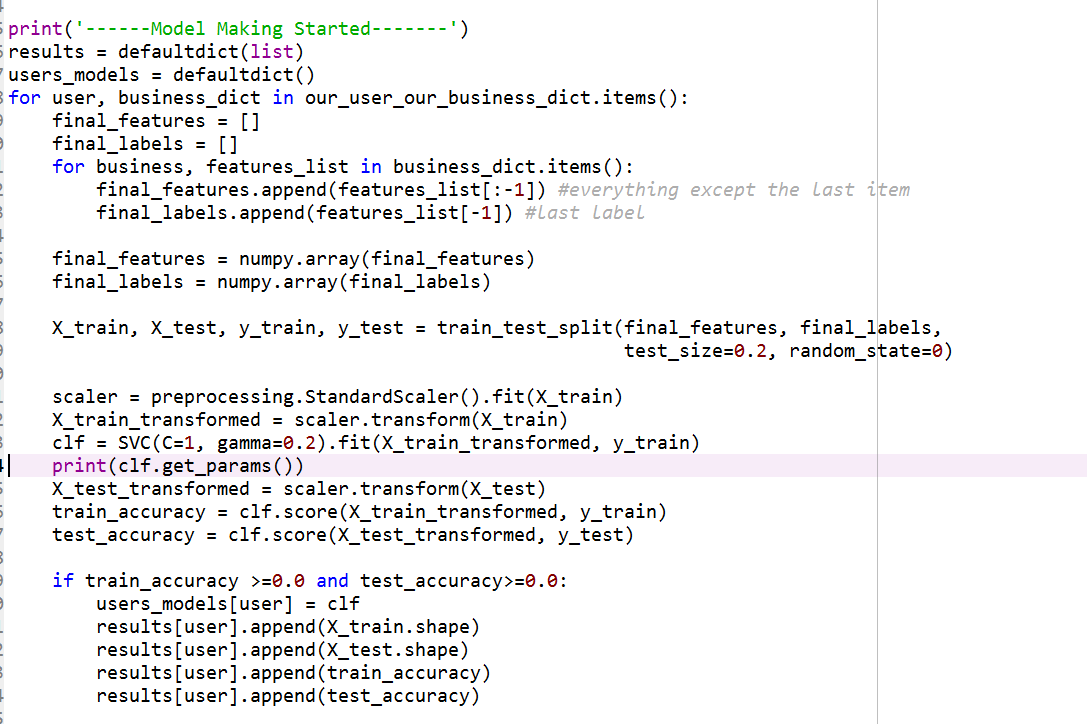
Figure : SVM results with different parameters

Based on the above results, we settled on C-SVC with RBF kernel to perform our prediction.

With C-SVC algorithm, our machine learning model acheives a decent accuracy of 62% and we are able to predict the rating of restaurants which have not been reviewed by a specific user nicely.

Radial Basis Function kernel network scale well to large numbers of features in the input space. The pre-processing module further provides a utility class StandardScalar that implements the Transformer API to compute the mean and standard deviation on the training set so as to be later reapply the same transformation on the testing set. The 'c' parameter trades off misclassifying of training examples against simplicity of the decision surface. When gamma is very small, the model is too constrained and cannot capture the complexity or “shape” of the data. The region of influence of any selected support vector would include the whole training set. The resulting model will behave similarly to a linear model with a set of hyperplanes that separate the centers of high density of any pair of two classes.

