Sentiment Analysis on Amazon Book Reviews

CMPE - 256 Large Scale Analytics Term

Project Report

Team SentiMap

Mrinmayi Gavali <mrinmayi.gavali@sjsu.edu>

FNU Aprajita <fnu.aprajita@sjsu.edu>

Priyanka Kumar <priyanka.kumar@sjsu.edu>

Introduction
- Motivation
- Objective

System design
- System architecture
- Visualizations

Experiments / Proof of concept
- Datasets used
- Data pre-processing

Methodology

Graphs

Analysis of results

Discussion and conclusion

Project plan

Introduction

Motivation

Sentiment analysis is often used to derive the emotion/ opinion expressed in a text. It has wide applications, including analysis of product reviews, discovering a brand's presence online and people's opinion on a subject online. Analyzing sentiments can prove beneficial in improving sales of a product and thereby aid product recommendations. The goal of this project is to perform sentiment analysis on Amazon Book Reviews using text-processing, tf-idf, vectorization and classification. The dataset consists of 8.9 million reviews of books sold on Amazon.com which is available here. Sentiment analysis of reviews can in turn be useful for improving product recommendations.

Objective

The objective of this project is to perform sentiment analysis (positive and negative sentiment) on reviews of books sold on Amazon.com. The trained model can be used to

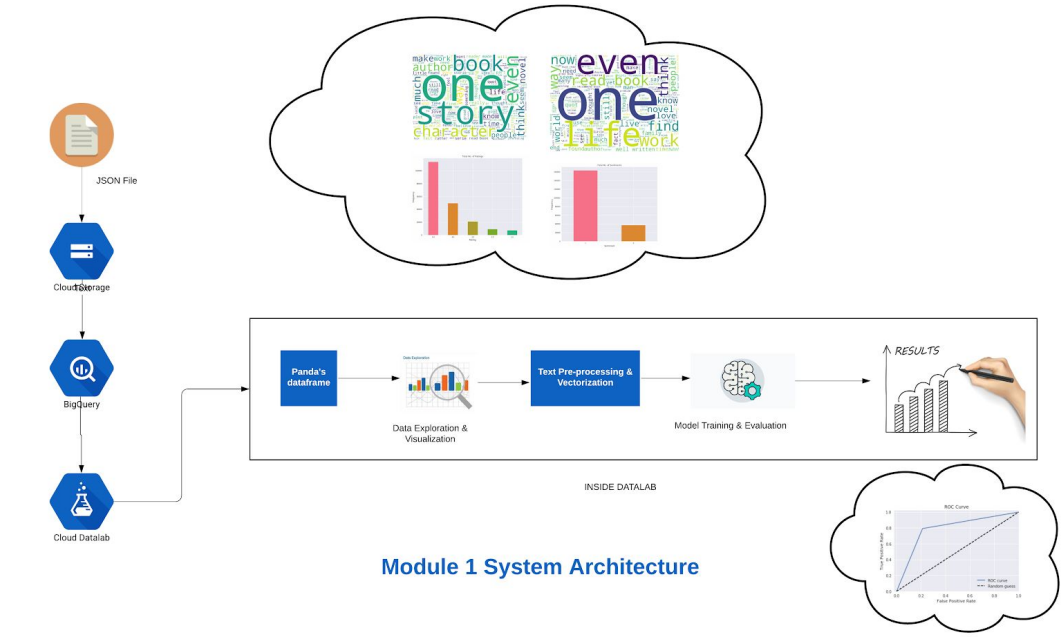
predict users' sentiment based on their online reviews. These sentiments can used as a metadata in building book recommender systems. The overall flow of the model is as follows:

- ❏ Step 1 - Preprocess raw text reviews to clean reviews
- ❏ Step 2 - TF-IDF transformation to encode reviews into numerical representations
- ❏ Step 3 - Fit numerical representations of reviews to machine learning algorithms
- ❏ Step 4 - Predict using test set of reviews and analyze results

