INSTITUTE FOR ADVANCED COMPUTING AND SOFTWARE DEVELOPMENT

AKURDI, PUNE – 411044

DOCUMENTATION ON

“RESTAURANT RECOMMENDATION SYSTEM”

PG-DBDA AUG 2024

SUBMITTED BY-

Group no-11

Mr. Sameer Mujawar (248541)

Mr. Abhishek Pawar (248526)

Mrs. Priti Take	**Mr. Rohit Puranik**
Project Guide	**Centre Coordinator**

DECLERATION

I, the undersigned, hereby declare that the project report titled “Restaurant Recommendation System " written and submitted by me to the Institute for Advanced Computing and Software Development, Akurdi, Pune, in fulfilment of the requirement for the award of the Post Graduate Diploma in Big Data Analytics (PG-DBDA) under the guidance of Mrs. Priti Take, is my original work.

I have not copied any code or content from any source without proper attribution, and I have not allowed anyone else to copy my work.

The project was completed using Python libraries. The project was developed as part of my academic coursework. I also confirm that the project is original and has not been submitted previously for any other academic or professional purpose.

Place: **Name: Sameer Mujawar/ Abhishek Pawar**

Date: **Signature:**

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to Mrs. Priti Take, Project Guide, for providing me with the guidance and support to complete this academic project. Their valuable insights, expertise, and encouragement have been instrumental in the success of this project.

I would also like to thank my fellow classmates for their support and cooperation during the project. Their feedback and suggestions were helpful in improving the quality of the project.

I would like to extend my gratitude to Mr. Rohit Puranik, Centre Coordinator, for providing me with the necessary resources and facilities to complete this project. Their support has been crucial in the timely completion of this project.

Finally, I would like to thank my family and friends for their constant encouragement and support throughout the project. Their belief in me has been a constant source of motivation and inspiration.

Thank you all for your support and guidance in completing this academic project.

ABSTRACT

The Zomato Recommendation System project aims to recommend restaurants based on several features such as location, rating, cuisine type, cost, and the availability of services like online orders or table booking. The system focuses on enhancing user experience by providing personalized restaurant recommendations. It leverages Apache Spark, a distributed computing framework, for handling large-scale data and generating real-time recommendations.

Data preprocessing was carried out using PySpark to clean and transform the dataset. Spark's MLlib library was utilized to build recommendation models based on collaborative filtering techniques, particularly the Alternating Least Squares (ALS) algorithm. These models were optimized for scalability and performance, enabling the system to manage high data volumes efficiently. Tableau was employed for data visualization, offering deep insights into user preferences, restaurant trends, and model performance through interactive dashboards and reports.

This project highlights the practical use of big data technologies in the food discovery domain, allowing users to make informed dining decisions. Future enhancements could include real-time integration, incorporating user reviews for sentiment-based recommendations, and deploying the system on a cloud platform for improved accessibility and scalability.

INDEX

CHAPTER 1

INTRODUCTION

INTRODUCTION:

1.1 Description

The Zomato Recommendation System aims to help users discover restaurants based on their preferences. The recommendation system leverages data such as ratings, cuisine types, cost, and more to generate personalized suggestions. The primary goal is to enhance user experience by providing relevant and accurate recommendations.

1.2 Objective of the Project

1. Build a recommendation system using collaborative and content-based filtering.
2. Improve accuracy and relevance of restaurant suggestions.
3. Utilize machine learning techniques to analyse user behaviour and preferences.

1.3 Scope of the Project:

The scope of this project includes building a recommendation engine that can analyse user preferences and restaurant attributes to generate personalized suggestions. The system is designed to improve user experience on the Zomato platform by providing more accurate and relevant recommendations.

Project Scope:

1. **Personalized Recommendations** – The system provides users with restaurant suggestions tailored to their preferences, improving their overall dining experience.
2. **Location-Based Filtering** – Users can search for restaurants based on their geographical location, making the tool highly useful for individuals exploring new areas.
3. **Cost & Rating Consideration** – The model incorporates restaurant pricing and ratings to suggest options that match user budgets and quality expectations.
4. **Machine Learning Integration** – Advanced recommendation algorithms, including collaborative and content-based filtering, enhance prediction accuracy.

