Review Classification using Yelp Dataset

Team Members for CSCI-5951 BIG DATA SYSTEMS Project

Student Name	Student ID	Student Mail-Id
Bhavana Kurra	110294451	bhavana.kurra@ucdenver.edu
Vijayasimha Bheemireddy	109965858	vijayasimha.bheemireddy@ucdenver.edu
Praveen Vatambeti	110312985	praveen.vatambeti@ucdenver.edu
Navya Mogili	110315073	navya.mogili@ucdenver.edu

Problem Statement and Background

a) Problem Statement:

The internet has changed over the last decade that resulted in the generating of massive amounts of data. People often express their thoughts using social media. Some of the major platforms for expressing views are Google, Facebook and Pinterest etc. People use these social media platforms to convey their perceptions via posts, comments and discussion forums. The technique of recognizing and classifying the reviews which are in the form of text helps businesses in understanding consumer behaviour and introducing new marketing strategies.

b) Background:

Every organization attempts to learn about customers' reactions about their products. Understanding customer needs is the key to success for any business. But it's always challenging to learn about each type of customer because each and every customer is not the same. For the better understanding of customers, businesses segment customers based on their behaviour and understand the needs of customers belonging to each segment.

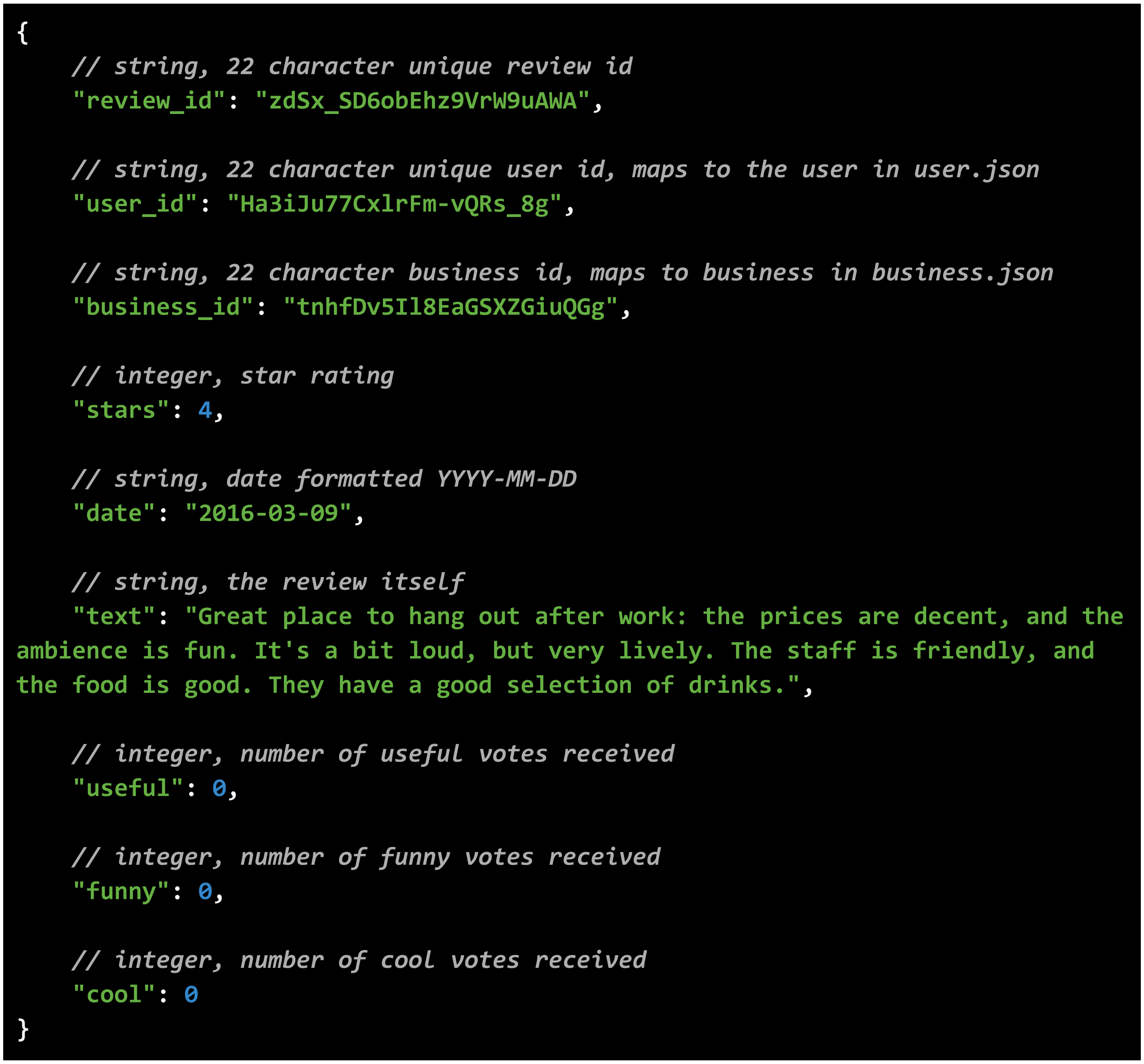
For example, let us segment the customers who purchase products during the discount season as the first group and the regular/loyal customers as the second group. This would help us in better understanding of consumer behaviour over different periods of time. Here, the consumer requirements might change between both the groups like the first group consumers might only care about the discounts over the product quality whereas the second group consumers might care more about product quality. Speaking with customers directly, interacting with them on social media, customer surveys are some of the effective ways to learn about customers of different kinds. Understanding the customer reviews is one such method. And segmenting the reviews into positive and negative would help a business in understanding its strength and the areas in which the business should improve. Further analysis of each review sentiment(such as funny, sad etc) rather than positive and negative reviews would help us in better understanding of customers.