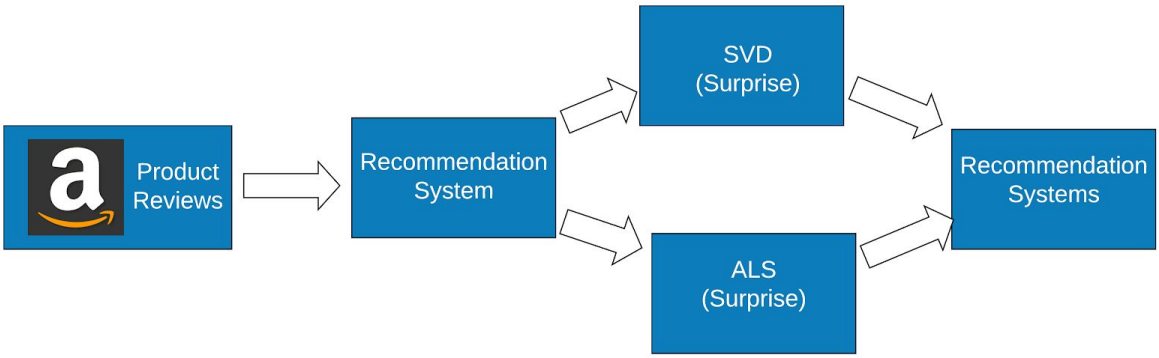
System design

	Module 1 (Sentiment analysis)	**Module 2 (Recommender system)**
Algorithms used	● Logistic Regression (***Performed Best***) ● Random Forest ● Multinomial Naive Bayes	● Singular Value Decomposition (SVD) ● Baseline
Tools and techniques used	1) Google Cloud Storage 2) Google BigQuery 3) Google Datalab 4) Packages used: bq, Pandas, NumPy, NLTK, WordCloud, Re, Scikit-Learn	1) Google Colab 2) Packages used: Pandas, Surprise, Scikit-Learn