1.4 Limitations of the Project

Despite the effectiveness and potential of the Zomato Restaurant Recommendation System, there are several limitations that impact its performance and accuracy. Some of these limitations include:

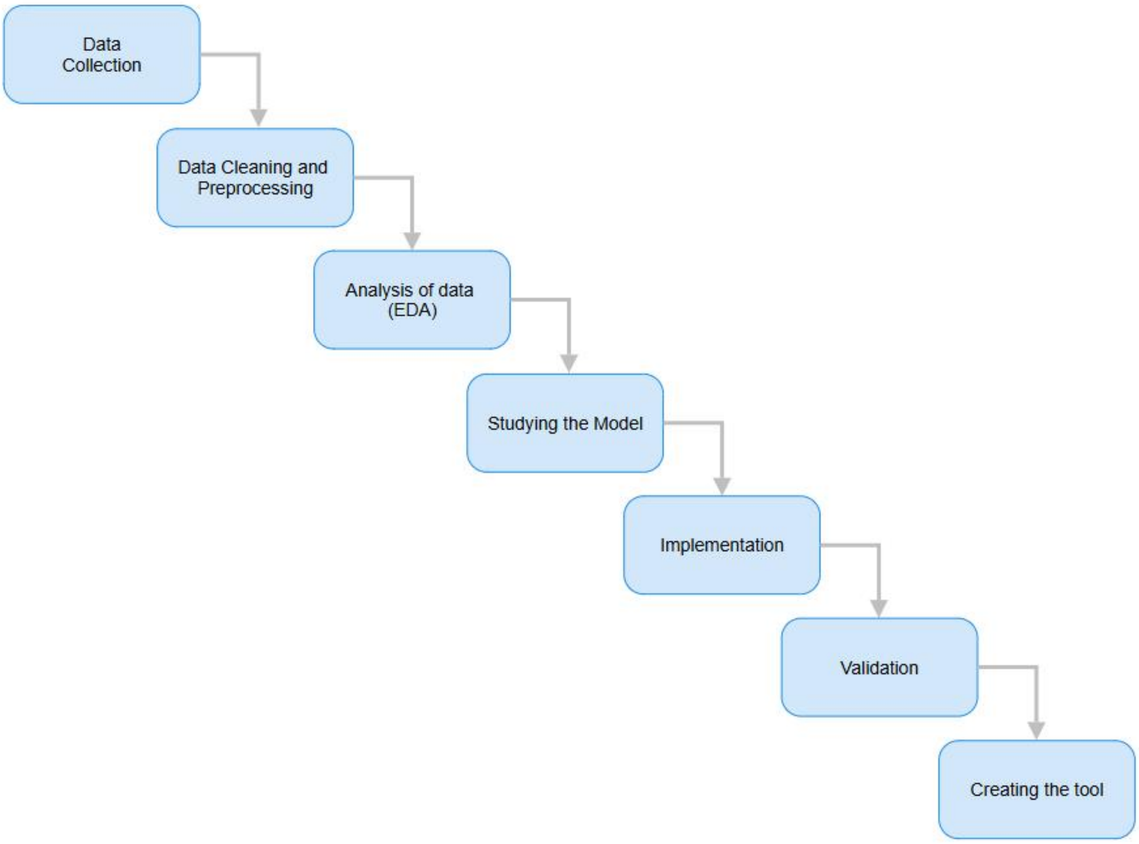
1. **Data Availability and Quality**
    1. The recommendation model is heavily dependent on the quality and completeness of the dataset. Missing or inaccurate restaurant information, customer reviews, and ratings can lead to suboptimal recommendations.
2. **Cold Start Problem**
    1. New restaurants and users with little to no historical data pose a challenge for the system, as collaborative filtering models struggle to provide accurate recommendations without sufficient prior interactions.
3. **Subjectivity in Ratings and Reviews**
    1. Personal tastes and preferences vary significantly, and user ratings may not always be consistent or objective. This can affect the accuracy of recommendations.
4. **Limited Consideration of External Factors**
    1. The model does not account for real-time factors such as restaurant availability, waiting time, special discounts, or temporary closures, which could significantly impact user choices.
5. **Dependency on Historical Data**
    1. The system primarily relies on past user interactions, which may not always reflect changing preferences or emerging restaurant trends.