c) Objective:

Creation of a machine learning model with and without using big data tools to classify the Yelp reviews data into positive and Negative reviews.

d) Data and its characteristics:

This dataset is a subset of Yelp's reviews data which has 8.6M reviews . It was originally put together for the Yelp Dataset Challenge which is a chance for students to conduct research or analysis on Yelp's data and share their discoveries. The features in the sample data and their description is as shown below:



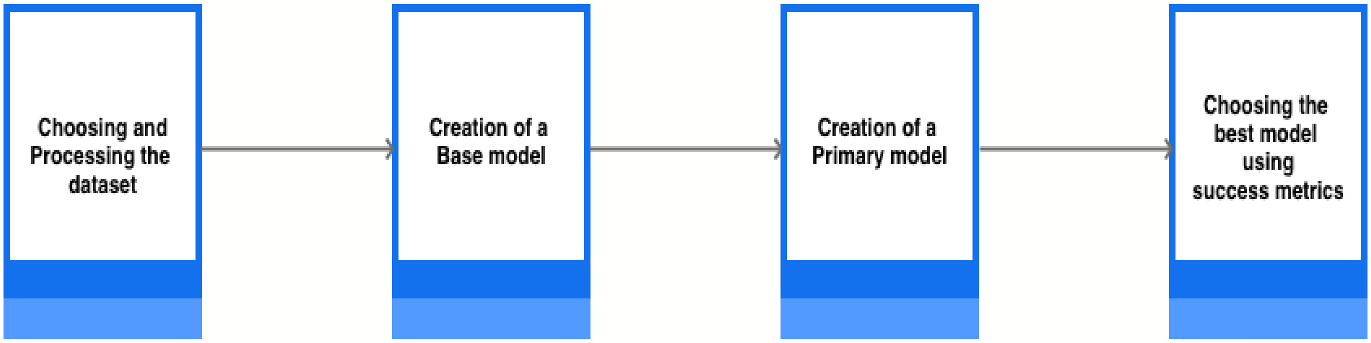
```
{
    // string, 22 character unique review id
    "review_id": "zdSx_SD6obEhz9VrW9uAWA",

    // string, 22 character unique user id, maps to the user in user.json
    "user_id": "Ha3iJu77CxlrFm-vQRs_8g",

    // string, 22 character business id, maps to business in business.json
    "business_id": "tnhfDv5Il8EaGSXZGiuQGg",

    // integer, star rating
    "stars": 4,

    // string, date formatted YYYY-MM-DD
    "date": "2016-03-09",

    // string, the review itself
    "text": "Great place to hang out after work: the prices are decent, and the
ambience is fun. It's a bit loud, but very lively. The staff is friendly, and
the food is good. They have a good selection of drinks.",

    // integer, number of useful votes received
    "useful": 0,

    // integer, number of funny votes received
    "funny": 0,

    // integer, number of cool votes received
    "cool": 0
}
```

e) Success Measures:

Evaluating the performance of the project is essential for any project. Though some algorithms might give impressive results in terms of performance , when working with data it is key to measure algorithms using metrics such as accuracy. Therefore we are using the test accuracy of a model as our success metric.

Architecture / Steps to be followed –

1. Choosing and cleaning/processing the dataset.
2. Creation of base models using Machine learning libraries such as sklearn.
3. Creation of a primary model using pySpark machine learning tools based on base models performances.
4. Choosing the best model based on the success metrics chosen.


Choosing and Processing the dataset
Creation of a Base model
Creation of a Primary model
Choosing the best model using success metrics

Methods

Step 1 - Data Preprocessing

The Data Preprocessing has two stages. The first stage consists of reading the yelp dataset which has 8.6M records as chunks, selecting the necessary features for the creation of the model and storing those chunks individually. Whereas the second stage consists of selecting the chunks individually and processing the text such that we can pass them through a machine learning model. Here, we have considered the first chunk which has 1M reviews and passed it through the next steps of architecture. And also we have considered only the ‘text’ and ‘stars’ columns which would be useful for review classification where the ‘text’ column consists of actual reviews and the ‘stars’ column will be used for labeling each review.

The second data processing stage has multiple steps such as lower case conversion, removal of punctuations, removal of stop words, lemmatization and labelling. The implementation of each step is as follows -

1. **Lower case conversion**

   Conversion of the text into lower case in order to avoid the computer treating the same words written in different cases as different entities.

2. **Removal of punctuations**

   Removing the punctuations in order to get rid of the computer treating the same words with different punctuations as different entities. For example, the text model treats the ‘study,’ and ‘study’ as different words because of the differences in punctuation.

3. **Removal of stop words**

   Stop words are a collection of words that do not add much meaning to the sentences but occur frequently in a language. We have used nltk library stopwords for language english and customized it for our problem statement. That is there are 179 stopwords that belong to the english language as per the nltk corpus, whereas we have eliminated few words such as 'against','not', "don't", "aren't", 'couldn', "couldn't" etc from those stopwords list which has brought the stop words list length to the 143. We have prevented the elimination of words such as “against", “not” etc from a sentence because elimination of these words would convert a negatively sounding sentence to a positively sounding sentence. For example, eliminating the word “don’t” from a sentence “I don’t like apple” would change the sentence to “I like apple”.

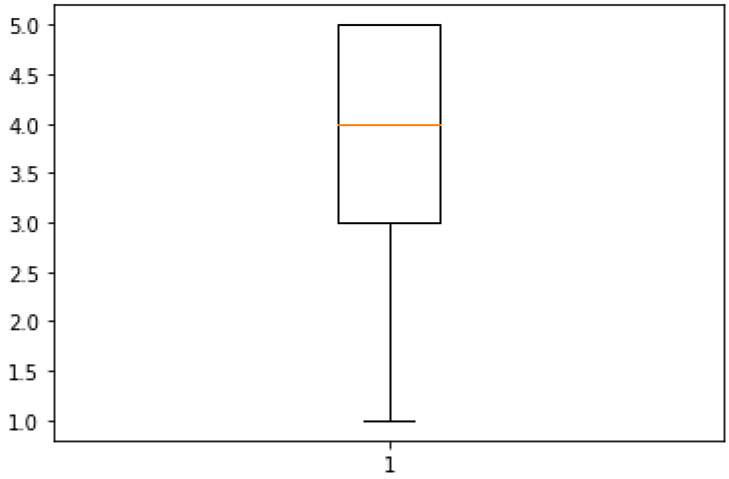
4. **Lemmatization**

   Lemmatization is the process of reducing a word into its root word. Stemming also does the same thing as lemmatization but the root word might not necessarily be valid as per dictionary. For example, stemming reduces the words “studying”, “studies” and “study” to “studi” where “studi” is not a valid English word. This is the reason behind us choosing lemmatization over the stemming in order to reduce a word to its root word.

