**Project Report**

**On**

**YelpAnalytics: US Based Restaurants-data Analysis & Visualization using Big Data Technologies, Machine Learning and Cloud Computing.**



*Submitted*

*In partial fulfilment*

*For the award of the Degree of*

**PG-Diploma in Big Data Analytics**

**(C-DAC, ACTS (Pune))**

**Guided By: Submitted By:**

Dr. Krishnanjan B.

Charudatta Marne (220940125006)

Jyoti Chaudhary (220940125016)

Paritosh Sangave (220940125030)

Saumyajit Guha (220940125040)

Saurabh Jamdade (220940125049)

**Centre for Development of Advanced Computing(C-DAC), ACTS (Pune- 411008)**

## Acknowledgement

This is to acknowledge our indebtedness to our Project Guide, **Dr. Krishnanjan Bhattacharya,** C-DAC ACTS, Pune for his constant guidance and helpful suggestion for preparing this project **US Based Restaurants-data Analysis & Visualization using Big Data Technologies, Machine Learning and Cloud Computing.** We express our deep gratitude towards him for inspiration, personal involvement, constructive criticism that he provided us along with technical guidance during the course of this project.

We take this opportunity to thank Head of the department **Mr. Gaur Sunder** for providing

such a great infrastructure and environment for our overall development.

We express sincere thanks to **Mrs. Namrata Ailawar**, Process Owner, for their kind cooperation and extendible support towards the completion of our project.

It is our great pleasure in expressing sincere and deep gratitude towards **Mrs. Risha P R (Program Head)** and **Mrs. Srujana Bhamidi** (Course Coordinator, PG-DBDA) for their valuable guidance and constant support throughout this work and help to pursue additional studies.

Also, our warm thanks to **C-DAC ACTS Pune**, which provided us this opportunity to carry out, this prestigious Project and enhance our learning in various technical fields.

Charudatta Marne (220940125009)

Jyoti Chaudhary (220940125016)

Paritosh Sangave (220940125030)

Saumyajit Guha (220940125040)

Sourabh Jamdade (220940125049)

## ABSTRACT

This project utilized various data analysis tools and techniques to extract business insights from the Yelp dataset. The data was initially stored in MongoDB and then preprocessed using Spark. Pandas was used to perform the analysis and visualizations were created in Tableau. Additionally, a machine learning model was built for sentimental analysis and a Recommender System was developed. The findings of the analysis were presented using Streamlit. Overall, this project demonstrates how businesses can utilize data analysis to gain valuable insights from customer reviews and improve their overall customer satisfaction.

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **S. No** | **Title** | **Page No.** |
|  | **Front Page**  **Acknowledgement**  **Abstract**  **Table of Contents** | **I II III**  **IV** |
| **1**  **1.1**  **1.2** | **Introduction**  Introduction  Objective and Specifications | **01-02**  **01**  **02** |
| **2** | **Literature Review** | **03-04** |
| **3** | **Methodology/ Techniques** | **05-13** |
| **3.1** | Approach and Methodology/ Techniques &  Proposed System Architecture | **05** |
| **3.2** | Dataset | **08** |
| **3.3** | Model Description | **09** |
| **4** | **Implementation** | **14-30** |
| **4.1** | Implementation | **14** |
|  |  |  |
| **5** | **Results** | **30-31** |
| **5.1** | Results | **30** |
| **6** | **Conclusion** | **32-33** |
| **6.1** | Conclusion | **32** |
| **7** | **References** | **34-35** |
| **7.1** | References | **34** |

**Chapter 1**

**Introduction**

Detection of text regions either from handwritten or printed document images containing various non-textual information is a difficult task, and it can be more challenging to locate the position of the text regions when we deal with a doctor’s prescription.

**1.1 Introduction**

In a society that is now digitally enhanced, we depend on computers to process huge amounts of data. Companies specifically need to process large volumes of data at a constant, fast pace. This requires specific tools that can deal with these large sets of data and process them efficiently in a parallel fashion. Something similar to this is needed when we are dealing with the Yelp dataset.

The Yelp dataset is a collection of data on local businesses, users, and reviews from Yelp.com. The dataset contains information on businesses such as their name, location, and rating, as well as information on users who have left reviews, such as their name and location. This report will analyze and visualize the Yelp dataset to gain insights into the data, as well as perform sentiment analysis on the reviews.

The dataset contains 908,915 tips by 1,987,897 users. 150k businesses in which we have 52k restaurants. Over 7 million individual reviews, it can be used for NLP or other purposes. Over 1.2 million business attributes like hours, parking, availability, and ambience. Aggregated check-ins over time for each of the 131,930 businesses.

In our project work, we want to store the raw data in MongoDB, and then we warehouse the data using pySpark. This is due the large nature of the data, and the fact that Spark is optimized for parallel computing. We clean the data and preprocess it in spark and then the data is sent back into MongoDB where we can pull it for further in-depth analysis. After that visualizations will be made and we will analyze that further along with specific business insights into the data. The data is processed further and then used for the end goal of the project a.k.a., making a sentiment analyzer and a recommender model utilizing the data. These models are then deployed onto a user interface with streamlit.

The entire goal of the project is business oriented. All the visualizations and even the models being made are done so with specific business purpose in mind.

**1.2 Objective**

* Set up data pipelines to store and process the data
* Warehouse the Yelp dataset for processing and analysis.
* Analysis and extraction of information from data to make pertinent visualizations to draw business insights.
* In depth analysis of individual cuisines among all the restaurants.
* Building and deploying Machine Learning models namely Recommender system and Sentiment analysis.
* Creating dashboards and User Interface using Streamlit. Deploy above mentioned ML models using Streamlit.

**Chapter 2**

**LITERATURE REVIEW**

Yun Xu, Xinhui Wu, Qinxia [**Sentiment Analysis of Yelp ‘s Ratings Based on Text Reviews**] In this research paper, a method has been proposed to do sentiment analysis where they note that the reviews are skewed in the favour of positive reviews. They a case for not equalizing the classes and let the model be biased towards positive reviews, and trying it we found much improved results, atleast in the context of Yelp reviews.

**Beyond the Stars: Improving Rating Predictions using Review Text Content**, Gayatree Ganu et al. [2]. This paper talks about how, most reviews are written in a free-text format, and are therefore difficult for computer systems to understand, analyze, and aggregate. One consequence of this lack of structure is that searching text reviews is often frustrating for users. User experience would be greatly improved if the structure and sentiment conveyed in the content of the reviews were considered. Their work focuses on identifying this information from free-form text reviews, and using the knowledge to improve user experience in accessing reviews. Specifically, we focused on improving recommendation accuracy in a restaurant review scenario.

**Large-Scale Sentiment Analysis for News and Blogs** (System Demonstration), Namrata Godbole et al. [3] In this research work, they demonstrate their large-scale sentiment analysis system for news and blog entities built on top of the Lydia text analysis system. They note how large scale sentiment tracking is very useful specially how it changes over time.

**Sentiment Analysis in Social Media and Its Application: Systematic Literature Review**: Zulfadzli Drus, Haliyana Khalid. [4]. Social media contain a large amount of raw data that has been uploaded by users in the form of text, videos, photos and audio. The data can be converted into valuable information by using sentiment analysis. A systematic review of studies published between 2014 to 2019 was undertaken using the following trusted and credible data.

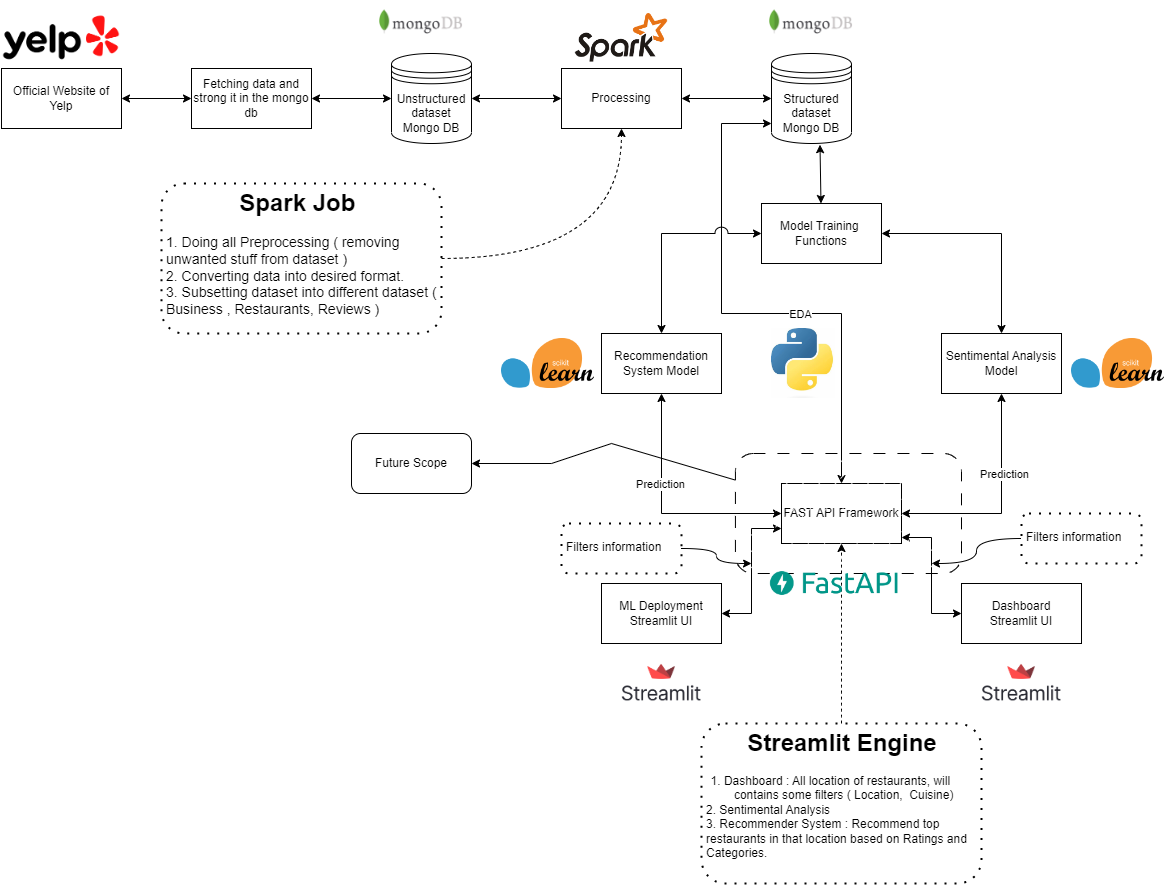
**Collaborative Filtering Recommender Systems** by J. Schafer, Dan Frankowski et al [5] in which they look into collaborative filtering systems brings together the opinions of large interconnected communities on the web, supporting filtering of substantial quantities of data. They also discuss how to evaluate CF systems, and the evolution of rich interaction interfaces

**A survey of active learning in collaborative filtering recommender systems** by Mehdi Elahi, Francesco Ricci et all where they present a comprehensive overview of the evaluation methods and metrics that have been employed by the research community in order to test active learning strategies for collaborative filtering. Finally, they compare the surveyed strategies and provide guidelines for their usage in recommender systems.