CHAPTER 2

PROJECT DESCRIPTION

Project Description:

2.1 Project work flow diagram:



2.2 Data Collection

Data Sources:
The primary data source for this project is the Zomato restaurant dataset, which contains information about various restaurants and their attributes. The data includes details such as restaurant name, location, user ratings, cuisine types, average cost for two, and availability of services like online ordering and table booking.

Data Variables:
The following data variables are collected and used for building the recommendation system:

Restaurant Name

Location

Cuisine Types

Average Cost for Two

Rating

Online Order Availability (Yes/No)

Table Booking Availability (Yes/No)

The data is cleaned and pre-processed using PySpark, handling missing values and standardizing formats for analysis. This structured data serves as the foundation for training the collaborative filtering model using Spark's MLlib library.

2.3 Studying the Data

In this step, the restaurant data collected from the Zomato dataset is analysed. It is essential to study and explore the data before using it for building the recommendation model. This step helps identify missing values, outliers, and patterns in the dataset that can influence the model's performance. Following are the key findings of the data:

- Total number of columns: 10
- Total number of numeric columns: 4 (e.g., cost, rating, location id, name id)
- Total number of categorical columns: 6 (e.g., location, cuisine, type)
- Total number of rows (records): 24000+

By understanding the data structure and its attributes, necessary preprocessing steps were performed, including handling missing values, standardizing column formats, and encoding categorical variables to ensure compatibility with the recommendation model.

2.4 Studying the Model

In this step, the recommendation model was selected. The Alternating Least Squares (ALS) algorithm from Apache Spark's MLlib library was chosen for this project. ALS is a collaborative filtering technique that is widely used in recommendation systems. This model takes the following input parameters:

- User ID
- Restaurant ID
- User Rating
- Regularization Parameter
- Rank (Number of Latent Factors)

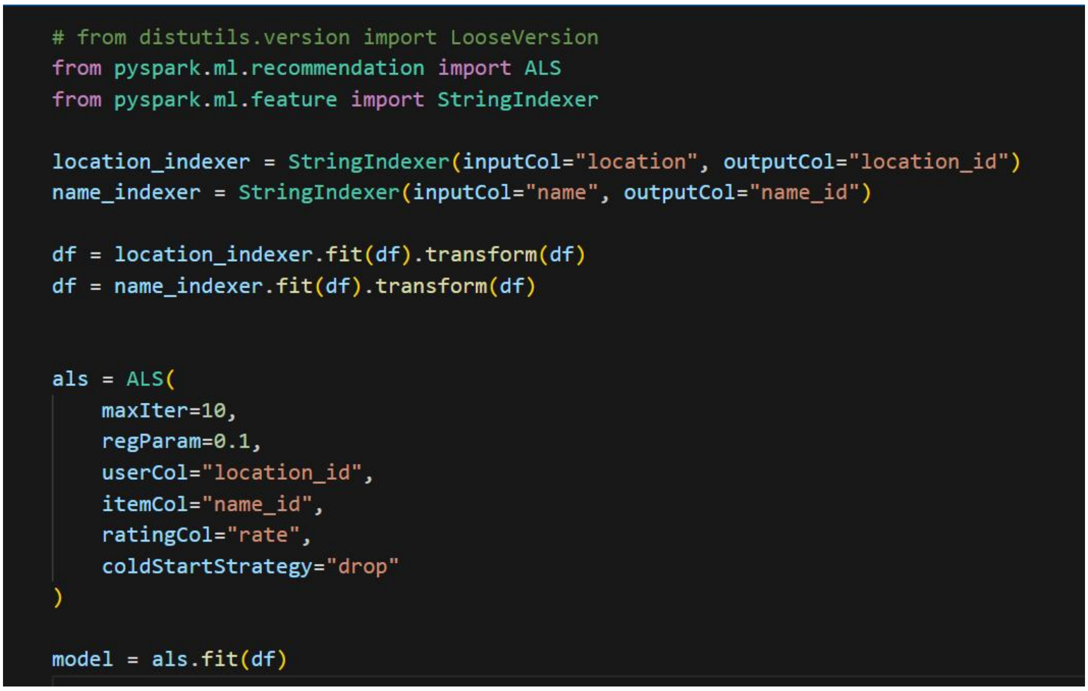
During the model selection process, it was observed that collaborative filtering works well in identifying user preferences by learning patterns from existing user-item interactions.