System architecture



Module 1 System Architecture



Module 2- System Architecture

Visualizations

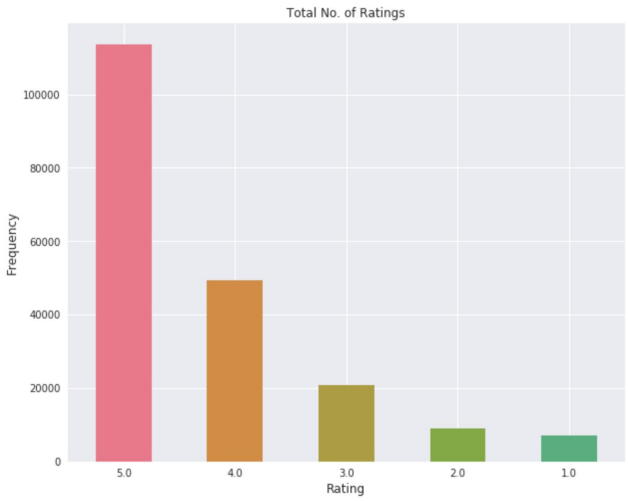


Fig. a) Distribution of ratings

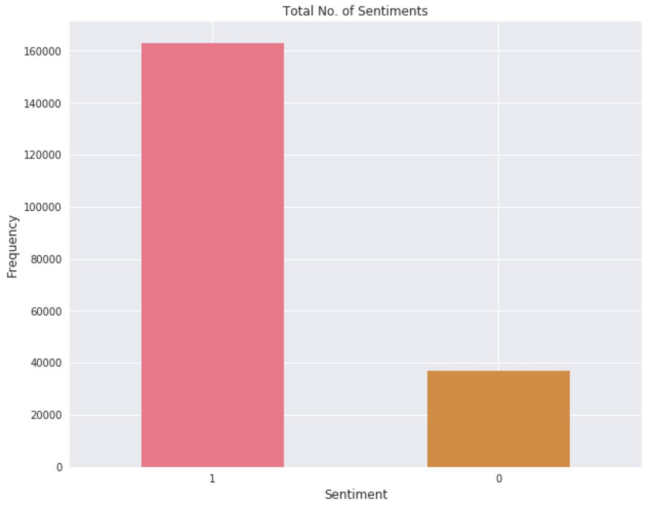


Fig. b) Distribution of sentiments

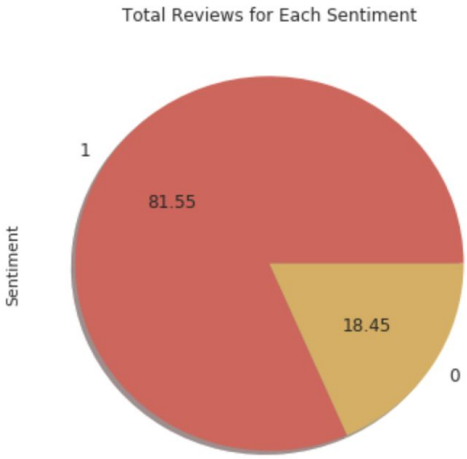


Fig. b) Distribution of Sentiments

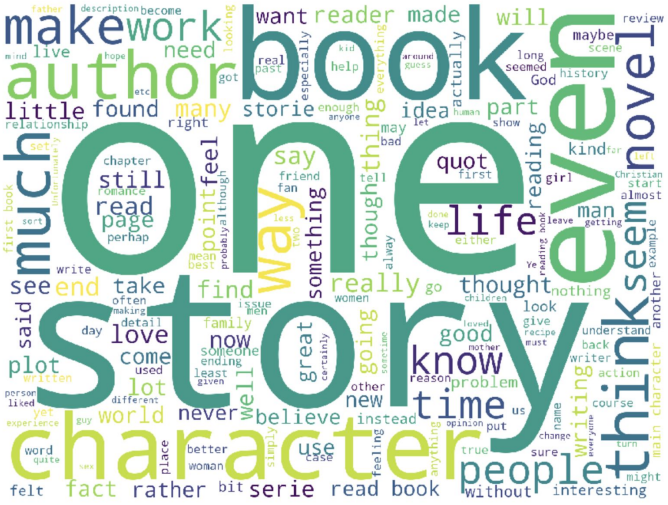


Fig. c) Word Cloud

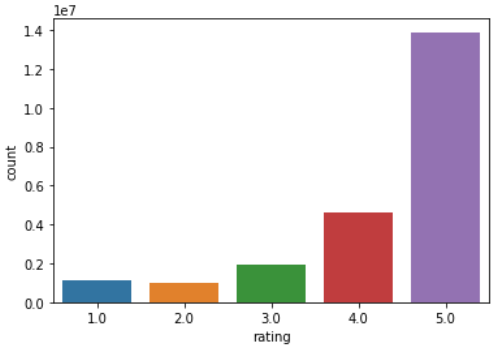


Fig. d) Rating distribution for module 2

Experiments / Proof of concept

Datasets used

	Module 1	**Module 2**
Name	Amazon book reviews	Amazon book ratings
Source	Link	Link
Size of data	10 GB	874 MB
Format	JSON	CSV
Number of instances	8.9M	22M

Data pre-processing

	Module 1	**Module 2**
Techniques used	1) Text cleaning using Regex 2) Lowercasing all words 3) Splitting words on whitespace 4) Removing stopwords 5) Encode vectors to integers using TF-IDF	1) Dropping feature "timestamp" 2) Removing duplicates 3) Removing null values

Methodology

Module - 1

1) Create a New Project on Google Cloud Platform (project name must be unique)
2) Create a Regional Bucket in Google Cloud Storage (bucket name must be unique)
3) Store original dataset available as JSON object (approx. 10 GB) in the Bucket created above
4) Open BigQuery and create a new dataset. Create a new table in this dataset using the JSON file stored in Cloud Storage