   For lemmatization of text, we have used both the ‘nltk’ and ‘pattern’ libraries of python and found the ‘pattern’ library lemmatization to be the best one among the both.

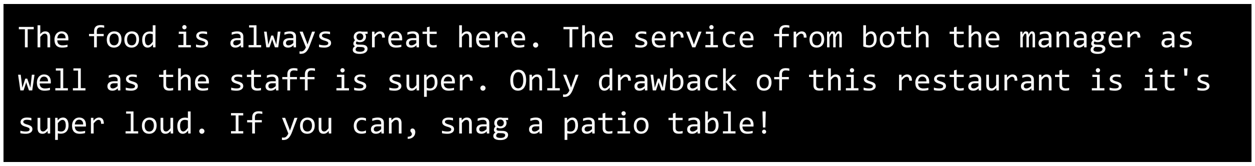
5. **Labelling**

   We have labeled each review based on the number of stars associated with that review. In order to understand the threshold based on which we can divide the positive and negative reviews, we have analyzed the distribution of the stars which is as shown in the below image -



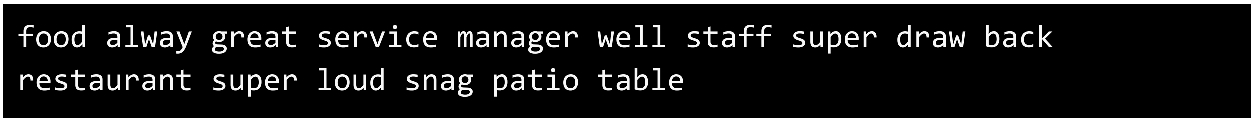
The mean of stars is 3.73 and the median of stars is 4, Inorder to decrease the class imbalance we have taken the threshold value to be 3.5. That is the reviews having stars/rating > 3.5 are labeled as positive reviews and the reviews having the stars/rating <=3.5 are labeled as negative reviews.

The below attached image has a review before passing the text through the second stage of data processing-



```
The food is always great here. The service from both the manager as
well as the staff is super. Only drawback of this restaurant is it's
super loud. If you can, snag a patio table!
```

The below attached image has a review after passing the text through the second stage of data processing -



```
food alway great service manager well staff super draw back
restaurant super loud snag patio table
```

Note :

1. Before passing the reviews that have passed through the second stage data processing into a machine learning model, we have applied the **count vectorization** inorder to convert the text to integers.
2. We have created a column where quotations are added at the beginning and ending of each review for the proper readability of columns in the pySpark.

Step 2 - Creation of a Baseline model

Creation of a baseline model helps us in understanding our data which is very helpful for developing our primary model. We have created the baseline models using the machine learning libraries such as sklearn, numpy etc. Based on the baseline models performances we have chosen an algorithm that should be used for classification in our primary model. We have implemented Logistic Regression and Decision Tree models as our baseline models.

1. Logistic Regression :

a) Definition

The process of modeling the probability of a discrete outcome given an input variable is called logistic regression. Logistic regression is a supervised machine learning algorithm which is used for classification.

b) Parameter choices

penalty='l2', dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=0, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None are the parameter choices that are considered for the logistic regression model.

2. Decision Tree Classifier:

c) Definition

Decision Tree is a supervised machine learning algorithm that is used for both classification and regression where the data is continuously split according to a certain parameter(we have used entropy as the splitting parameter in our code). In our project, we have used the decision tree classifier.

d) Parameter choices

criterion='entropy', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=0, max_leaf_nodes=None, min_impurity_decrease=0.0, class_weight=None, ccp_alpha=0.0 are the parameter choices considered for the decision tree model.