**Chapter 3**

**Methodology and Techniques**

**Proposed System Architecture**

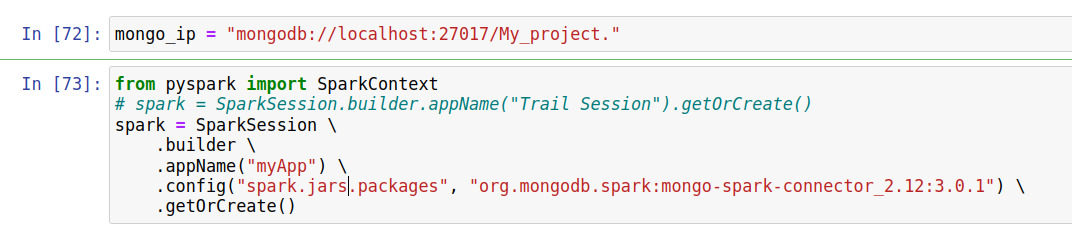


**3.1 Methodology:**

**3.1.1 Pipelines**

We set up data pipeline that involves loading data from MongoDB into Spark, processing and extracting the data in Spark, splitting the data into chunks required for different use cases, and exporting the data back into MongoDB.

The first step in this pipeline is to load the data from MongoDB into Spark. This can be done using the MongoDB Spark Connector, which allows Spark to read data directly from MongoDB collections. This was a major challenge since very little information is available on how, and a lot of it is old. We had to manually go through the errors and fix them by downloading specific dependency JAR files from Maven Central. Once the data is loaded into Spark, it can be processed and cleaned to prepare it for analysis.



The next step is to split the data into chunks required for different use cases. This involved partitioning the data by location, content or other relevant criteria. Once the data is partitioned, it can be exported back into MongoDB using the MongoDB Spark Connector and written into a new collection.

Before exporting the data, it is important to pre-process and warehouse the data in Spark. This involved performing data cleaning and transformation, aggregating data for analysis, joining different data sets. Once the data is pre-processed and warehoused in Spark, it can be exported back into MongoDB for further analysis or use.

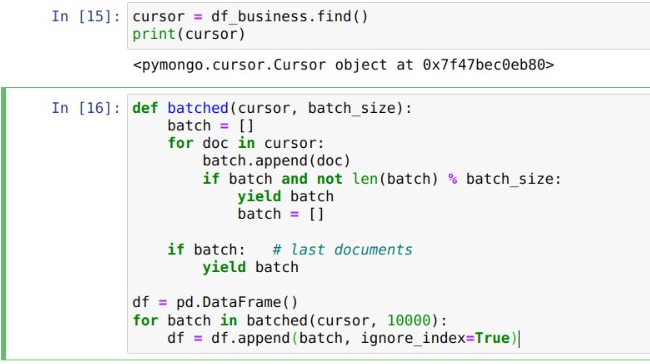
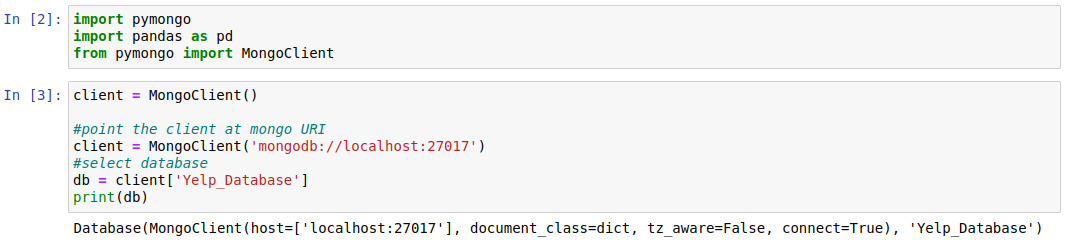
Overall, this data pipeline allows organizations to extract insights from large datasets stored in MongoDB by leveraging the power and scalability of Spark. By splitting the data into chunks and pre-processing it in Spark, organizations can optimize their analysis for different use cases and gain a more comprehensive understanding of their data.

A second pipeline is built, Once the data is processed and cleaned through PyMongo, which provides a Python interface to MongoDB.

The next step is to use Pandas to load the data from MongoDB and perform further data manipulation and analysis. Pandas provides a powerful set of tools for working with structured data, including filtering, grouping, and aggregating data.

After the data is processed and analysed in Pandas, Python libraries such as Seaborn and Matplotlib can be used to create various visualizations to gain insights into the data. Seaborn provides a high-level interface for creating statistical graphics, while Matplotlib provides more fine-grained control over the appearance and layout of the visualizations.

Overall, this data pipeline provides a powerful way to extract insights from processed data stored in MongoDB, using Pandas as the primary tool for data manipulation and analysis, and Python libraries such as Seaborn and Matplotlib for visualization. By using this pipeline, businesses can gain valuable insights into their data, such as trends in customer behavior, popular products, or areas for improvement.



**3.1.2 Spark**

Apache Spark is an open-source distributed computing system designed for processing large amounts of data. It was first developed at the University of California, Berkeley in 2009, and is now maintained by the Apache Software Foundation.

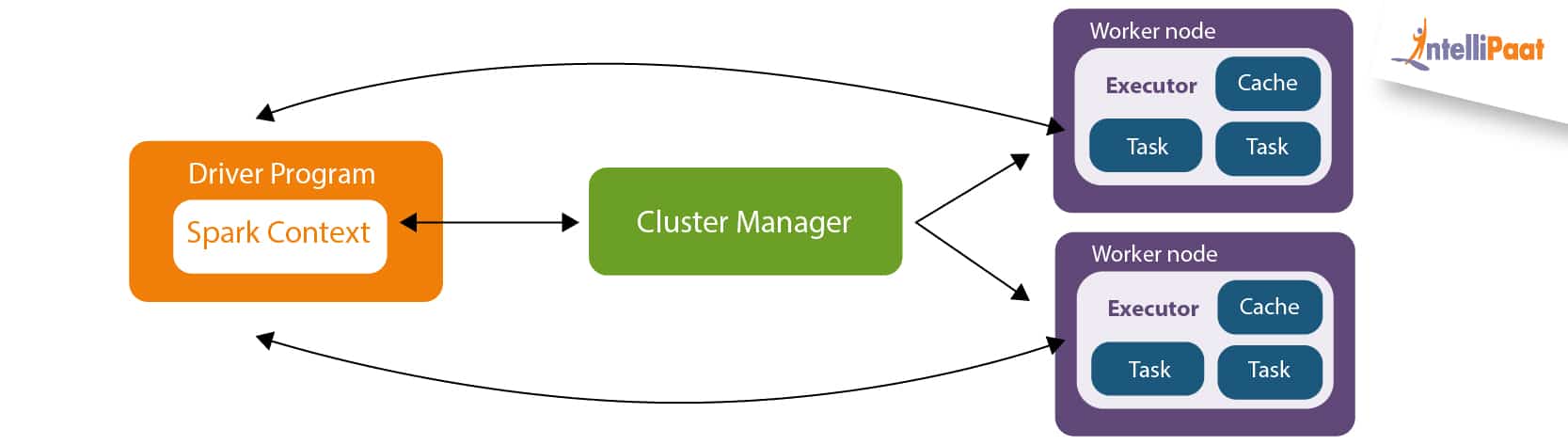
Spark provides a unified computing platform for big data processing, analytics, and machine learning, with support for multiple programming languages, including Java, Scala, Python, and R. It is built on top of the Hadoop Distributed File System (HDFS) and is designed to run in a cluster environment, making it highly scalable and capable of handling massive amounts of data.

One of the key features of Spark is its in-memory processing capability, which allows it to store and process data in memory, rather than on disk, resulting in much faster data processing speeds. It also provides a wide range of built-in libraries and APIs for various use cases, including Spark SQL for SQL-like querying, Spark Streaming for real-time data processing, and MLlib for machine learning.

Spark also supports a wide range of data sources, including HDFS, Cassandra, MongoDB, and more, making it easy to work with a variety of data formats and sources. Additionally, Spark provides a rich set of tools for data visualization and debugging, making it easier to analyze and understand large datasets.

Overall, Apache Spark is a powerful and versatile tool for big data processing, providing organizations with a scalable and efficient way to analyze and derive insights from large and complex datasets.

**Fig 2. Spark Framework**



**3.1.3 MongoDB**

MongoDB is a popular document-oriented NoSQL database that is designed for high scalability, performance, and flexibility. Unlike traditional relational databases, MongoDB does not store data in tables with fixed columns and rows. Instead, it stores data in flexible documents with dynamic schemas, which can be nested and easily expanded as needed.

One of the key features of MongoDB is its ability to handle large volumes of unstructured and semi-structured data, such as social media posts, sensor data, and log files. It also provides powerful tools for data analysis and manipulation, including aggregation, indexing, and map-reduce operations.

MongoDB is highly scalable, and can be easily distributed across multiple servers to handle large-scale deployments. It also provides a range of security features, including authentication, authorization, and encryption, to ensure the privacy and security of sensitive data.

MongoDB is widely used in a variety of applications, including e-commerce, social media, and IoT. Its flexible schema design and high scalability make it a popular choice for applications with rapidly changing data requirements or high volumes of data.

**3.2 Dataset**

The Yelp dataset is a large collection of data that contains information on local businesses, user reviews, and ratings. The dataset was created by Yelp, a popular online platform that allows users to rate and review businesses in their area, and is made available for research purposes.

The Yelp dataset contains over 7 million reviews and 150,000 businesses from 10 metropolitan areas in North America, including Las Vegas, Philadelphia etc. It also includes data on over 6 million users, including their profile information and review history.

The reviews in the dataset include information on the business being reviewed, the user who wrote the review, and the rating given to the business. The dataset also includes information on the date the review was written and the text of the review itself.

In addition to the reviews, the Yelp dataset also includes data on the businesses themselves, such as their name, location, hours of operation, and categories. This information can be used to analyse trends in the types of businesses that are popular in different areas, as well as to identify businesses that are performing well or struggling.

Overall, the Yelp dataset provides a rich source of data for research on user behaviour, sentiment analysis, and machine learning. The large size of the dataset and the variety of data it contains make it a valuable resource for researchers and data scientists working in a wide range of fields.

**3.3 Model Description**

**Preprocessing**

First, the data was loaded into a Spark environment for processing. This is because the data was very large and it was not possible to do this processing without specialized tools like spark which is specialized for parallel and large-scale data processing.

The data contained several nested JSON attributes, from which relevant information was extracted using Spark SQL. Columns that were un-necessary for certain use cases were dropped and re-added depending on changing requirements of the project.

Next, the data was filtered to only include restaurants. The cuisine information was extracted from the category column, which was a string of categories. The data was further filtered to only include restaurants in Florida, and this was separated into a separate dataset for more detailed analysis.

The restaurants data was joined with their specific review data, in order to perform more in-depth analysis. The review text column was separated for sentiment analysis, which could be used to understand customer opinions and preferences.

Finally, the data was filtered to only include restaurants in Tampa, as this was the focus of the recommendation system. The recommendation system was designed to provide personalized recommendations to users based on their preferences and behaviour, and the data was processed and analysed using various techniques such as collaborative filtering and content-based filtering. Overall, this data analysis and recommendation system pipeline provided valuable insights into customer behaviour and preferences, and could be used to improve the user experience and drive business growth.