2.5 Implementing the Model

After studying the ALS model, it was implemented using Apache Spark's PySpark module. The following steps were followed:

1. Data Preparation:
   The dataset was cleaned and transformed into the required format for ALS. Missing values were filled with appropriate defaults, and user and restaurant IDs were indexed for the model.

2. Model Training:
   The ALS model was trained using different hyperparameter combinations to optimize performance. Key parameters such as the number of latent factors, regularization parameter, and number of iterations were tuned to minimize error.

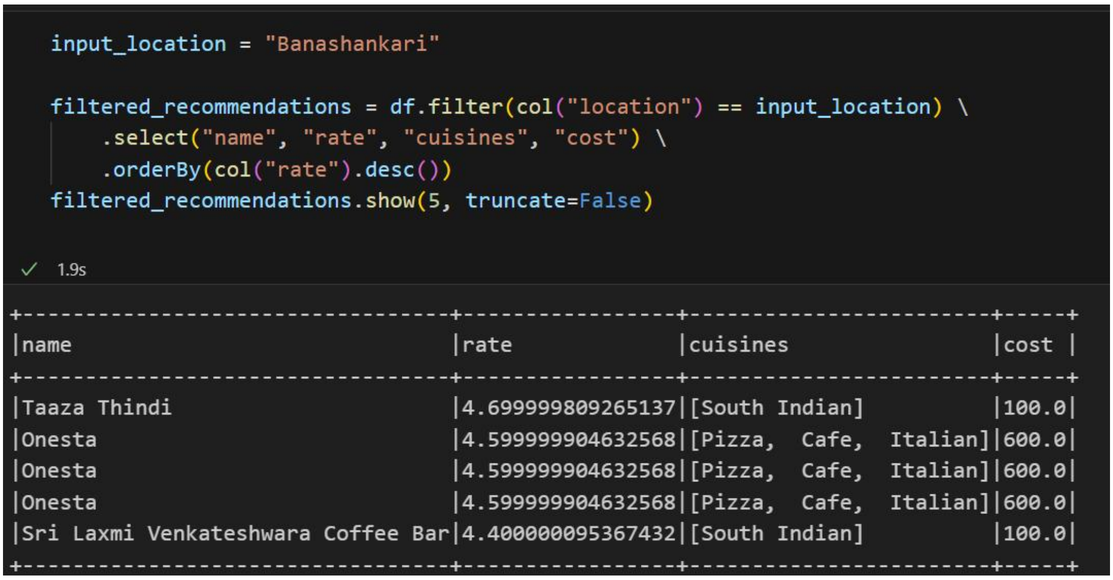


```python
# from distutils.version import LooseVersion
from pyspark.ml.recommendation import ALS
from pyspark.ml.feature import StringIndexer

location_indexer = StringIndexer(inputCol="location", outputCol="location_id")
name_indexer = StringIndexer(inputCol="name", outputCol="name_id")

df = location_indexer.fit(df).transform(df)
df = name_indexer.fit(df).transform(df)


als = ALS(
    maxIter=10,
    regParam=0.1,
    userCol="location_id",
    itemCol="name_id",
    ratingCol="rate",
    coldStartStrategy="drop"
)

model = als.fit(df)
```