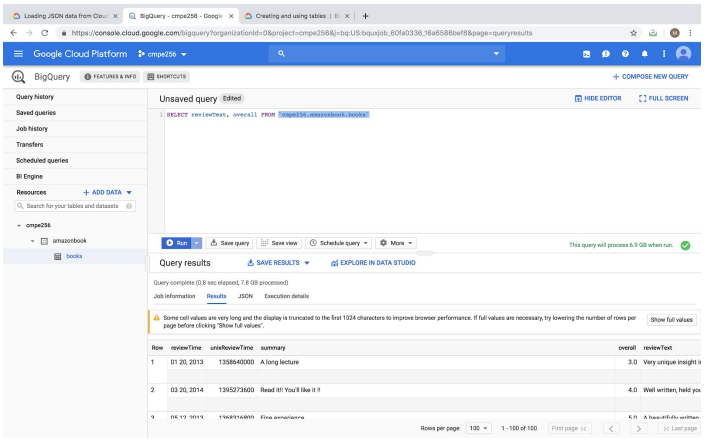


Fig. e) Screenshot of BigQuery and extracted Schema

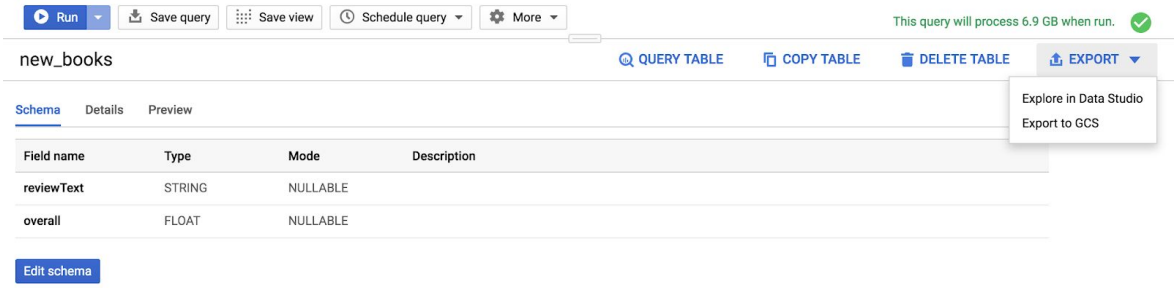


Fig. f) Screenshot showing data types of BigQuery table columns

5) Save the BigQuery table after selecting desired columns using Legacy SQL

6) Using Cloud Shell, create a new Datalab instance using available nearest zone (e.g us-central1-c).

7) In Datalab perform the following:

7.1) Import dataset stored as BigQuery Table into a Pandas dataframe. (Select a sample with ~ 200000 data points)

7.2) Explore and Visualize the data using Pandas, Matplotlib and Seaborn.

7.3) Create a Word Cloud for positive and negative sentiments.

7.4) Perform text processing as mentioned in the section above.

7.5) Split the dataset in 80% - 20% ratio where the latter is the test set.

7.6) Fit and transform the training set into a sequence of numeric vectors using TF-IDF. Transform the test set using the same vectorizer.

7.7) Create the classifier model and fit on the encoded training vectors.

7.8) Predict the sentiments using the encoded test vectors.

7.9) Since classes are imbalanced, check the Classification Report, ROC Curve, AUC Score, Precision Recall Curve

Module - 2

1) Load ratings.csv dataset file into a pandas dataframe.
2) Rename dataframe columns as userId, itemId, ratings and timestamp.
3) Drop timestamp columns as part of Feature engineering.
4) Drop duplicates and missing values from the dataframe(Noticed there were no duplicate as well as missing values in the dataset).
5) Visualize the dataset using seaborn to see the distribution of ratings.
6) Create sample dataframe of size 1000000 from the pandas dataframe.
7) Create a surprise dataset from the sample dataframe.
8) Divide the data into 70/30 ratio using train set and test set.
9) Run 5-fold cross validation and print results.
10) Define svd algorithm using the fine tuned parameters.
11) Train the model on train data on SVD algorithm and predict values for test test using the model.
12) Validate ratings predictions by calculating RMSE.
13) Similarly, train the model for ALS algorithm and calculate RMSE for ALS model.
14) Compare the RMSE values to analyze which model gives best RMSE.
15) Give top 10 recommendations to 5 users using both models.

Graphs

ROC Curve of Logistic Regression Model was better than the remaining models.