.

**Xtreme Gradient Boosting (XGBoost)**

**XGBoost** (Extreme Gradient Boosting) is a popular open-source machine learning library that is designed for creating and optimizing gradient boosting models. Gradient boosting is a powerful technique for creating high-performing models by combining many weak classifiers or predictors into a single, strong model.

XGBoost has several features that make it a popular choice for machine learning tasks, including its scalability, speed, and accuracy. It can handle both classification and regression problems and has support for a variety of loss functions and regularization techniques.

One of the key features of XGBoost is its ability to handle large and complex datasets. It has built-in support for parallel processing, enabling it to efficiently handle large datasets with millions or billions of rows and columns. Additionally, it can handle missing values, which are common in real-world datasets.

XGBoost also provides a range of advanced features and optimization techniques, such as tree pruning, early stopping, and custom objective functions, which enable it to create highly accurate and robust models.

XGBoost includes a technique called "regularization," which helps to prevent overfitting and improve the generalization of the model. Regularization involves adding a penalty term to the objective function that the model is trying to optimize, which encourages the model to choose simpler or more robust solutions that are less likely to overfit the training data.

Secondly, XGBoost includes a range of optimization techniques that improve the efficiency and speed of the training process. For example, it uses a technique called "approximate greedy algorithm" to find the best split points in each tree, which reduces the computational complexity and speeds up the training process.

Another advantage of XGBoost is its ability to handle missing data. It can automatically learn how to best handle missing data during the training process, without the need for explicit imputation or pre-processing.

Finally, XGBoost provides advanced features such as parallel processing, early stopping, and custom objective functions, which is specifically useful for us since we have a large dataset to train our sentiment model on

.

**Content Based Filtering Recommender System**

In this approach, user reviews and restaurant categories are analyzed to extract relevant features or attributes. The features can include keywords, topics, or sentiments from the reviews, and categories of the restaurants. The features are then used to create feature vectors for each user and each restaurant.

To make recommendations, the correlation distance between the feature vectors of each user and restaurant is calculated. Restaurants that have a high correlation distance to the user's feature vector are considered to be a good match and are recommended to the user.

This approach has the advantage of being able to capture the specific preferences of users based on their previous reviews, and the categories of restaurants they prefer. However, it may not be effective in recommending new or diverse items, and may suffer from the "cold-start" problem for new users or restaurants with few reviews.

For us this is not a problem since we are not accepting any new entries into the data but in-fact we are using already data present in the data to make recommendations.

.

**Chapter 4**

**Implementation**

1. Use of Python Platform for writing the code with **Pandas, Matplotlib, Seaborn, Tableau** for analysis and visualization**, XGBoost, SciPy and SkLearn** forSentiment Analysis and Recommender Systems
2. Hardware and Software Configuration:

Hardware Configuration:

* + CPU: 16 GB RAM, Quad core processor

Software Required:

**Anaconda**:

It is a package management software with free and open-source distribution of the Python and R programming language for scientific computations (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify deployment.

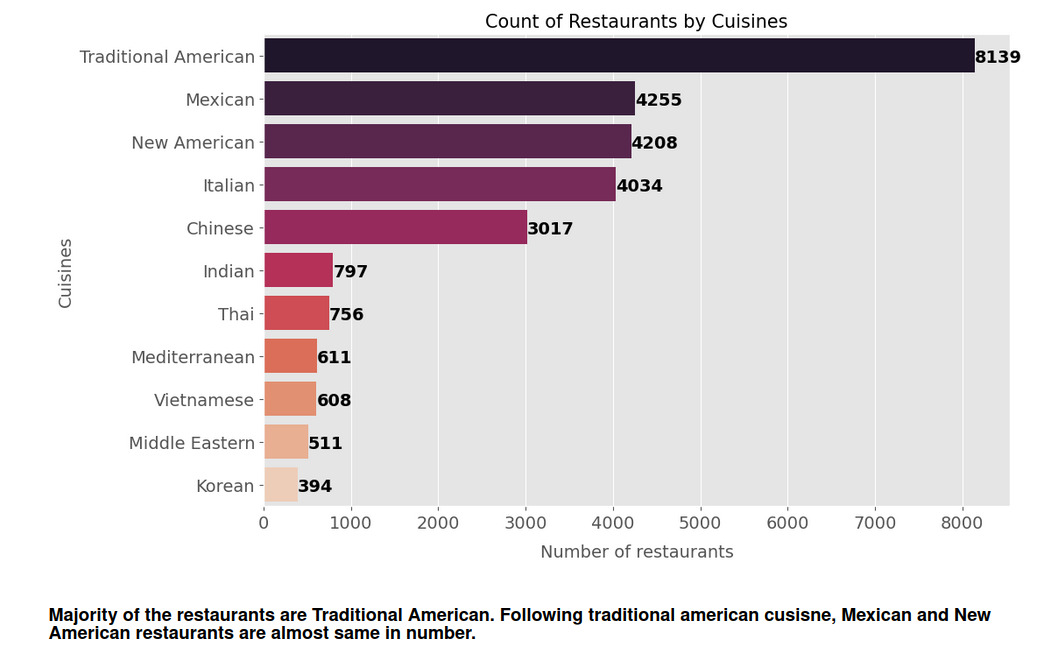
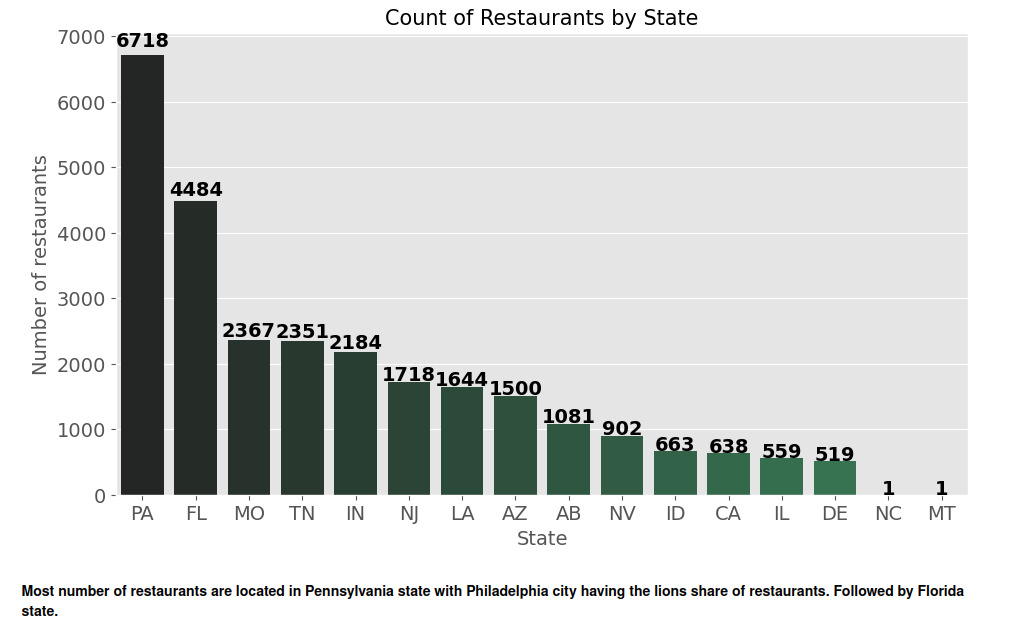
**Jupyter Notebook**:

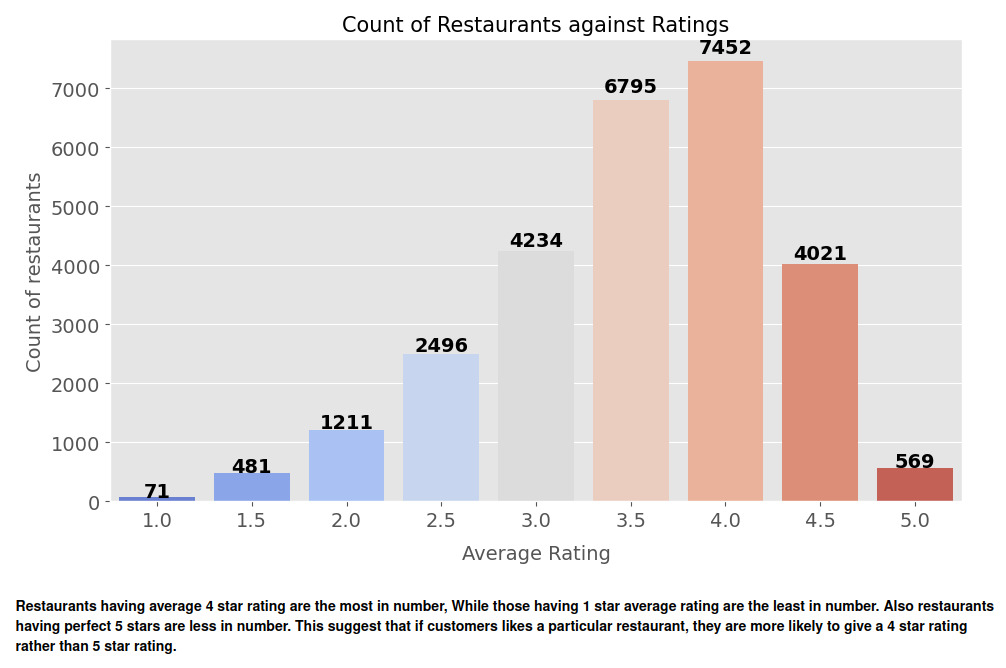
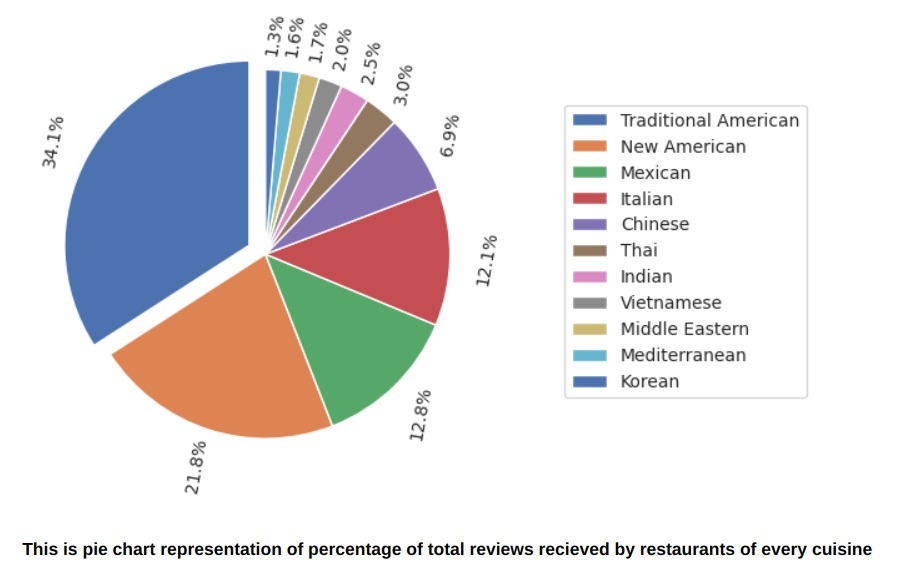
Jupyter is a web-based interactive development environment for Jupyter notebooks, code, and data.