3. Input Parameters:
   The following inputs were used for the ALS model:

   - Location ID
   - Restaurant ID
   - Rating

4. Model Output:
   The trained ALS model provides personalized restaurant recommendations for each user based on their past interactions and similarities with other users.



```
input_location = "Banashankari"

filtered_recommendations = df.filter(col("location") == input_location) \
    .select("name", "rate", "cuisines", "cost") \
    .orderBy(col("rate").desc())
filtered_recommendations.show(5, truncate=False)
```

✓ 1.9s

```
+---------------------------------+-----------------+-----------------------+-----+
|name                             |rate             |cuisines               |cost |
+---------------------------------+-----------------+-----------------------+-----+
|Taaza Thindi                     |4.699999809265137|[South Indian]         |100.0|
|Onesta                           |4.599999904632568|[Pizza,  Cafe,  Italian]|600.0|
|Onesta                           |4.599999904632568|[Pizza,  Cafe,  Italian]|600.0|
|Onesta                           |4.599999904632568|[Pizza,  Cafe,  Italian]|600.0|
|Sri Laxmi Venkateshwara Coffee Bar|4.400000095367432|[South Indian]         |100.0|
+---------------------------------+-----------------+-----------------------+-----+
```

CHAPTER 3

Model Description

3. Model Description

3.1 Alternating Least Squares (ALS) Model

The Alternating Least Squares (ALS) algorithm is a widely used collaborative filtering technique for building recommendation systems. It works by factorizing the user-item interaction matrix into smaller latent factors and iteratively refining predictions to minimize error. This approach is particularly suited for large-scale datasets like restaurant recommendation systems, providing personalized suggestions based on user preferences and historical data.

In this project, the ALS model from Apache Spark's MLlib library was chosen due to its scalability and ability to handle sparse matrices effectively. The model recommends restaurants to users based on features such as location, cuisine type, cost, and rating.

3.2 Important Terms Related to the Recommendation Model

- **User ID**: Unique identifier for each user in the dataset.
- **Restaurant ID**: Unique identifier for each restaurant in the dataset.
- **User Rating**: The rating given by the user to a specific restaurant. This serves as the primary input for training the recommendation model.
- **Location**: The geographical area where the restaurant is located, which plays a crucial role in personalizing recommendations.
- **Cuisines**: Types of cuisines offered by the restaurant (e.g., South Indian, Chinese, Seafood), helping in filtering suggestions based on user preferences.

3.3 Data Features

Key features from the dataset used in this recommendation system:

- **Location**: Indicates the restaurant's geographical location, with 43 unique locations in the dataset.
- **Cost**: Represents the average cost for two people, which ranges from low-cost eateries to high-end dining.

- **Rating**: User ratings on a scale, helping to rank restaurants by popularity and user satisfaction.
- **Cuisine**: Lists the type of cuisines available at each restaurant, enabling content-based filtering for cuisine-specific recommendations.