Note : we have maintained class balance while creating a model by taking an equal number of positive and negative records. We have run both the above mentioned logistic regression and Decision tree models over 0.67M records. And we have taken a train test split of 80% and 20%.

Step 3 - Creation of a Primary model

Based on the previously created baseline models we have chosen to implement logistic regression using the pyspark.ml as our primary model.

a) Parameter choices for spark session :

The following are the parameter choices for the spark session -

```
 ('spark.app.name', 'Colab'),
 ('spark.driver.host', '6db2e71ff6fe'),
 ('spark.master', 'local[4]'),
 ('spark.executor.id', 'driver'),
 ('spark.sql.warehouse.dir', 'file:/content/spark-warehouse'),
 ('spark.driver.port', '41307'),
 ('spark.ui.port', '4050'),
 ('spark.rdd.compress', 'True'),
 ('spark.app.id', 'local-1639188694967'),
 ('spark.serializer.objectStreamReset', '100'),
 ('spark.submit.pyFiles', ''),
 ('spark.submit.deployMode', 'client'),
 ('spark.ui.showConsoleProgress', 'true'),
 ('spark.app.startTime', '1639188693150')
```

b) Parameter choices for logistic regression model :

We have taken all the default parameters that exist for the Logistic Regression model that exists in the pyspark.ml.classification library by only changing the “maxIter” parameter to 10.

Note : we have maintained class balance while creating a model by taking an equal number of positive and negative records. We have run both the above mentioned logistic regression model over 0.67M records. And we have taken a train test split of 80% and 20%.

Step 4 - Choosing the best model using success metrics

We have chosen the test accuracy of the model as a success metric. The results and the best model accuracy will be discussed in the results section.

Results

The below table summarizes the outcomes of all the models created -

Model	Training Accuracy	Test Accuracy	F1-score
Base model1(Logistic Regression)	0.8271	0.8280	0.8274
Base model2(Decision Tree)	0.9988	0.7311	0.7332
Primary model1(logistic regression)	0.9469	0.8913	0.9179

We are considering test accuracy inorder to determine the best model. From the above table, we can infer that the best model is the Primary model(Logistic Regression using PySpark) having a test accuracy of 89.13%.

Big Data Systems and Tools

The following are the tools that are used in the process of developing this project

- Google Colab
- Python 3.8
- Apache Spark
- Java Development Kit

The data stored in the google drive is accessed in the google colab by mounting the drive. Then the Java Development Kit, Spark and Hadoop are downloaded in the google colab itself. After the installations are successful we have built the machine learning models using the pyspark.ml and sklearn libraries. We have chosen spark.ml over spark.mllib since spark.mllib has API's on top of RDDs whereas spark.ml has API's built on top of DataFrames.

Lessons Learnt:

- One should be vigilant while installing spark in a python environment since it requires a lot of installations.
- Errors faced in the process of developing the project are time taking and not easy to understand.
- Overfitting a model leads to wrong predictions.
- Sometimes, simpler machine learning models such as logistic regression works better for classification than the higher level machine learning models such as decision trees.

Summary:

This project aims to process Yelp dataset and classify them into positive and negative reviews. We have created 3 models in total, two models(logistic regression, decision tree) using the sklearn library of python and one model(logistic regression) using the pyspark.ml library. The model proposed in this report helps us to understand what customers think about the restaurants and their services. This analysis helps the business owners make key decisions regarding their marketing strategy. The Yelp dataset is preprocessed to remove unnecessary spaces, punctuations, stopwords so that the data will be more accurate for further processing and analysis. The preprocessed data is then modeled and estimated using two different sets of tools i.e. using python ML libraries alone and with spark cluster. Both the models perform well but the model built with spark cluster is more precise and accurate when compared to the first one. This can be further trained to detect fake reviews by using keywords which are more positive or more negative which can be taken as future scope.

Team Contribution:

Team Member	Role	Contribution
Bhavana Kurra	Building a Machine Learning Model	30%
Vijayasimha Bheemireddy	Analyzing and Creating Visualizations	25%
Praveen Vatambeti	Building a Machine Learning Model	25%
Navya Mogili	Extraction and Processing of Data	20%