Jupyter is flexible: configure and arrange the user interface to support a wide range of workflows in data science, scientific computing, and machine learning.

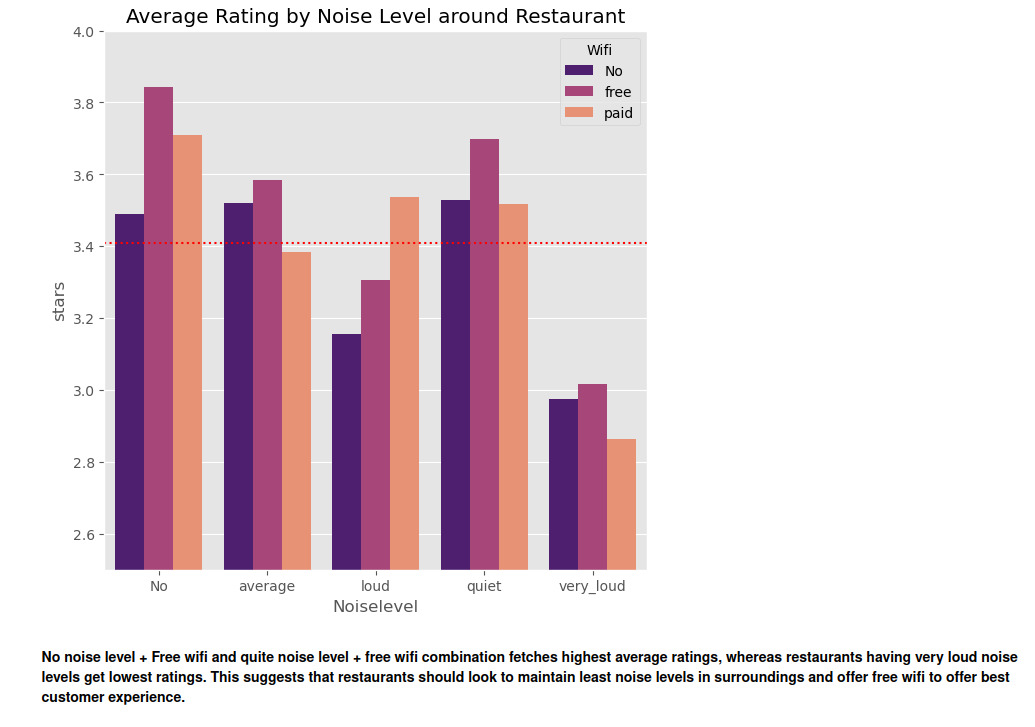
Jupyter is extensible and modular: write plugins that add new components and integrate with existing ones.

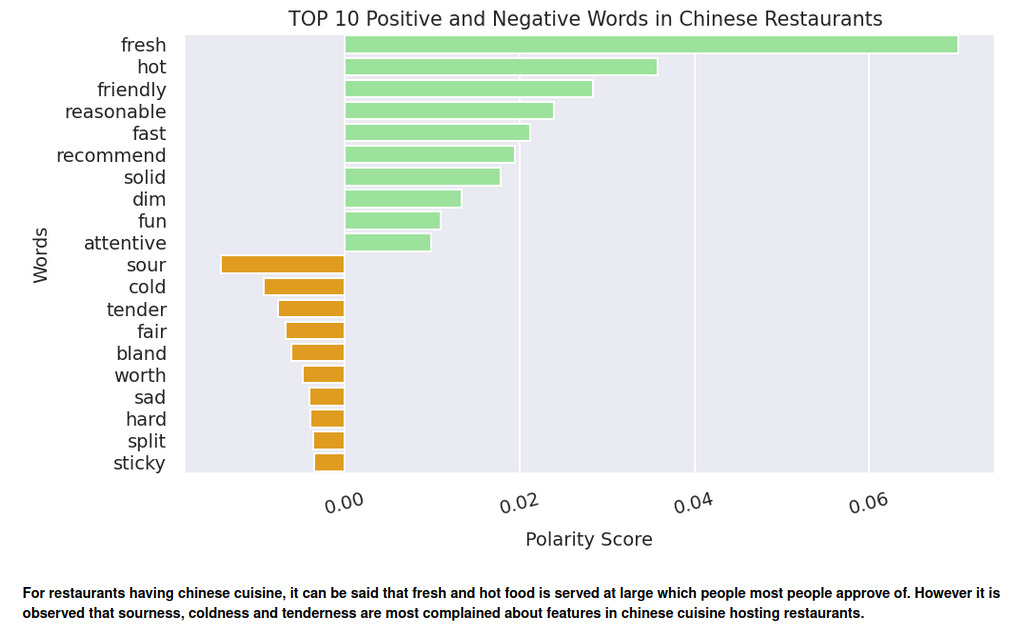
**Some Excerpts of Analysis and Visualizations**