Python Code for Modelling

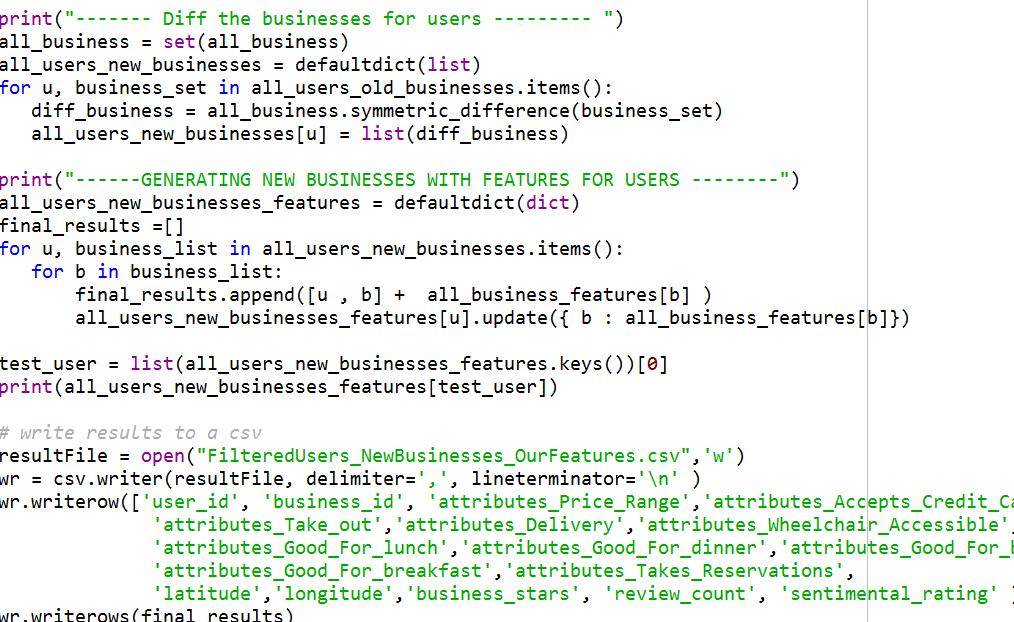


* + 1. **Finding new business**

After creating the model for all the users based on the features of their already reviewed businesses, our python code finds out the businesses that the user has not yet reviewed and on which we will perform the prediction.

There were 982 Users and approximately 36991 were non predicted businesses for each users. So the final file with all the features required for the prediction resulted in a 3.47 GB file on which the prediction was done.

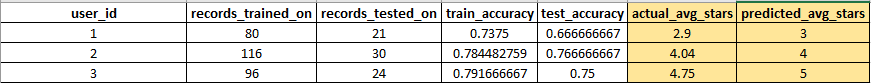
Python Code for new businesses identification



* + 1. **Prediction**

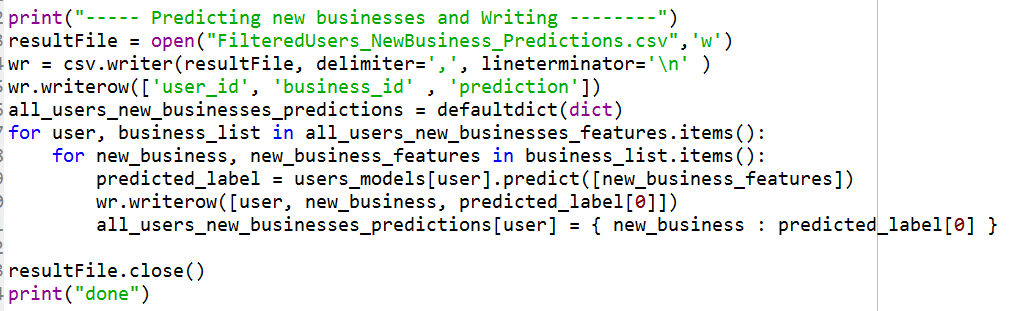
To predict, we use the SVC-model generated for each user to predict the User Rating for these non-predicted businesses. We use an AWS EC2 instance with 32GB, 1TB configuration and it took approx. 52 minutes 29seconds to complete the prediction.

