Mini-Project 1: Yelp Business Rating Prediction using Pandas and Sklearn

Version 1.0

CSC215, Fall 2018

Sept 24th, 2018

Prepared by: Chandini Nagendra

Siddharth Chittora

Contents

[1. Problem Statement 3](#_Toc525183061)

[2. Design 3](#_Toc525183062)

[2.1. Data Preprocessing 3](#_Toc525183063)

[2.2. Supervised Learning Framework 3](#_Toc525183064)

[2.2.1. Problem: The system is trying to predict star rating of a business based on the review and the review count 3](#_Toc525183065)

[2.2.2. Extract Features as Input: 3](#_Toc525183066)

[2.2.3. Extract Output Labels: 3](#_Toc525183067)

[2.2.4. Models Used for Data Training 4](#_Toc525183068)

[2.3. Testing 4](#_Toc525183069)

[3. Task Division 4](#_Toc525183070)

[3.1. Chandini Nagendra: 4](#_Toc525183071)

[3.2. Siddharth Chittora 4](#_Toc525183072)

[4. Project Reflection 5](#_Toc525183073)

[5. Additional Features 5](#_Toc525183074)

# Problem Statement

Implementation of machine-learning based system to predict a business’s star rating using all the reviews of that business and review count. This problem is a regression problem where the output predicted is a continuous value, the star rating for the business.

# Design

## Data Preprocessing

* Json files used are Business.json, Review.json
* Load json into pandas
* Convert json into tabular format pandas can read
* Group all the reviews by each business and create a new dataframe, where each line is a business with all it’s reviews
* Join dataframes based on the value of a common column, we joined Business.json with review.json based on business\_id column
* Clean the reviews text data by removing stop words, punctuations and white spaces.
* Convert text data into TFIDF vectors
* Normalize the review count field so it is comparable
* Split the data into train and test data

## Supervised Learning Framework

### Problem: The system is trying to predict star rating of a business based on the review and the review count

### Extract Features as Input:

* BusinessID
* Review Count
* Review Text

### Extract Output Labels:

* Business star rating

### Models Used for Data Training

#### Linear Regression Model

##### Logistic Regression Model

##### Nearest Neighbor

##### Support Vector Machine

##### Multinomial Naïve Bayes

The KNN algorithm has a high prediction cost for large datasets. This is because in large datasets the cost of calculating distance between new point and each existing point becomes higher.

SVM does very well in text classification. It should be noted that Naïve Bayes does achieve very good scores as well. In short, given a text classification problem, any of the two could be used without a big compromise on the classification accuracy. In addition, factoring in training time, in some cases it would be advised to use Naïve Bayes instead of SVM

## Testing

Did not perform testing in this project

# Task Division

## Chandini Nagendra:

* Data Preprocessing
* Linear Regression
* Logistic Regression Model
* Multinomial Naïve Bayes
* Report

## Siddharth Chittora

* Data Preprocessing
* Linear Regression
* Support Vector Machine
* Multinomial Naïve Bayes
* Report

Discussed together on how to improve the model and came up with the following solution

# Project Reflection

Data Preprocessing took most of the time, in the process learnt to use the libraries, numpy, pandas, scikit learn

# Additional Features