3.3 ALS Assumptions

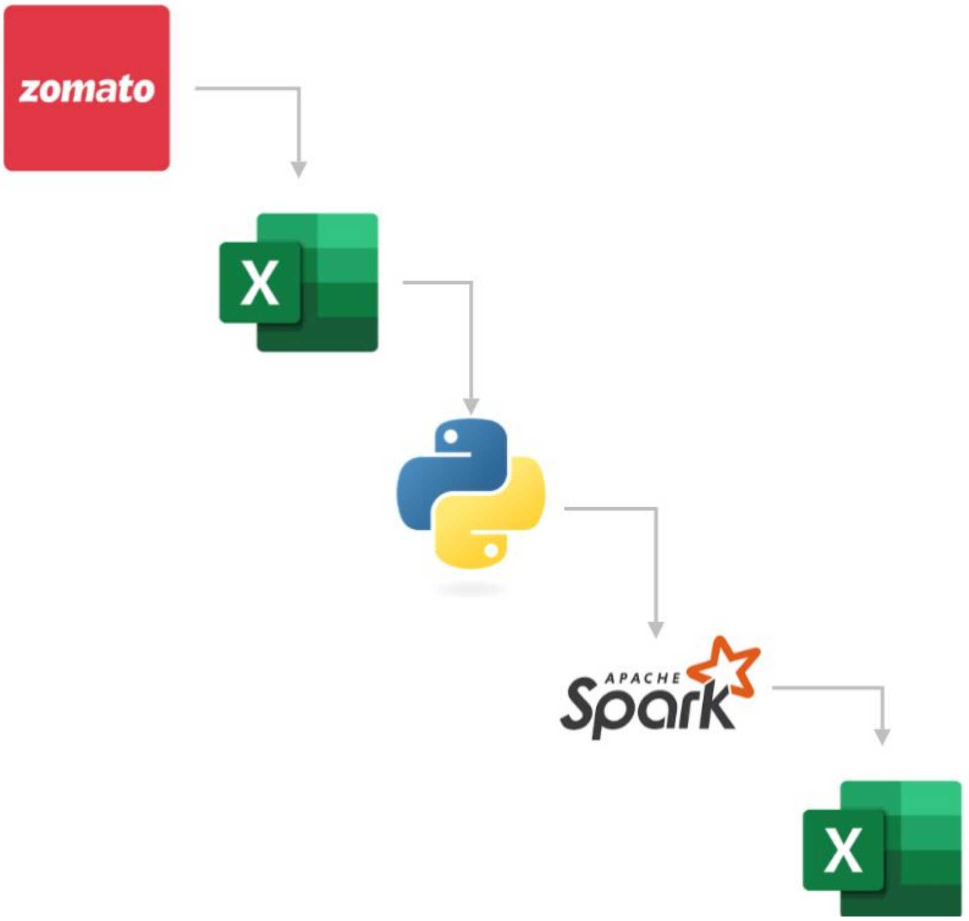
The ALS model is based on the following assumptions:

1. **Implicit Feedback**: ALS can handle implicit feedback (e.g., clicks or views) and explicit ratings, but in this project, explicit ratings were used.
2. **Linear Independence**: The user and item factors should be linearly independent for effective factorization.
3. **Regularization**: Regularization helps reduce overfitting by penalizing large factor values.
4. **Sparsity in the Rating Matrix**: The user-item matrix is expected to be sparse, meaning most users have rated only a small subset of the available items.
5. **Additive Latent Features**: Each latent feature contributes additively to the final prediction score.

CHAPTER 4

Data Flow

4.1 Data Flow of Project



The data flow in the Zomato Recommendation System project is explained as follows:

1. Data Source:
The primary data source for this project is the Zomato dataset, which contains detailed information about restaurants. The data includes attributes such as restaurant name, location, cuisine type, rating, cost, and availability of online ordering and table booking services.

