RSET
RAJAGIRI SCHOOL OF
ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

Project Report On

MapMySuccess

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

By

Rahul Varghese (U2103169)

Rebecca Liz Punnoose (U2103171)

Richu Kurian (U2103175)

Rohan Joseph Arun (U2103180)

Under the guidance of

Ms. Ann Grace

Assistant Professor

Department of Computer Science and Engineering
Rajagiri School of Engineering & Technology (Autonomous)
(Parent University: APJ Abdul Kalam Technological University)
Rajagiri Valley, Kakkanad, Kochi, 682039
April 2025

CERTIFICATE

This is to certify that the project report entitled ***"MapMySuccess"*** *is a bonafide record of the work done by* ***Rahul Varghese (U2103169), Rebecca Liz Punnoose (U2103171), Richu Kurian (U2103175) , Rohan Joseph Arun (U2103180)*** *submitted to the Rajagiri School of Engineering & Technology (RSET) (Autonomous) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in "Computer Science and Engineering" during the academic year 2021-2025.*

Ms. Ann Grace
Project Guide
Assistant Professor
Dept. of CSE
RSET

Ms. Sangeetha Jamal
Project Co-ordinator
Assistant Professor
Dept. of CSE
RSET

Dr. Preetha K G
Professor and HoD
Dept. of CSE
RSET

ACKNOWLEDGMENT

We wish to express our sincere gratitude towards **Rev. Dr. Jaison Paul Mulerikkal CMI**, Principal of RSET, and **Dr. Preetha K.G.**, Head of the Department of Computer Science for providing us with the opportunity to undertake our project, "MapMySuccess".

We are highly indebted to our project coordinator, **Ms. Sangeetha Jamal**, Assistant Professor, Computer Science and Engineering, for her valuable support.

It is indeed our pleasure and a moment of satisfaction for us to express our sincere gratitude to our project guide **Ms. Ann Grace**, Assistant Professor for her patience and all the priceless advice and wisdom she has shared with us.

Last but not the least, we would like to express our sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

Rahul Varghese
Rebecca Liz Punnoose
Richu Kurian
Rohan Joseph Arun

Abstract

This MapMySuccess project is a clever machine learning-based way to assess how well a restaurant will perform in a specific area. It takes into account a number of important elements, like the location's prominence, crowding, number of nearby eateries, etc. Highly skilled predictive algorithms that leverage real-time data gathered from Google APIs are used to assess these parameters. The correctness and integrity of these data are guaranteed by careful preprocessing and feature engineering. Map-MySuccess differs from traditional approaches that are typically speculative or superficial.In contrast, Map-MySuccess uses a methodical, data-driven approach. By basing its forecasts on real environmental and commercial data, it does more than just forecast success ratings; it also gives customers confidence in the results. With the help of its dynamic map display and clear visual feedback, the tool enables restaurant operators to choose locations more wisely. MapMySuccess wants to transform site planning in the restaurant business by helping them reduce risk and make better decisions by providing hard facts available insight.

Contents

List of Abbreviations

- **ML** - Machine Learning
- **AI** - Artificial Intelligence
- **SHAP** - Shapley Additive Explanations
- **API** - Application Programming Interface
- **PCA** - Principal Component Analysis
- **CV** - Computer Vision
- **NLP** - Natural Language Processing
- **MSE** - Mean Squared Error
- **DMLRA** - Deep Multi-Task Learning with Relational Attention
- **ROC** - Receiver Operating Characteristic
- **AUC** - Area Under the Curve
- **CPU** - Central Processing Unit
- **GPU** - Graphics Processing Unit
- **RAM** - Random Access Memory
- **SSD** - Solid-State Drive
- **SVM** - Support Vector Machine
- **XAI** - Explainable Artificial Intelligence

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The decision of location is essential to a restaurant's existence. Due to a lack of suitable tools and methods, many restaurant operators are currently faced with the prospective evaluation of new locations. Usually, decisions are made intuitively or based on scant knowledge, which results in poor site selection, wasted investment, and failed businesses. Data-driven decision-making is now even more important in the food and beverage business due to urbanization and increased competition. The main factors that affect success include visibility, parking availability, population density, and other factors close to competition. However, these factors are interconnected and require in-depth investigation to produce useful knowledge.Our method uses machine learning to forecast success rates for potential restaurant locations in an effort to address these problems. It collects data in real time, analyzes it, and applies predictive modeling to help restaurant operators make better decisions, reducing investment risks and increasing revenue.

1.2 Problem Definition

The project's goal is to create a machine learning-based system that can forecast restaurant sites' likelihood of success by analyzing critical elements like population density, parking availability, visibility, and nearby competition. In general, restaurant owners face serious difficulty in determining their locations owing to lack of sound tools for evaluating such above factors, mostly leading to inappropriate site selection coupled with fiscal losses and failures in business. Thus, by providing data-driven insights, this project allows restaurant owners to make smart decisions towards better chances of success.

1.3 Scope and Motivation

1.3.1 Scope

This study intends to create a machine learning-based model that predicts the potential for success in restaurant site locations. The system integrates and analyses very critical parameters such as population density, parking availability, visibility, and competitors. Taking this into real-time capabilities like API inputs, web scrapping hence very accurate and reliable predictions according to individual input model. This project shall be beneficial for restaurant owners when minimizing the chances of possible risk as far as site selection is concerned and profitability.

1.3.2 Motivation

This project is motivated mainly by the high rate of failures of new restaurants due to poor choice in locations. The old method usually depended on the subjective judgment of some individuals or incomplete data that led to very unreliable outputs. With the sudden influx of data and the high level of advancement in machine learning, there is a big opportunity out there in improving decision-making in this area. it may convince and motivate non-technical users to adopt the system. Eventually, it will enable restaurateurs to possess effective, reliable, and data-driven tools for making profitable informed decisions.