For example,

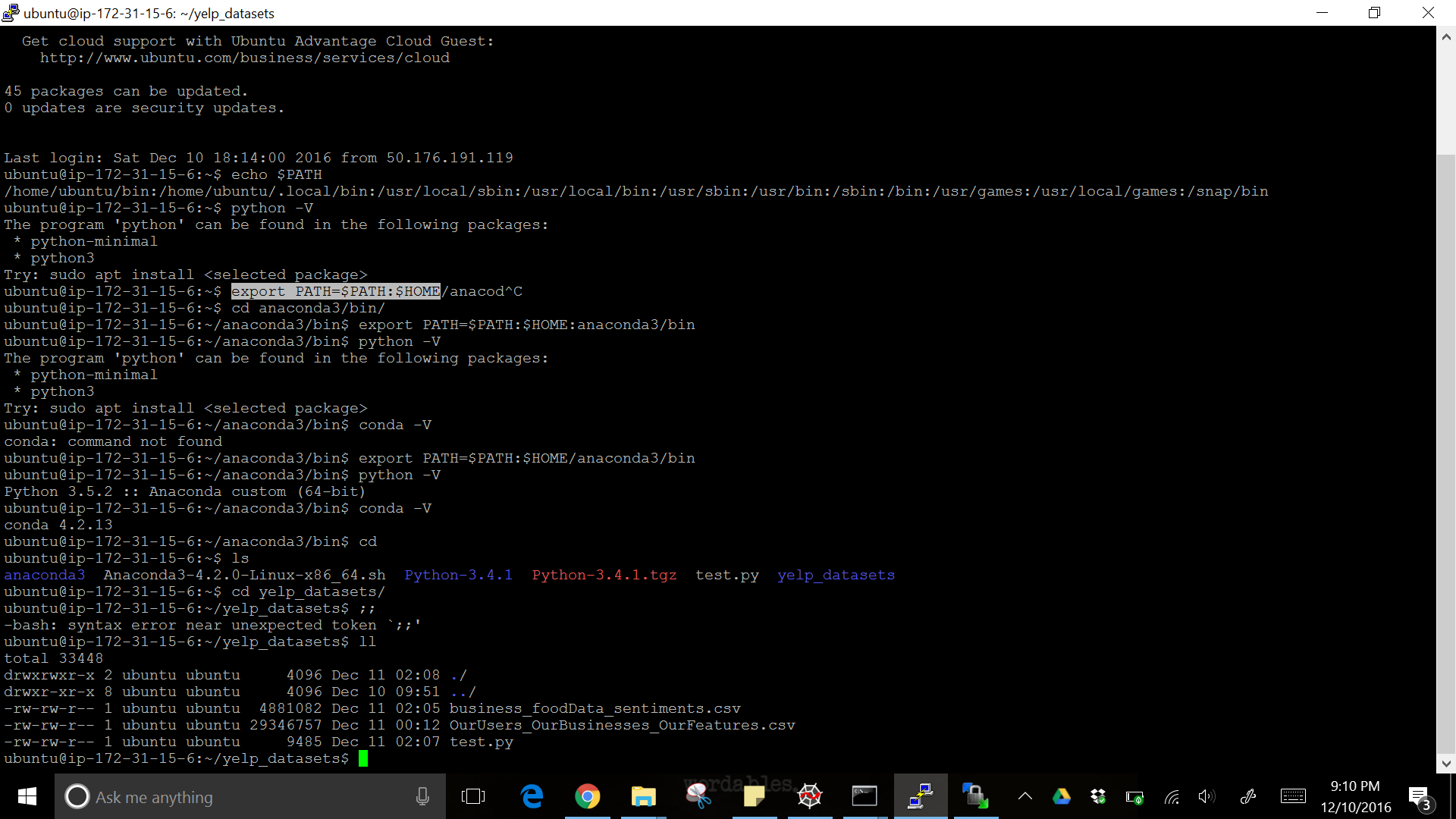


User\_id 1 was trained on 80 records and tested on 21 records and we got a train accuracy 73% and a test accuracy 66%. The actual average stars were 2.9 for the user and the model predicted an average stars of 3 for new businesses.