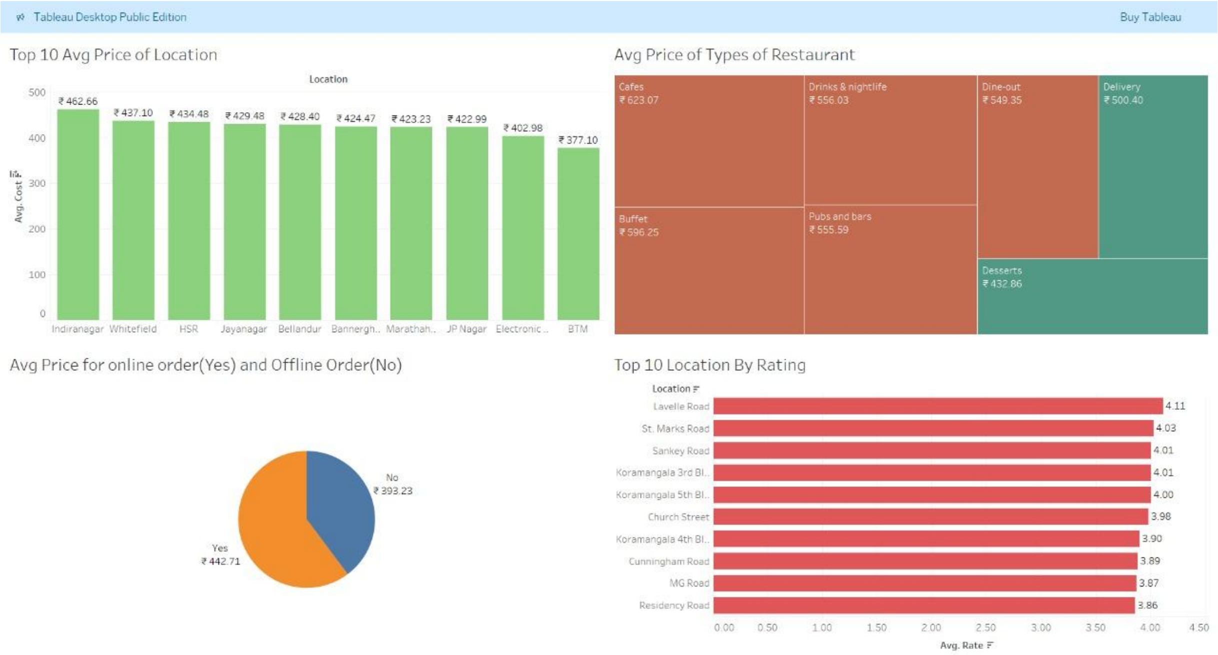
2. Data Preprocessing:
Data is cleaned and transformed using PySpark to ensure consistency and accuracy. This includes handling missing values, encoding categorical variables, and normalizing numeric fields like ratings and cost

3. Data Preprocessing:
Data is cleaned and transformed using PySpark to ensure consistency and accuracy. This includes handling missing values, encoding categorical variables, and normalizing numeric fields like ratings and cost.

4. Model Training:
The pre-processed data is used to train the recommendation model using the ALS (Alternating Least Squares) algorithm from Apache Spark's MLlib library. The model learns patterns from user-restaurant interactions to generate personalized suggestions.

6. Recommendation Generation:
Once the model is trained, it predicts restaurant recommendations for each user based on their preferences and interactions.

7. Visualization:
The results are visualized using Tableau, where interactive dashboards display insights such as popular restaurants, user trends, and model performance metrics.



Top 10 Avg Price of Location

Avg Price of Types of Restaurant

Avg Price for online order(Yes) and Offline Order(No)

Top 10 Location By Rating

CHAPTER 5

Tools and Technologies used

Tools used in Project:

5.1 Programming Languages & Frameworks

Python

Python is a powerful, high-level programming language known for its simplicity and extensive libraries for data science, machine learning, and web development. In this context, Python is used for:

- **Data Preprocessing**: Cleaning, transforming, and structuring data before feeding it into machine learning models.
- **Model Training**: Implementing and training machine learning models, including recommendation systems, using libraries such as PySpark.
- **Backend Development**: Handling server-side logic, integrating machine learning models with web applications, and managing API interactions.

Flask

Flask is a lightweight and flexible web framework for Python. It is often used to build web applications and APIs with minimal overhead. In this case, Flask is used for:

- **API Development**: Creating RESTful APIs that allow communication between the machine learning model and the frontend application.
- **Handling User Interactions**: Receiving user inputs, processing requests, and returning responses (such as recommendations).
- **Integration with Machine Learning Models**: Serving predictions by exposing trained models as endpoints that frontend applications can query.

5.2 Machine Learning Libraries

PySpark

PySpark is the Python API for Apache Spark, a distributed computing framework designed for big data processing and analytics. It is especially useful for:

- **Collaborative Filtering using ALS**: Alternating Least Squares (ALS) is a matrix factorization technique used in recommendation systems to predict user preferences.

- **Handling Large-Scale Data**: PySpark is optimized for processing large datasets efficiently, making it ideal for real-world recommendation systems.