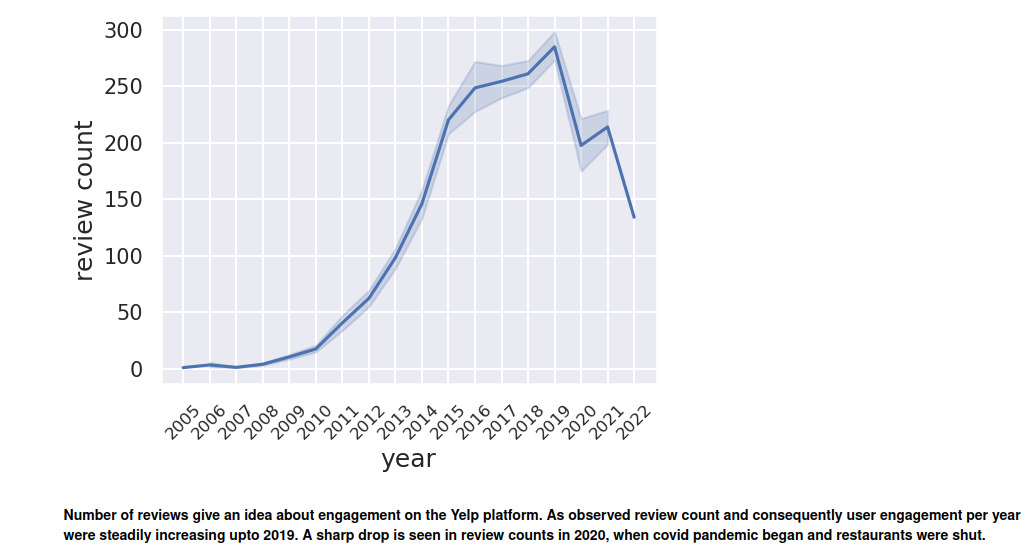


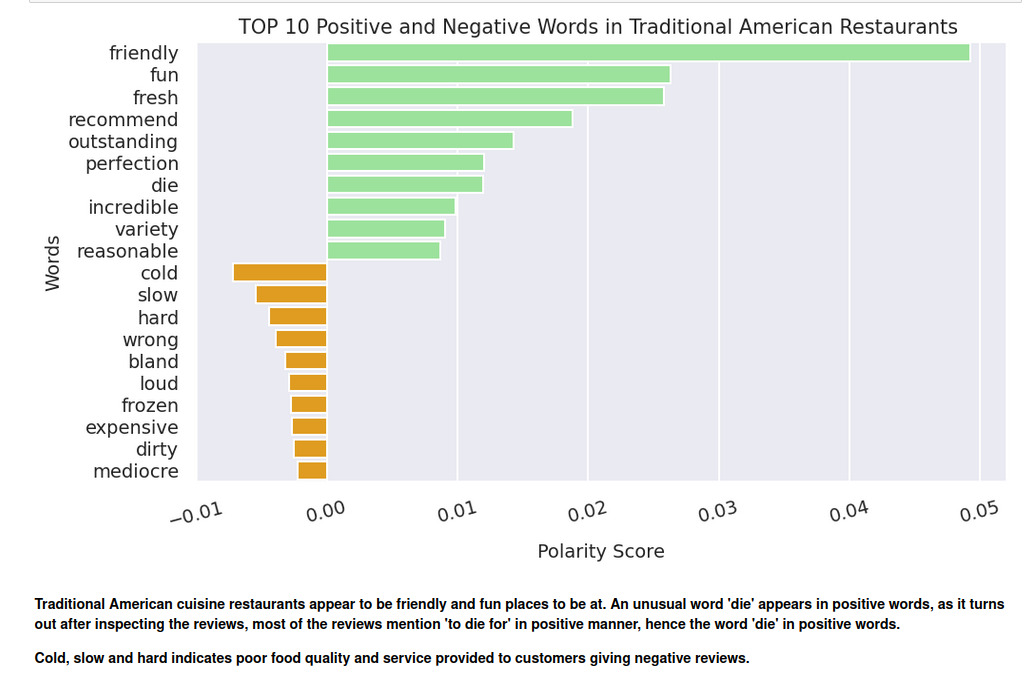
 

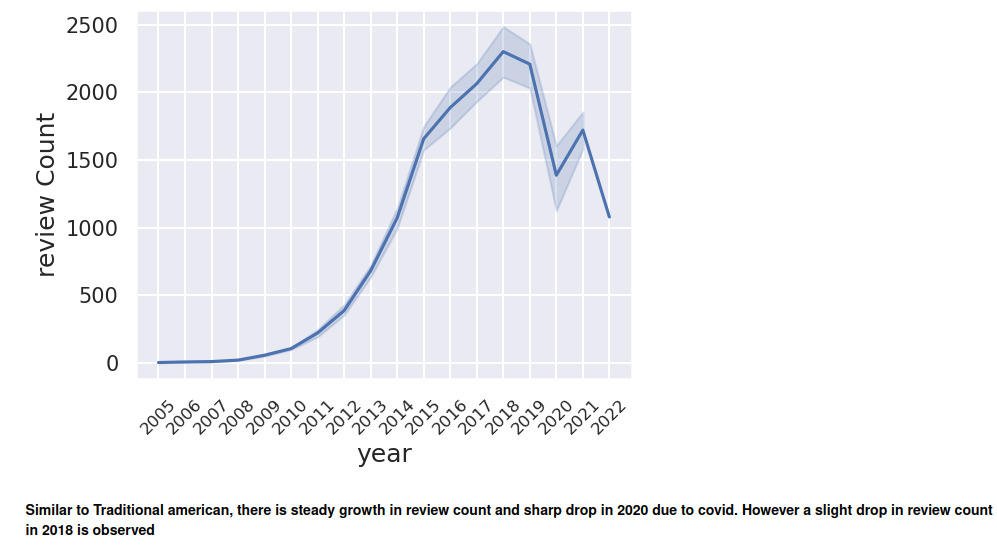






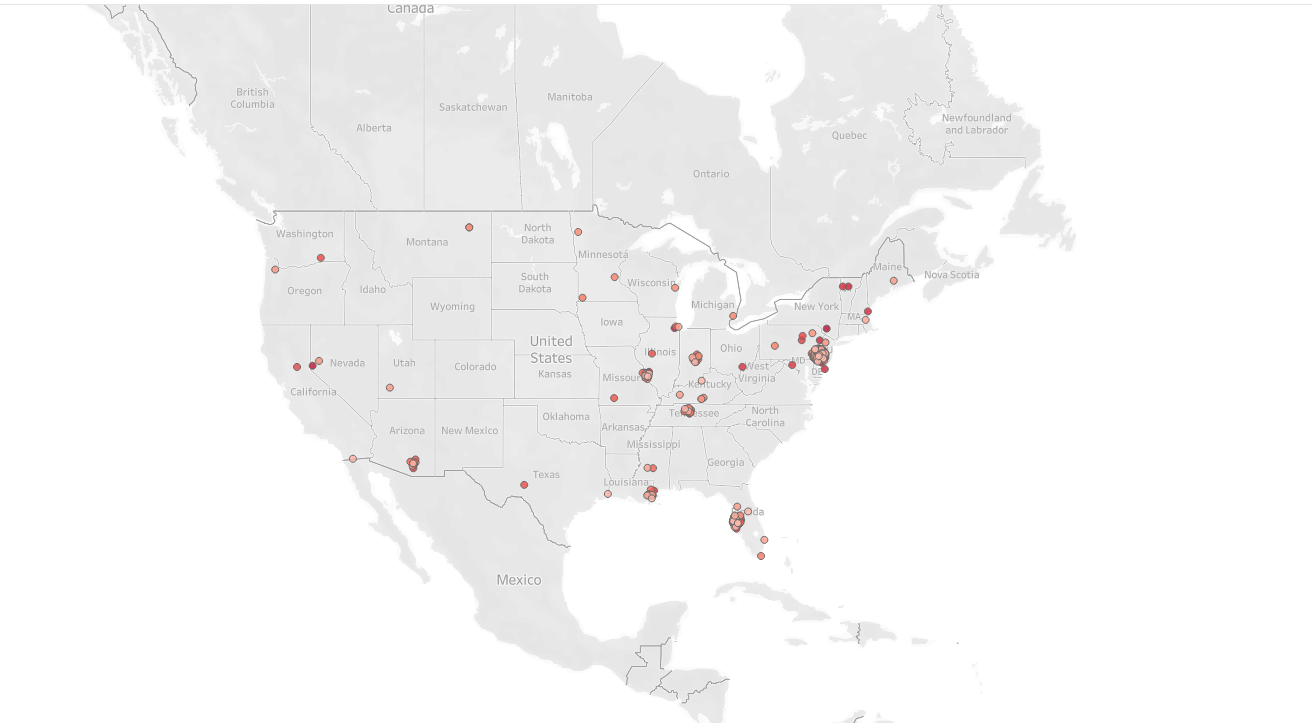


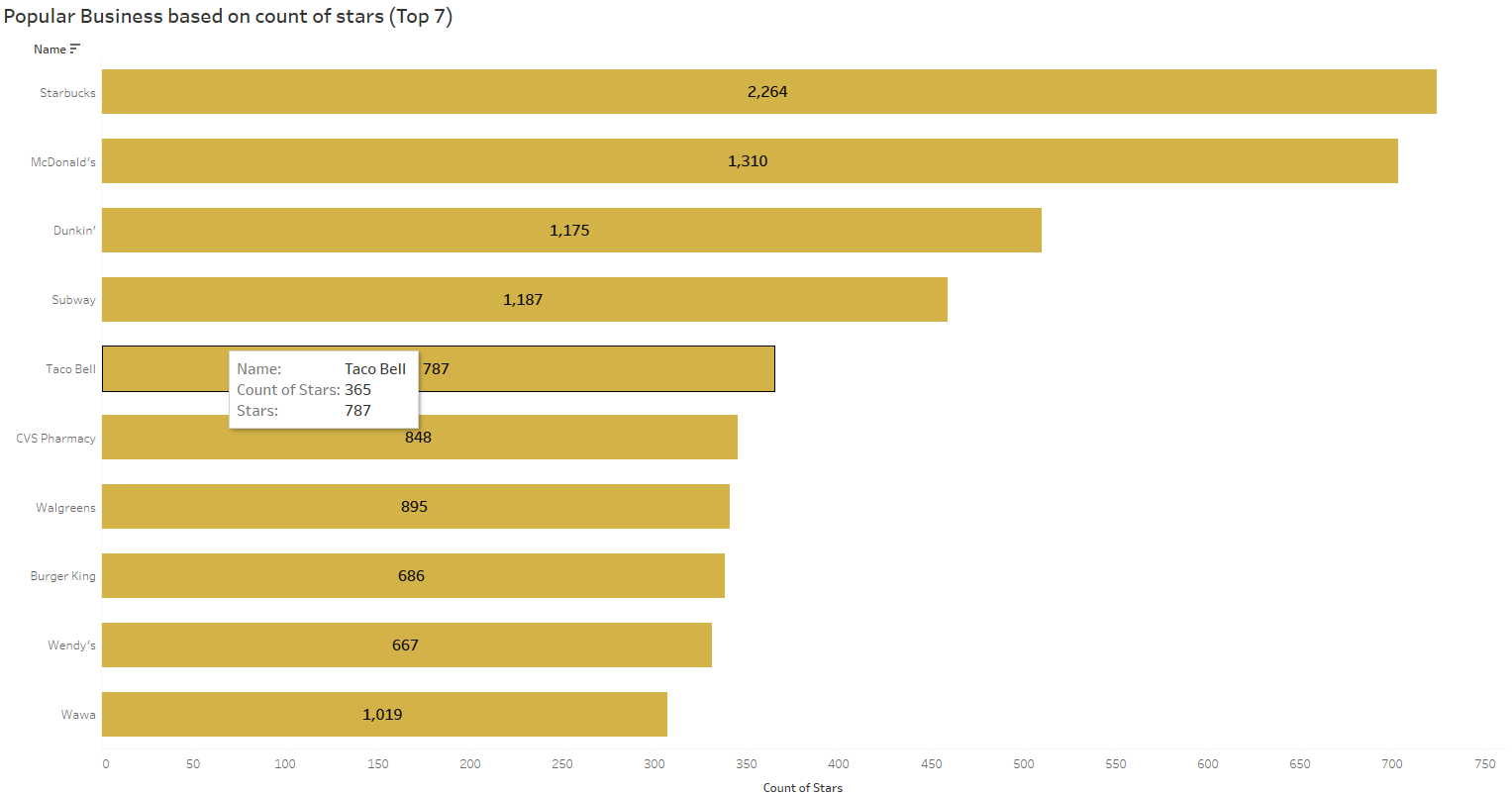


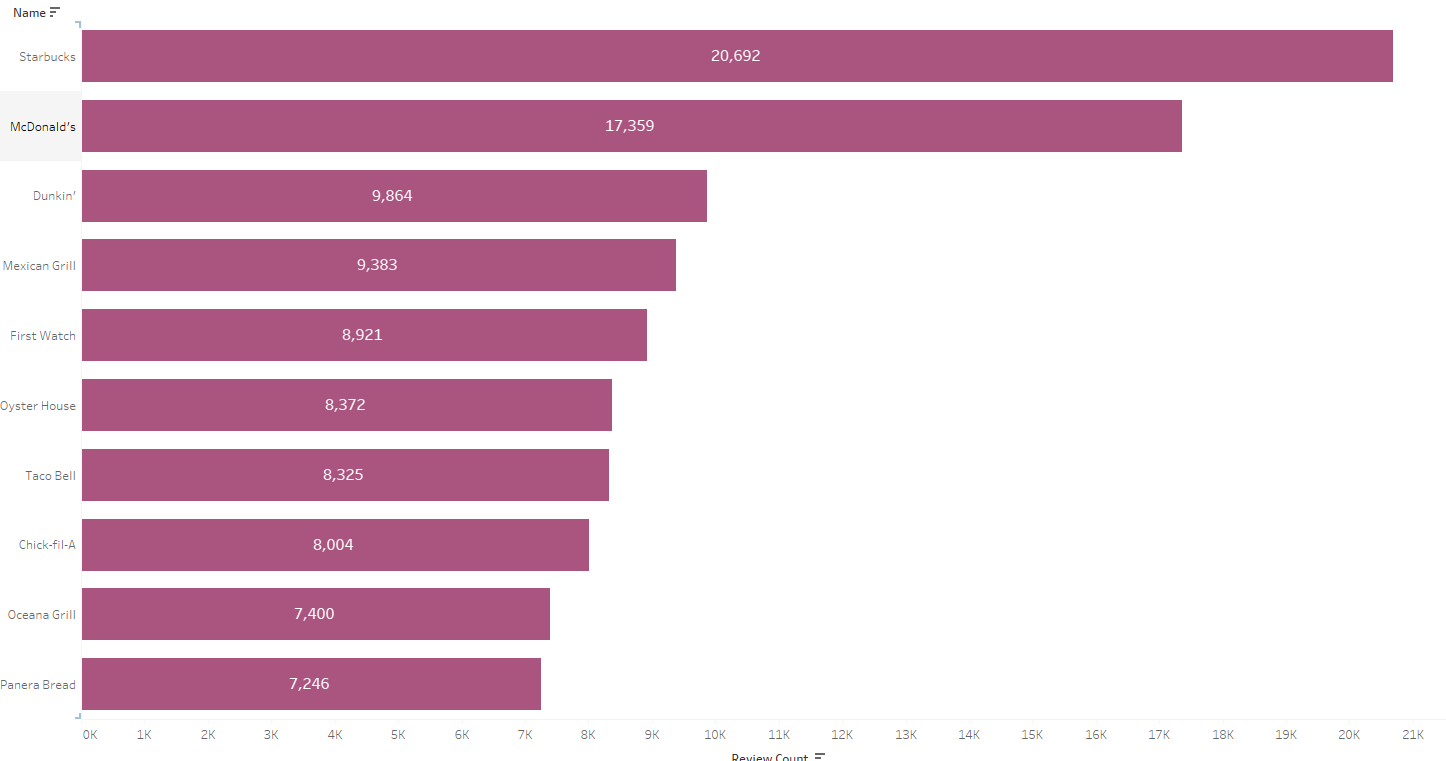


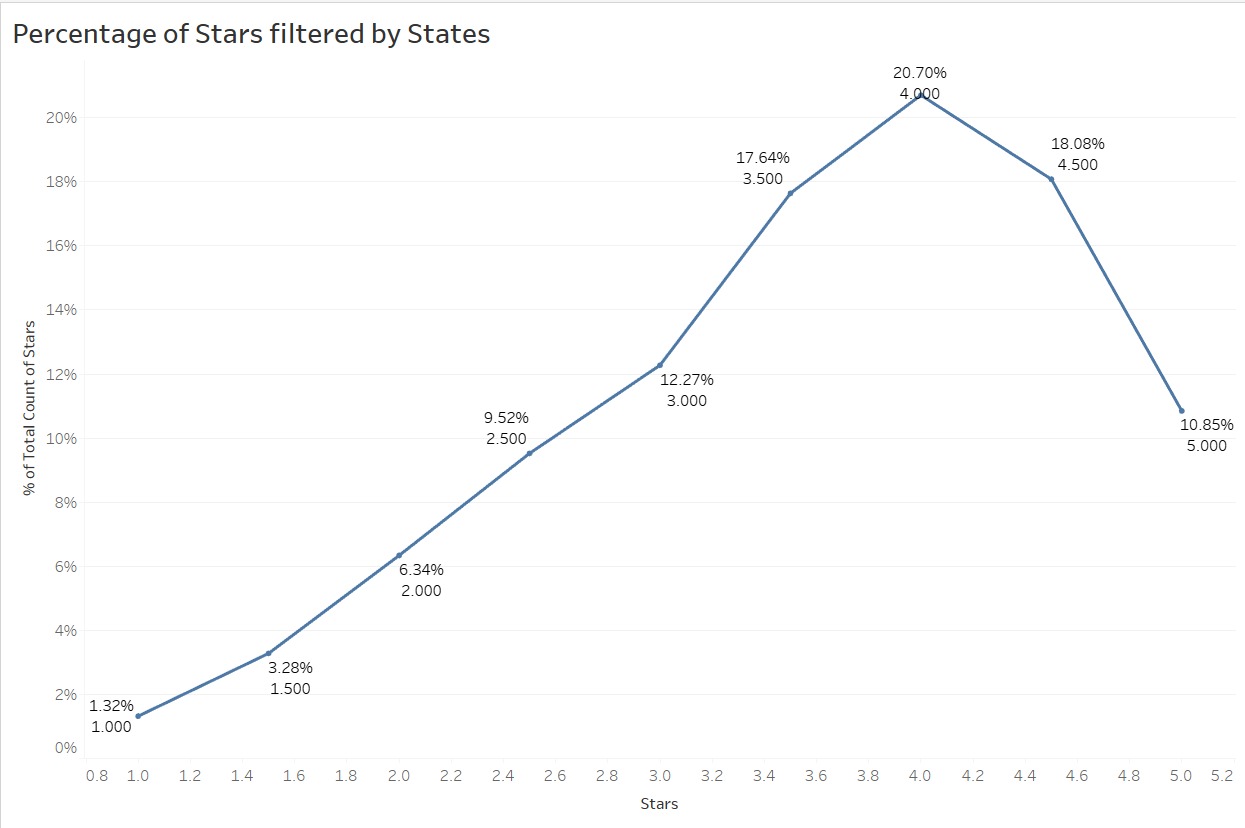
**Tableau Visualizations**

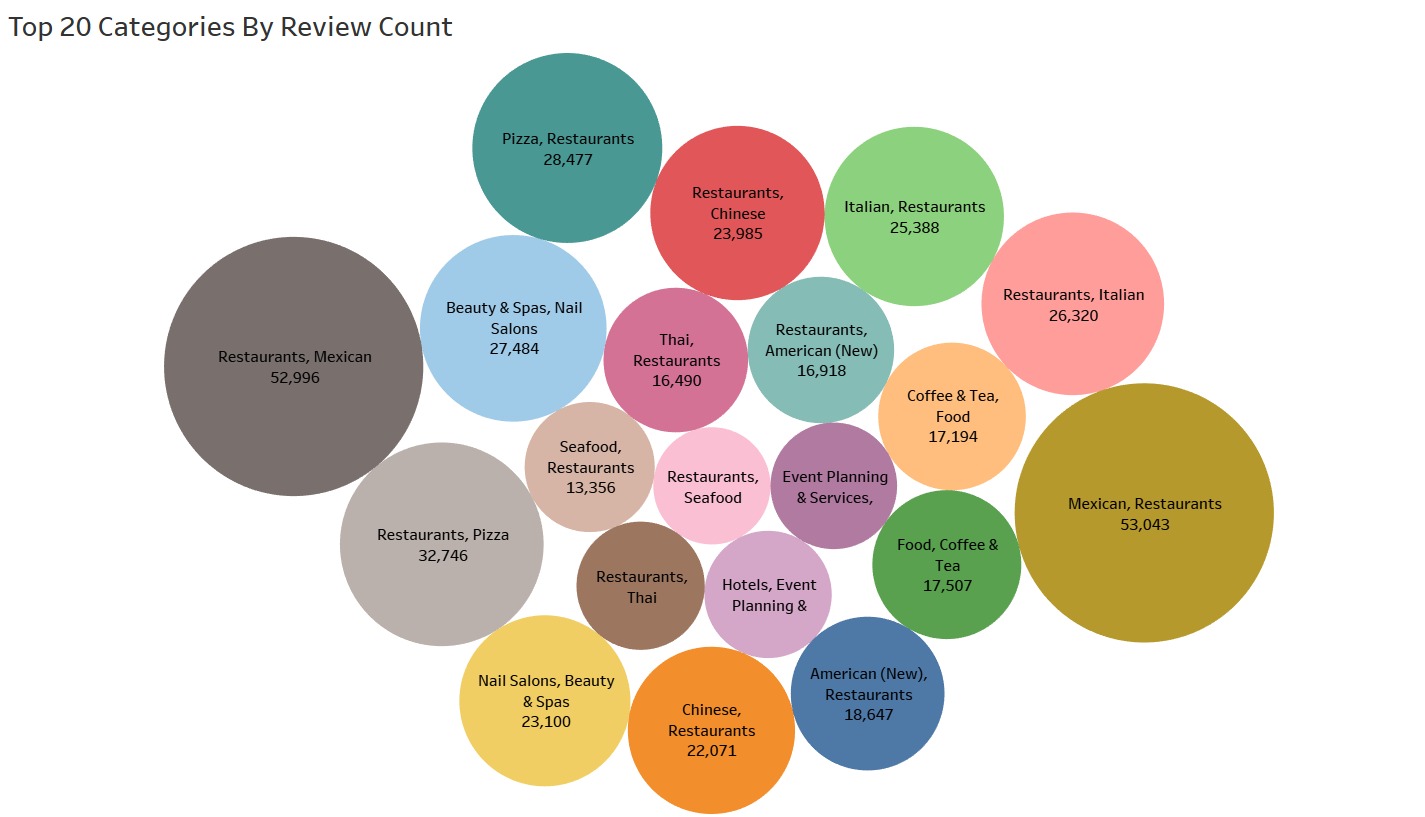
A number of visualizations was also done in Tableau to enable professional level dashboarding and Business Intelligence Tasks:



****

****





**Sentiment Analysis**

We are trying to get sentiment information from textual reviews using XGBoost, a machine learning algorithm for classification and regression problems.

In this approach, the Yelp reviews dataset is pre-processed to remove stop words, punctuations, and to lower-case the text. The reviews are then transformed into a numerical format, such as word counts or TF-IDF vectors, which are used as input features for the XGBoost algorithm.

The XGBoost algorithm is then trained on a labelled dataset, where the reviews are labelled as positive or negative based on the rating given along with the reviews.

* + Greater than or equal to 3.5 => Positive Review
  + Less than or equal to 2.5 => Negative Review

The trained model can then be used to predict the sentiment of new reviews.

This approach has the advantage of being able to handle large amounts of textual data and achieve high accuracy in sentiment classification. However, it may require a significant amount of labelled data for training the model and may not perform well on reviews with less length than the average yelp review and is skewed towards positive data since people in general tend to give positive reviews than negative unless they have very bad experiences.