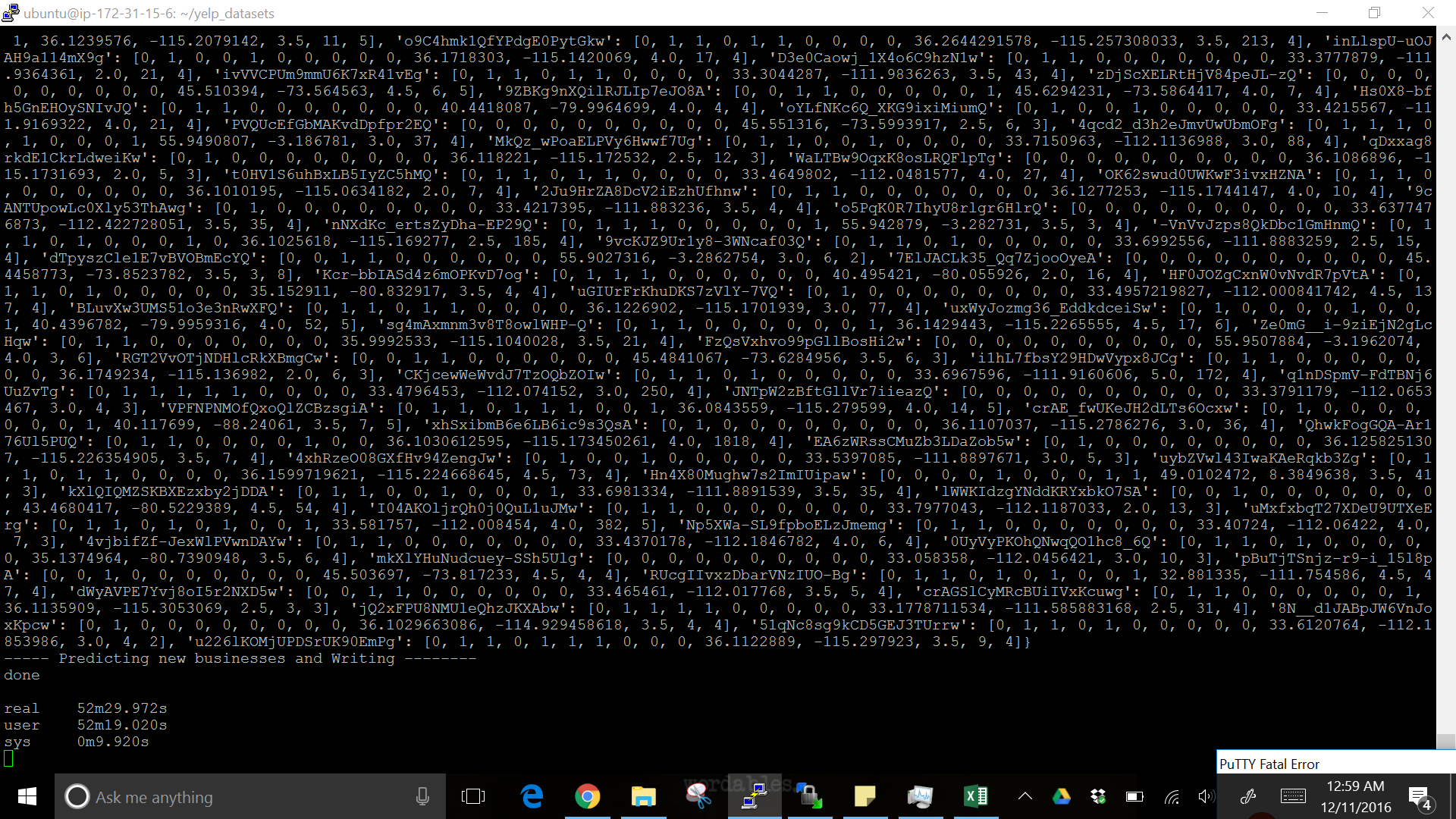
Python Code for Predicting



AWS EC2 Instance



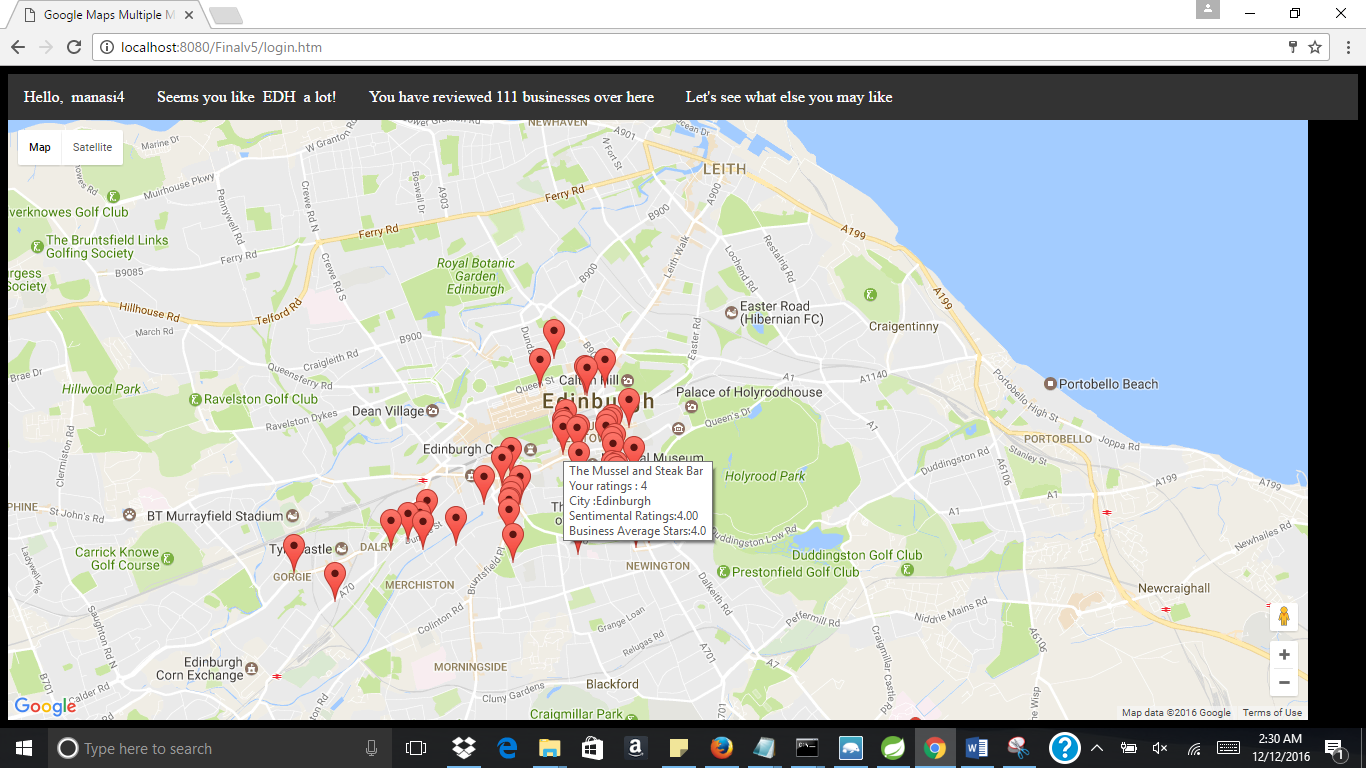
Machine Learning Modelling



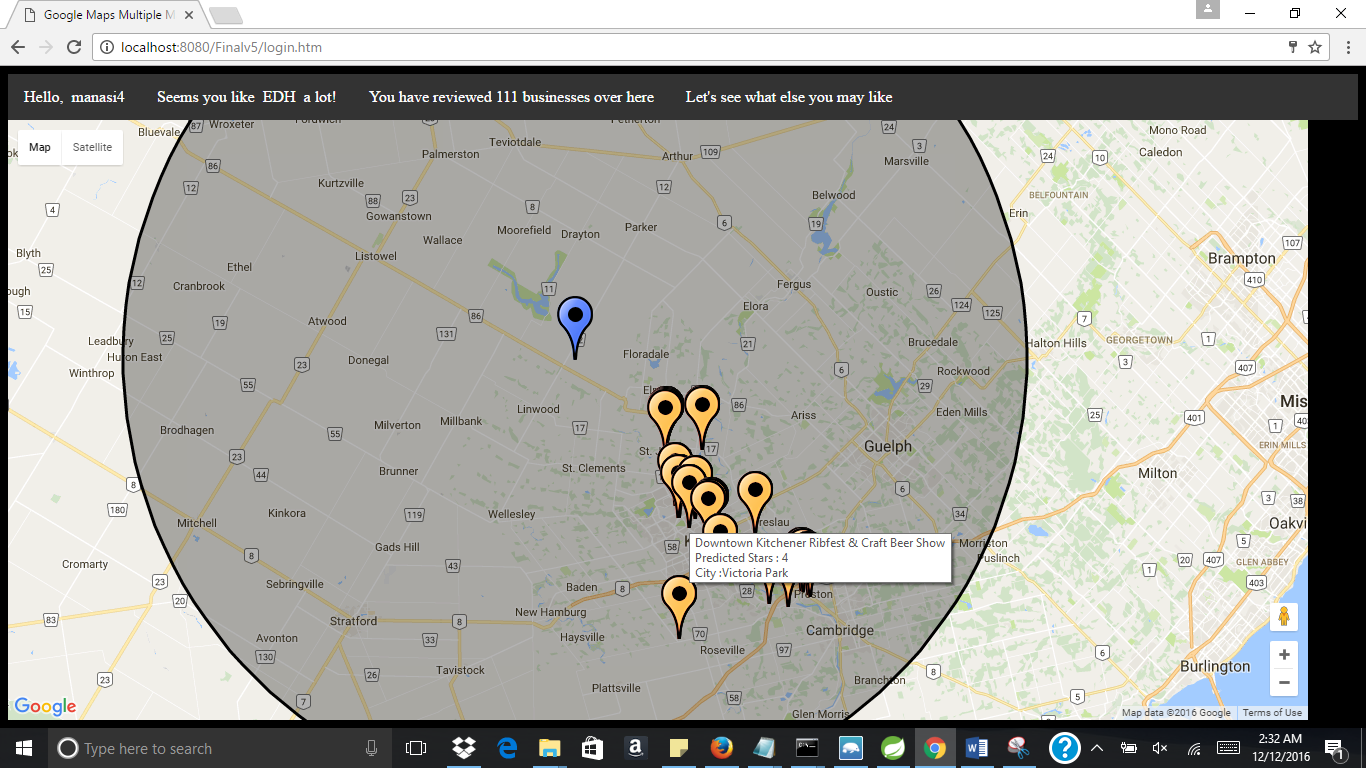