Pandas

Pandas is a data manipulation and analysis library for Python. It provides data structures like Data Frames, which allow for efficient data handling. It is used for:

- **Data Manipulation**: Cleaning and structuring data for use in machine learning models.
- **Transformation**: Applying functions, aggregations, and filters to extract meaningful insights.
- **Numerical Operations**: Performing calculations, aggregations, and statistical analysis on datasets.

5.3 Visualization Tools

Tableau

Tableau is a powerful data visualization tool that helps in creating interactive and insightful visual representations of data. It is used for:

- **Plotting Restaurant Ratings**: Visualizing user reviews, average ratings, and trends.
- **Cost Distributions**: Analysing price variations among restaurants to help users make cost-effective decisions.

CHAPTER 6

Project Requirements

6. Project Requirements

6.1 Hardware Requirement

- 500 GB Hard Drive (Minimum requirement)
- 8 GB RAM (Minimum requirement)
- PC with x64-bit CPU
- Internet Connection with a minimum speed of 5 Mbps

6.2 Software Requirement

- Operating System: Windows / Mac / Linux
- Python Version: 3.9 or above
- Python Libraries:

o PySpark

o Pandas 1.5.3: For data manipulation and analysis.

o Scikit-learn (Sklearn 0.0. post1): For data splitting and evaluation metrics.

o Tableau: For interactive data visualization.

- Development Environments:

o VS Code / Anaconda / Spyder (for coding and development)

- Microsoft Excel (MS Office 2016 or above): For initial data collection and reporting

CHAPTER 7

Future Scope

7. Future Scope

The Zomato Recommendation System can be enhanced and expanded in the following ways:

1. Cuisines-based Personalization
   Improve the recommendation system by adding more advanced filtering and personalized suggestions based on user cuisine preferences and cost sensitivity.

2. Geolocation-based Recommendations
   Incorporate geolocation to provide proximity-based restaurant suggestions, helping users find options nearby in real time.

3. Visualization Enhancements
   Create real-time dashboards in Tableau to display dynamic insights such as trending restaurants, top-rated cuisines, and user preference patterns. This would help stakeholders and users better understand the data.

Overall, these enhancements would significantly improve the Zomato Recommendation System's scalability, accuracy, and user engagement, making it a comprehensive restaurant discovery tool.

CHAPTER 8

Conclusion

8. Conclusion

In conclusion, this project successfully demonstrated the use of Apache Spark and Tableau to build a scalable and efficient recommendation system for Zomato. The recommendation engine was designed to analyse restaurant data and provide personalized suggestions based on attributes such as location, rating, cuisine type, and cost.

The project highlighted the power of PySpark for handling large-scale data and ALS (Alternating Least Squares) for collaborative filtering to generate recommendations. By preprocessing and cleaning the data, training the model, and visualizing results in Tableau, the system delivered insights into user preferences and restaurant trends.

This recommendation system serves as a valuable tool for improving user experience on platforms like Zomato by helping users make better dining decisions. The integration of big data technologies, combined with powerful visualization tools, ensures that the system is both robust and scalable.

Overall, the project showcases the practical application of distributed computing and data-driven approaches in the restaurant discovery domain. Future enhancements, such as real-time data integration and cloud deployment, would further expand the system's capabilities and improve its relevance and usability for a broader audience.

CHAPTER 9

References

9. References

Bibliography

- Apache Spark. (2023). MLlib Guide. Retrieved from https://spark.apache.org/docs/latest/ml-guide.html
- Pandas Documentation. (2023). User Guide. Retrieved from https://pandas.pydata.org/docs/user_guide/index.html
- Scikit-learn Documentation. (2023). Examples. Retrieved from https://scikit-learn.org/stable/auto_examples/index.html
- Schedule Library. (2023). Examples. Retrieved from https://schedule.readthedocs.io/en/stable/examples.html
- PySpark API Documentation. (2023). Retrieved from https://spark.apache.org/docs/latest/api/python/
- Tableau. (2023). Visual Analytics Guide. Retrieved from https://www.tableau.com/