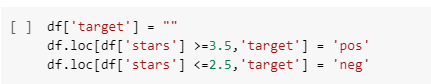
* Random Sampling of 100k rows for training the model 10k more random samples from reviews dataset for testing the model
* Tried Different Models and Strategies: XGBOOST, LightGBM, Random Forest, Naïve Based, Logistic Regression
* BEST MODEL: XGBOOST

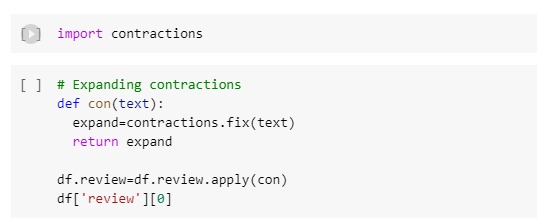
**Word Clouds**

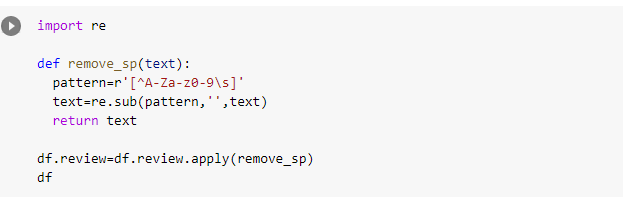
****

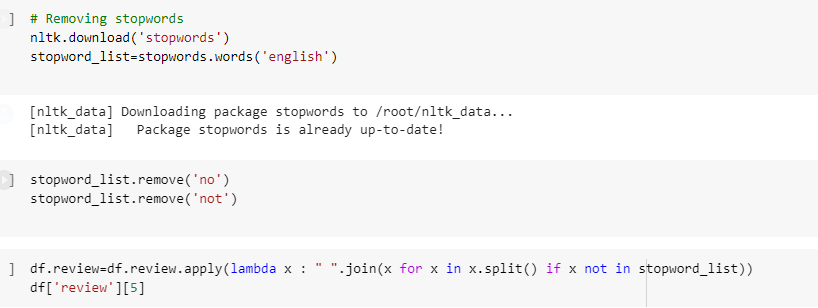
****

**Preprocessing**

****

****

****

****

****

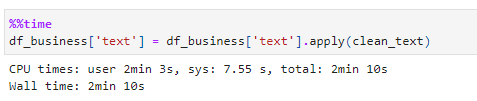
**Model**

****

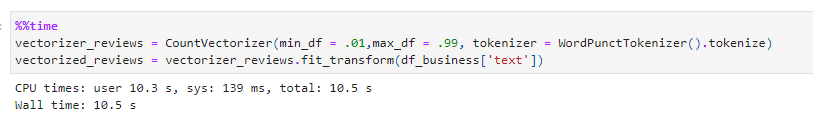
**Recommender**

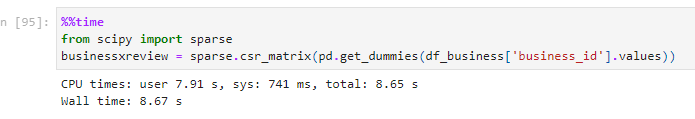
For the recommender system we have gone with a simpler approach where we use the categories and the reviews provided for the restaurant to build a sort of **Content Based Filtering** model to provide the user recommendations for other similar restaurants.

We process the “review” text to remove and filter out any sort of garbage or unwanted characters.

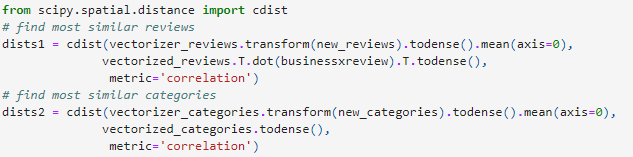


We then use **sklearn CountVectorizer** to create a Bag of Words for both the reviews and the categories

We create a sparse matrix of business\_ids of all reviews vs unique business\_ids by dummying it using get\_dummies to get a numerical representation of the business\_ids



Select a restaurant and calculate the correlation distances of the categories and reviews of that restaurant vs all of the others.