Chapter 5

1. **Application**

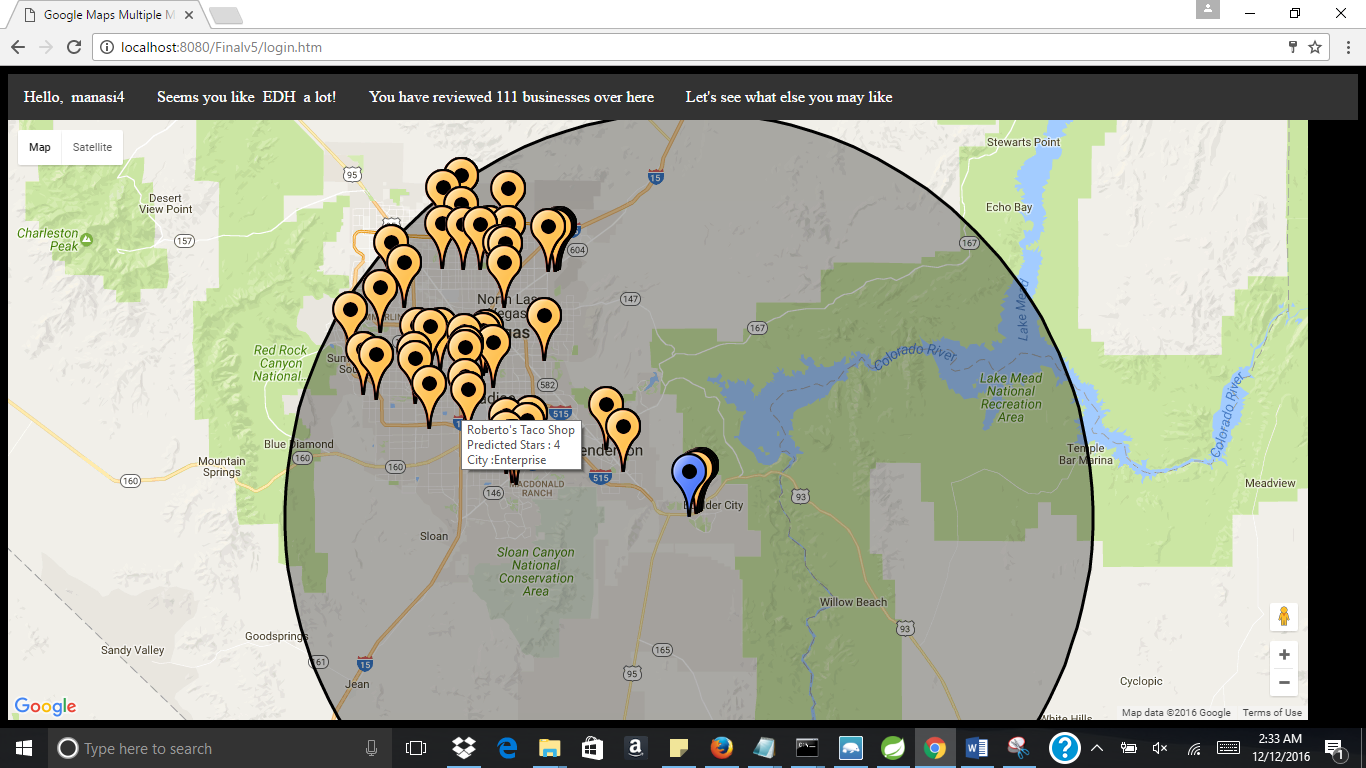
* Consider a user who had reviewed the food outlets in his region in Edinburgh, UK
* When the user logs in, he sees his rated businesses along with the sentiment score of his reviews and business average ratings
* This region is termed as hot spot of the user
* Google Maps API and Geometry Library were used