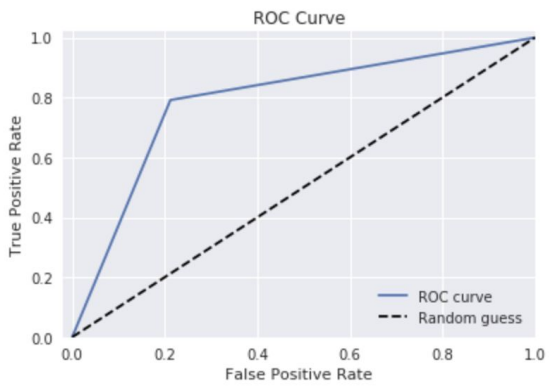


Fig. g) ROC Curve for Logistic Regression

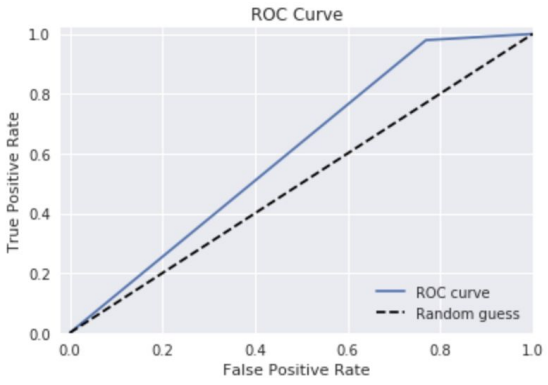


Fig. h) ROC Curve for Random Forests

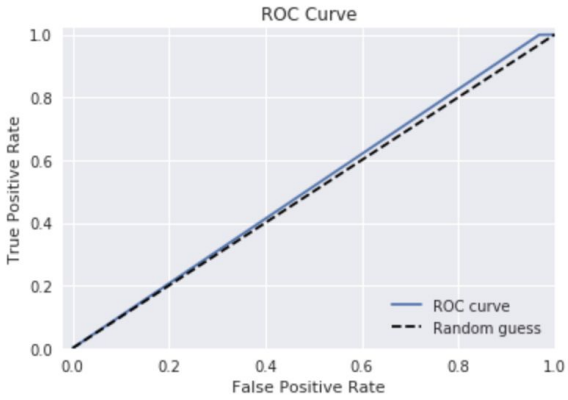


Fig. i) ROC Curve for Multinomial Naive Bayes

Analysis of results

Module - 1

Logistic Regression performed best on the given dataset.

Algorithm	ROC-AUC Score (Sampled data)
Logistic Regression	0.79
Multinomial Naive Bayes	0.52
Random Forest Classifier	0.61

Module - 2

SVD performed better than ALS for sample size of 1 Million but ALS is faster for even large sample size of 10 Million.

Algorithm Used	RMSE(1 Million Sample Data)	Parameter Values
SVD	1.0733	n_factors=132, n_epochs=50, lr_all=0.01, reg_all=0.2, random_state=42, verbose=True
ALS	1.0785	method': 'als', 'n_epochs': 5, 'reg_u': 12, 'reg_i': 5

Discussion and conclusion

Module 1:

1) Size of dataset is very large and required Google Cloud Storage and BigQuery to obtain the BigQuery table from JSON object.
2) Dataset is highly imbalanced, approximately 20% reviews have a negative sentiment and the rest are positive. We used class_weights parameter in the classifiers to handle this issue.
3) Due to inconsistent Google Datalab session, we had to sample the dataset to extract a subset.
4) Data visualization, word cloud, text cleaning, vectorization, model training and evaluation were carried out successfully
5) We had decided to deploy the sentiment analysis model as a REST API using FLASK microframework but things didn't work out well.

Module 2:

1) Due to the large size of dataset, we were unable to run SVD for a sample size of 2 Million.
2) Google colab crashed while cross validation for sample size > 1 Million.
3) We decided to model using Apache Spark but due to very large size of dataset, it did not work out very well.
4) ALS is faster even for large dataset of size 10 Million.
5) For sample size of 1 Million, both ALS and SVD performed well. SVD was slower but RMSE for SVD was slightly better than ALS.

Project plan

Task owner	**Module 1**	**Module 2**
Mrinmayi Gavali	1) Getting the data 2) Loading data to Google Cloud Storage Bucket 3) Creating the dataset and BigQuery Table 4) Instantiating a Datalab VM instance 5) Datalab - Data Fetching, Data Manipulation, Exploration,	1) Fine tuning hyperparameters for the recommender systems 2) Evaluating results

	Visualization, Text Processing, TF-IDF, Encoding, Train-Test Split, Model Selection and Training, Metric Selection, Evaluation and Results	
Priyanka Kumar	1) Explore ways to carry out distributed computing using Apache Spark to perform Sentiment Analysis 2) Data Exploration using Apache Spark 3) Implement one machine learning model for binary classification and convey results	1) Implement a recommender system for Amazon Books using ALS. 2) Data Exploration, Manipulation, Model Selection, Training, Cross-Validation, Metric Selection
FNU Aprajita	1) Implement sentiment analysis on a small sample using NLTK's built-in sentiment analyzer 2) Implement one machine learning model for binary classification and convey results	1) Implement a recommender system for Amazon Books using SVD. 2) Data Exploration, Manipulation, Model Selection, Training, Cross-Validation, Metric Selection