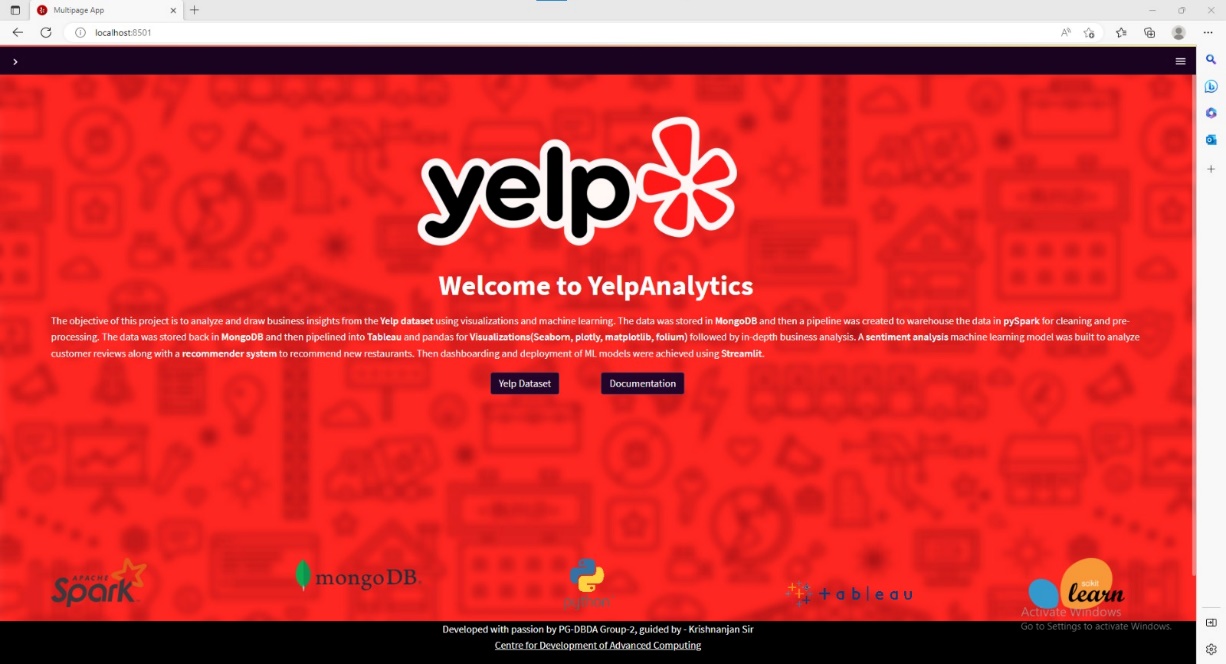


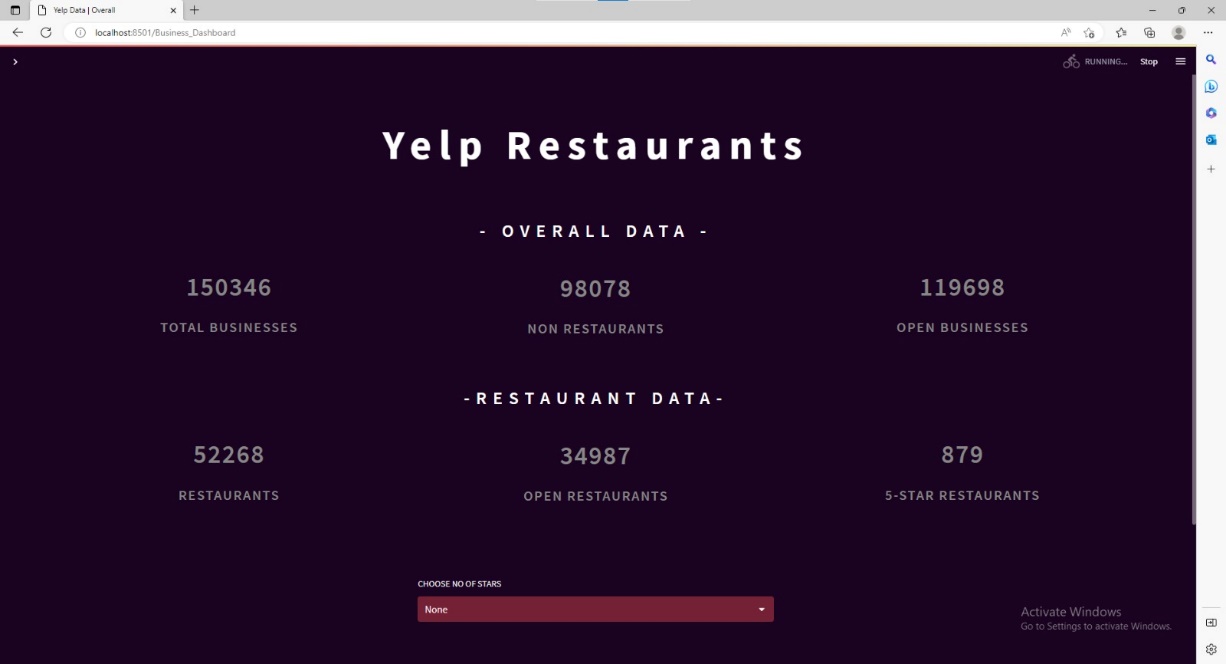
Combine the vectors vertically and find the mean, the arguments of which when sorted should give us the closest matches