* When the user moves to different location, say Cambridge in US, he receives the predictions for the food businesses he has not reviewed yet
* These predictions will be based on businesses that he has reviewed in Edinburgh, UK



* The yellow markers show the new businesses
* Similarly, when he moves his location to Las Vegas, he sees a different prediction model



Chapter 6

1. **Challenges faced**

* **Selecting features:**

Since there were 95 features for the businesses, it was a challenge to select the relevant features to predict with high accuracy. To overcome this problem, we analyzed each attribute and eliminated the features that had a lot of Null values and which were not relevant for Food related businesses.

* **Loading the data**

We first faced this issue while loading the reviews data feed into R for data analysis since it was around 2gb. We tried increasing the heap memory of R but still it was not able to process. To overcome this problem we used the *fread* function of *data.table* library which is meant to import big data from regular delimited files directly into R, without any detours.

* **Processing the review data for sentiment analysis**

To perform the sentiment analysis on each and every word of all the reviews, we developed a R code which was not able to process because of memory limits. Therefore, we implemented the sentiment analysis algorithm on Hadoop MapReduce on AWS EC2 instance.

* **Data standardization**

Since the algorithm that we implemented assumes that all features are centered around zero and have variance in the same order, we were getting lower accuracy. To resolve this issue we used the *preprocessing* module of *scikit-learn* library for mean removal and variance scaling.

Chapter 7

1. **Tools & Technologies Used**

|  |  |
| --- | --- |
| **Technology** | **IDE/ Framework** |
| Python | Spyder |
| R | R Studio |
| MapReduce | Eclipse/ Hadoop |
| Spring MVC | STS |
| Hadoop cluster | AWS EC2 instance |
| MongoDB | Mongo 3T Chef |
|  |  |

Chapter 8

1. **Conclusion**

We were able to build a prediction model which is able to help a yelper by predicting the ratings for the new businesses at a new location based on his/her historical data. This way we were able to mirror his/her personal likings on a new geographical location and were able to map his model.



Chapter 9

1. **Future scope**

* All Business Categories

We can use this model to predict the user rating for all businesses which are a part of the data set provided by Yelp

* Social Graph Mining

We can analyze and understand the yelping pattern that can calculate the degree of similarity between the user and his friends. By this we can figure out which user is a trend setter for a particular business.

* #YELFIE

According to the current '#Yelfie' trend we can analyze the sentiment of a user by image recognition which can be included as a feature in our current data set.

Chapter 10

1. **Work Distribution**

|  |  |
| --- | --- |
| **Member Name** | **Work Responsibility/ Distribution** |
| Bhavna, Sarthak, Manasi | Research applicable Machine Learning Algorithms, Frameworks, Libraries and Technical Papers |
| Kunal, Swarna | Featuring Engineering – Data cleaning, data conversation and feature analysis |
| Bhavna, Sarthak, Manasi, Kunal, Swarna | Designing our Predictive Model Architecture of data analysis |
| Bhavna, Kunal, Sarthak | Setting up the cloud cluster |
| Manasi, Swarna, Sarthak | Developing and Testing the neural networks and machine learning techniques used in the model |
| Bhavna, Kunal, Manasi, Sarthak | Deploy the solution on code and tune the accuracy |
| Swarna, Manasi, Bhavna, Sarthak, Kunal | Virtualization of the data, documentation and presentation |

Chapter 11

1. **References**
2. Data Analytics using Yelp Data [Nevil Patel, Suraj Ponugoti, Doan H Nguyen]
3. LIBSVM: A Library for Support Vector Machines [Chih-Chung Chang and Chih-Jen Lin, Department of Computer Science, National Taiwan University, Taipei, Taiwan]
4. Applications of Machine Learning to Predict Yelp [Kyle Carbon, Kacyn Fujii, Prasanth Veerina]