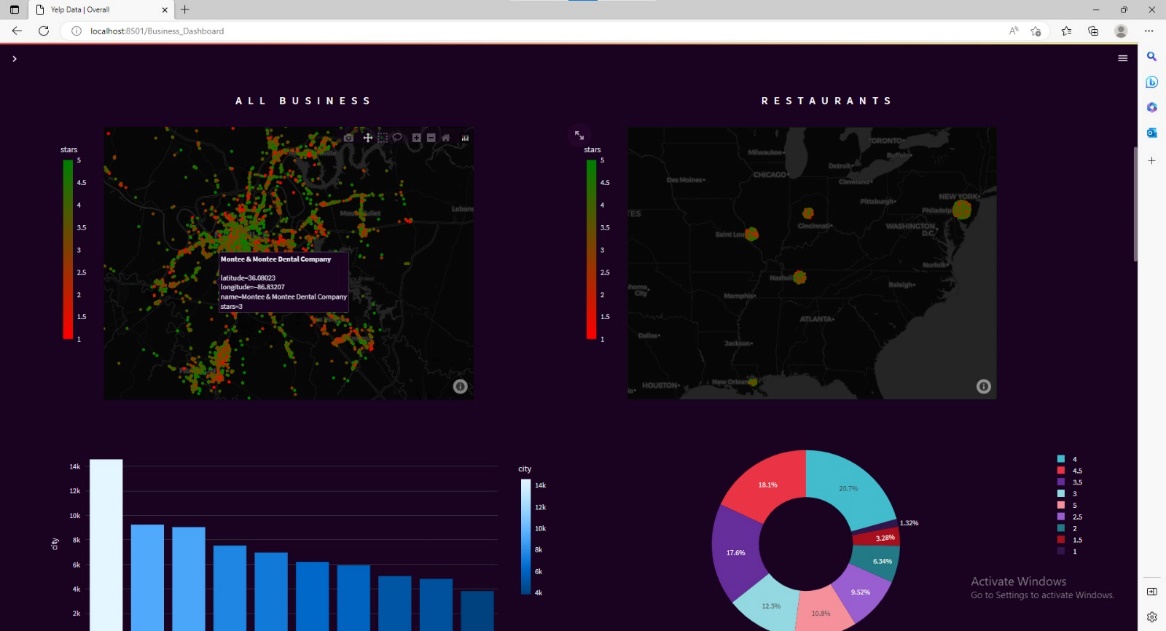


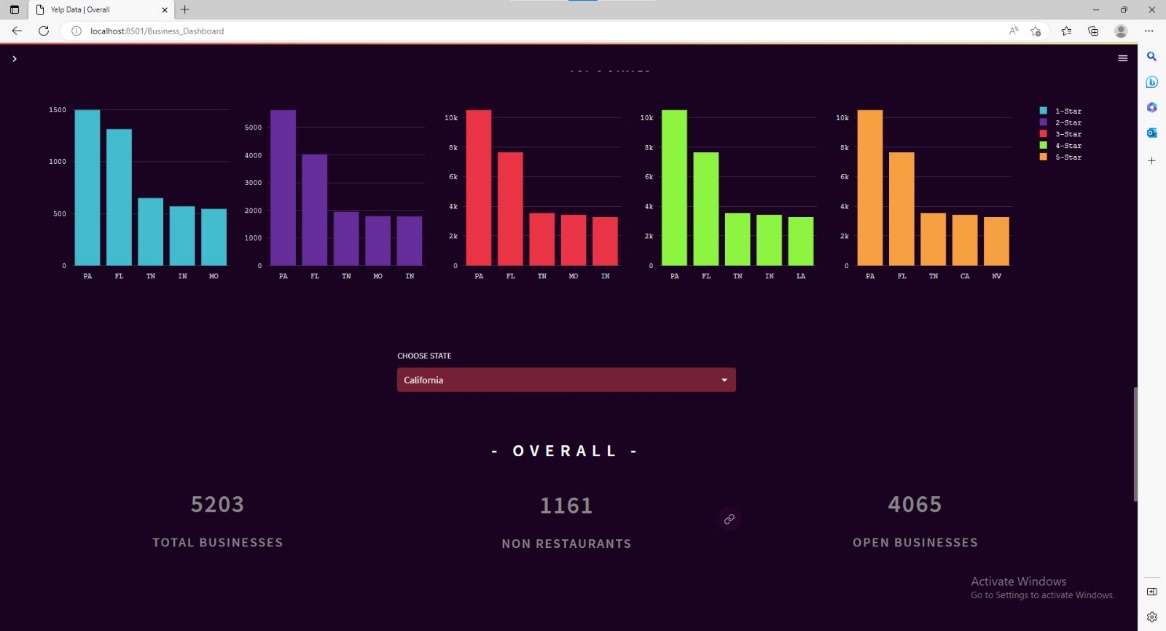


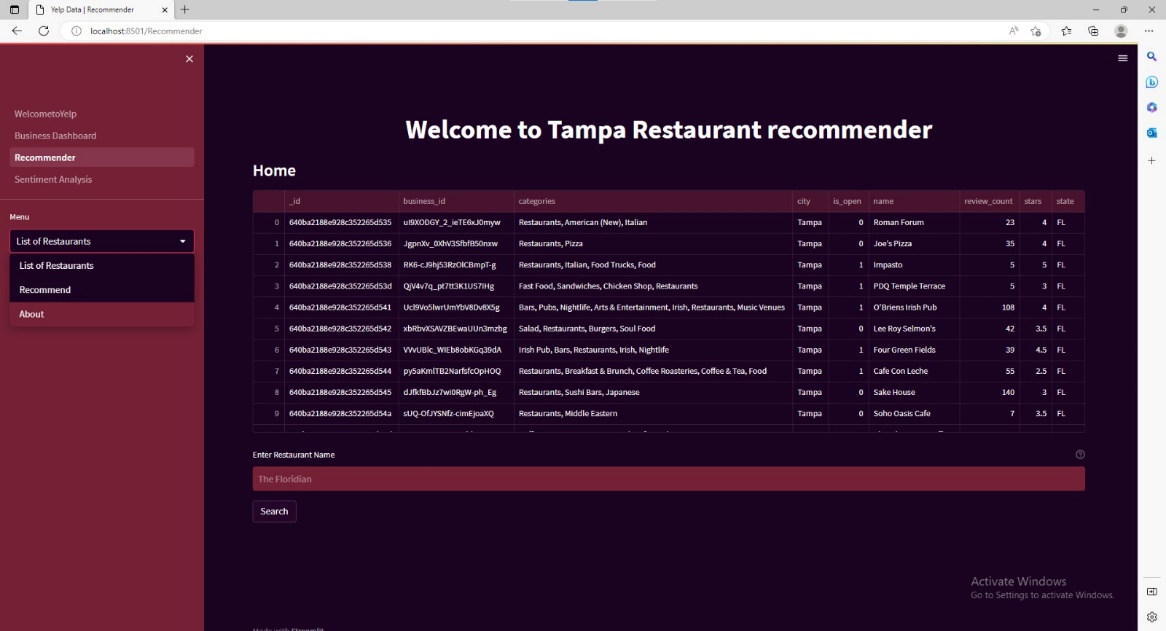
**User Interface**

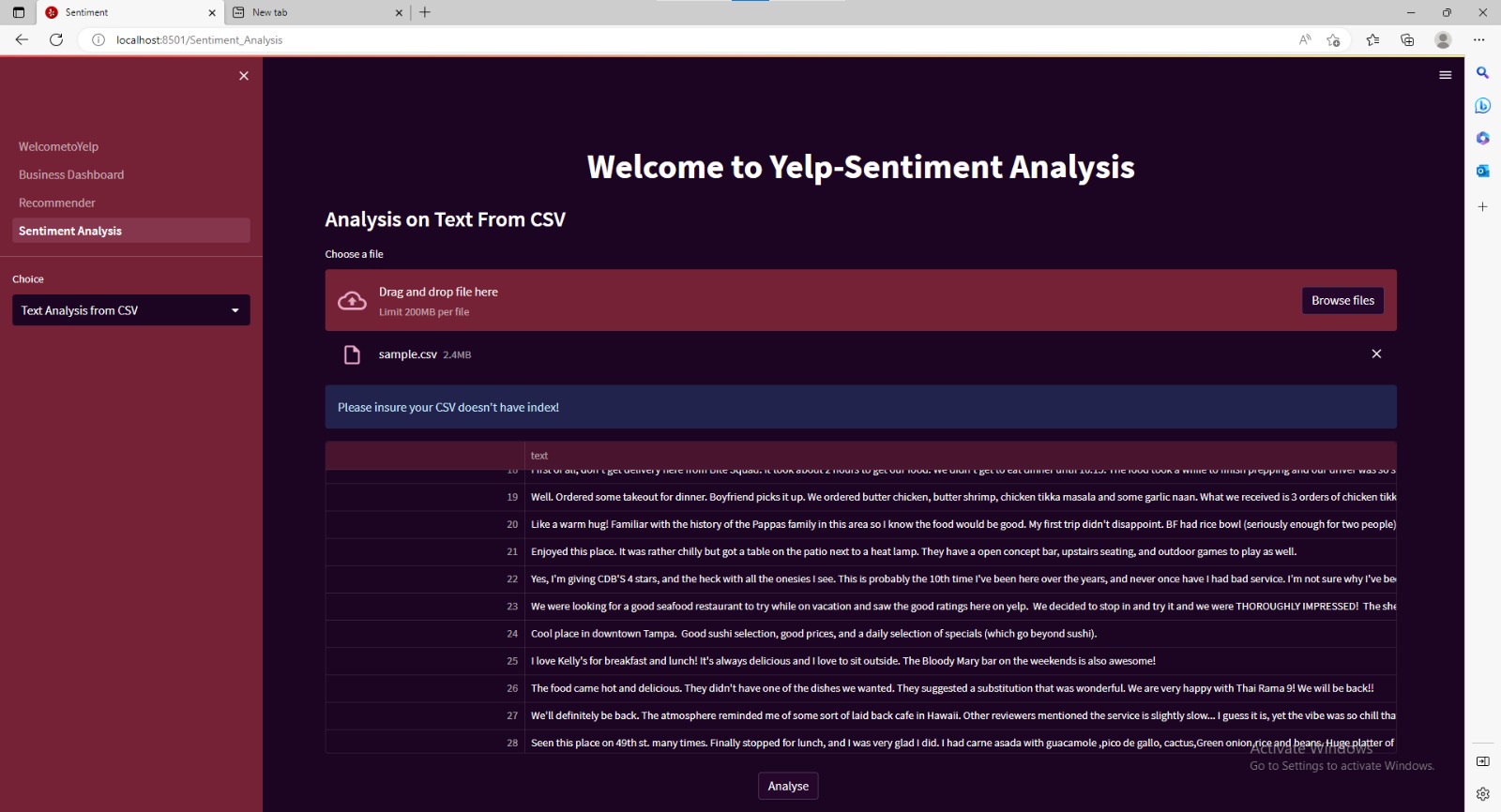






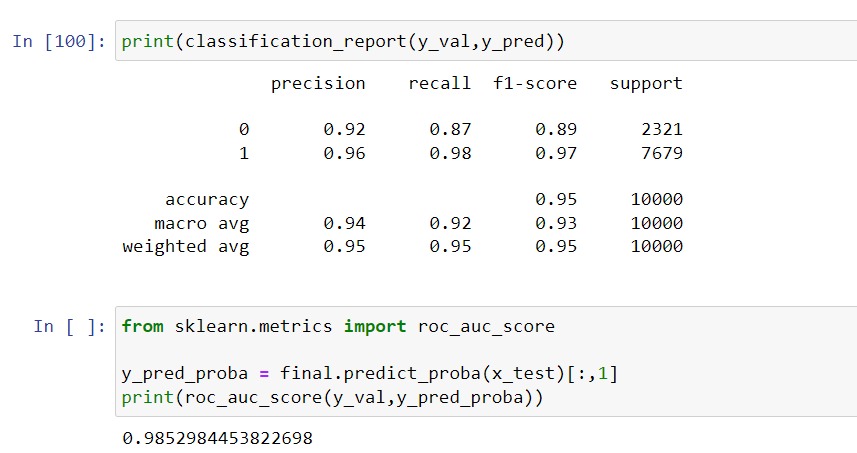






**Chapter 5**

**Results**



**Some Predicted Results –**

**Sentiment Analysis:**



**Recommender System –**



**Chapter 6**

**Conclusion**

**6.1 Conclusion**

The project involved analysing the Yelp reviews dataset using various techniques such as sentiment analysis and content-based filtering. Through the analysis, we were able to gain valuable insights into the preferences of users for restaurants and the sentiments expressed in their reviews.

Based on the analysis, we can conclude that businesses can benefit from leveraging customer reviews to understand their preferences, identify areas for improvement, and make data-driven decisions. For instance, restaurants can use sentiment analysis to identify and address negative reviews to improve customer satisfaction. Additionally, content-based filtering can be used to make personalized recommendations to users based on their preferences. The analysis of the most popular positive and negative words can be expanded to every individual restaurant which can help owners understand what they can improve up and what they are doing good.

It can also help new business owners understand what are the demands of a good restaurant of a specific cuisine and what features they could provide would actually lead to tangible improvement in ratings and revenue.

In conclusion, the use of data analytics techniques on the Yelp reviews dataset can provide businesses with valuable insights that can help them optimize their operations, improve customer experience, and drive business growth.

**6.2 Future Enhancement –**

1. Integration with online platforms: One potential area of future scope for the project is the integration of the analysis with online platforms such as TripAdvisor or Zomato. This would enable restaurants to receive real-time feedback on customer experiences and sentiments, and make data-driven decisions to improve their business operations and expand to include a wider geographic area.
2. Natural Language Processing (NLP): Another area of future scope is to enhance the sentiment analysis process by using NLP techniques such as Named Entity Recognition (NER) and Part of Speech (POS) tagging. This would enable businesses to extract more meaningful insights from the reviews, such as identifying the specific aspects of a restaurant that customers liked or disliked.
3. Collaborative filtering: In addition to content-based filtering, businesses can explore collaborative filtering techniques to make recommendations based on the preferences of similar users. Collaborative filtering can also help address the cold-start problem for new users or businesses with limited data. We can utilize the yelp User dataset for this.
4. Building an API for the whole process and hosting the UI and website on the cloud. This would help in Continuous improvement of the model: As new data becomes available; the model can be continuously improved. The API can be integrated with other systems, such as chatbots, mobile apps, or web applications, to provide a seamless user experience. This would enable businesses to provide customers with personalized recommendations or real-time feedback based on their preference.

**Chapter 7**

**References**

[1] **Sentiment Analysis of Yelp ‘s Ratings Based on Text Reviews** Yun Xu, Xinhui

Wu, Qinxia, Stanford University,

[2] **Beyond the Stars: Improving Rating Predictions using Review Text Content**, Gayatree Ganu, Computer Science Rutgers University, [gganu@cs.rutgers.edu](mailto:gganu@cs.rutgers.edu) | Noemie Elhadad, Biomedical Informatics Columbia University | [noemie@dbmi.columbia.edu](mailto:noemie@dbmi.columbia.edu)

Amelie Marian, Computer Science, Rutgers University | [amelie@cs.rutgers.edu](mailto:amelie@cs.rutgers.edu)

[3] **Large-Scale Sentiment Analysis for News and Blogs**, Namrata Godbole, Manjunath Srinivasaiah, Steven Skiena | Google Inc., New York NY, USA | Dept. of Computer Science, Stony Brook University, Stony Brook, NY 11794-4400, USA

[4] **Sentiment Analysis in Social Media and Its Application: Systematic Literature Review,** Zulfadzli Drus, Haliyana Khali, Azman Hashim International Business School, Kuala Lumpur, 54100, Malaysia

[5] **Collaborative Filtering Recommender Systems** J. Ben Schafer(1) , Dan Frankowski(2) , Jon Herlocker(3) , and Shilad Sen(4) | **(1)** Department of Computer Science University of Northern Iowa Cedar Falls, IA 50614-0507 schafer@cs.uni.edu | **(2)** Department of Computer Science University of Minnesota 4-192 EE/CS Building 200 Union St. SE Minneapolis, MN 55455 {dfrankow , ssen}@cs.umn.edu | **(3)** School of Electrical Engineering and Computer Science Oregon State University 102 Dearborn Hall Corvallis, OR 97331 herlock@eecs.oregonstate.edu

[6] A Survey of Active Learning in Collaborative Filtering Recommender

Systems, Mehdi Elahi (a), Francesco Ricci (b), Neil Rubens (c) | **(a)** Politecnico di Milano, Milan, Italy| **(b)** Free University of Bozen-Bolzano, Bozen-Bolzano, Italy |

**(c)** University of Electro-Communications, Tokyo, Japan

[7] https://xgboost.readthedocs.io/en/stable/

[8] https://scipy.org/

[9] https://scikit-learn.org/stable/index.html

[10] https://spark.apache.org/docs/latest/