1.4 Objectives

1. To build a machine learning system that can predict the success of a restaurant based on key factors such as location, cuisine, and average price.

2. To use a Swiggy dataset as the base and enrich it using real-time data collected via Google APIs (Places, Roads, and Routes), capturing features like rating, traffic, population density, road proximity, and nearby competition.

3. To handle large-scale data collection efficiently by splitting the dataset into subsets and processing them in parallel.

4. To derive additional meaningful features such as competition score and true success score using domain logic and existing data (e.g., rating and rating count).

5. To experiment with different machine learning models such as Random Forest, XGBoost, and LightGBM, and identify the best-performing model for this use case.

6. To design and implement a user-friendly React-based frontend that allows users to input location, cuisine, and price, and view predictions on an interactive map.

7. To connect the frontend with a backend that fetches required data via APIs, preprocesses inputs, sends them to the model, and returns the predicted success score.

1.5 Challenges

Inconsistencies, redundancies, or missing values that require thorough preprocessing may arise when integrating and preparing data from many sources, such as web scraping and APIs. Furthermore, maintaining the model's predictions in a dynamic, real-world environment is an especially difficult undertaking. The project's complexity is further increased by making trade-offs between computational resources, prediction accuracy, and easily explicable explainability.

1.6 Assumptions

The project makes the assumption that the input data—such as traffic volume and population density—is accurate and sourced from reliable sources, such as external data sets and APIs. Additionally, it assumes that they enter precise details on the type of restaurant, price range, and preferred location. Additionally, it is expected that external variables like the state of the economy and the dynamics of competitors will remain relatively stable throughout the forecast period.

1.7 Industrial Relevance

Both society and the restaurant industry can greatly benefit from this endeavor. It provides the industry with a data-supported decision-making tool that helps owners, managers, and investors make well-informed decisions on restaurant locations. It helps to maximize business investments, reduce risk, and increase profitability by analyzing an area's population density, parking availability, and nearby rivals. In terms of society, the idea might support economic growth by promoting the establishment of profitable eateries

in suitable areas, creating jobs, and enhancing community involvement. The project contributes to the development of a more vibrant and successful local economy by promoting sustainable business practices and minimizing wasted expenditure.

1.8 Organization of the Report

The project is fully summarized in this paper. The study's history is given in the introduction, which also explains the goals, driving forces, and problem description. A comprehensive literature analysis that explains relevant research on predicting staff turnover and business performance follows. This section outlines current approaches, their shortcomings, and the current work's contributions. The methodology section provides a thorough explanation of the stages involved in gathering data, preprocessing, training the model and choosing features, evaluating the results, and carrying out the research. The findings and an examination of performance metrics like R2 (R-squared) and MSE (Mean Squared Error) are then shown in the results section. Additionally, a comparison of the outcomes with existing models is given.

Chapter 2

Literature Survey

2.1 Deep Multi-Task Learning with Relational Attention

2.1.1 Introduction

Deep Multi-Task Learning with Relational Attention, or DMLRA, resolves the problem of dynamically learning task relationships in multi-task learning (MTL) [1]. Strict associations, which are the basis of traditional MTL, fail to properly represent the dynamics of actual data. By creating task relationships during training, DMLRA's relational attention module promotes performance and broad scalability in CV and NLP tasks.

2.1.2 Methodology

The DMLRA model includes:

- **Shared and Task-Specific Layers:** Acquiring general information in shared layers, task-specific features in task-specific layers, and shared and task-specific layers.
- **Relational Attention Module:** A trainable relationship matrix dynamically captures task interdependencies without manual definitions.
- **Training and Evaluation:** The model is trained using a combined loss function then tested on benchmarks over NLP and CV tasks.

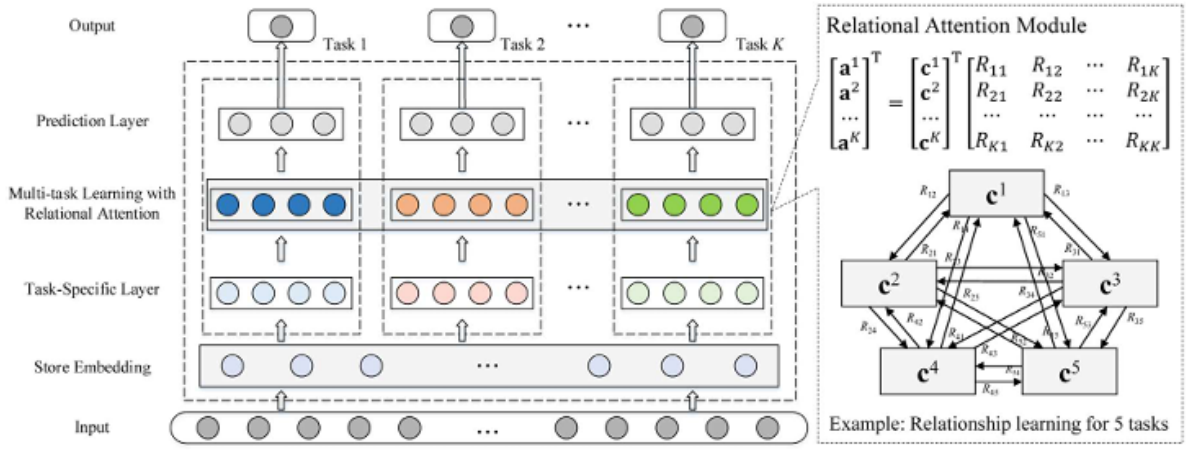


Figure 2.1: An overview of the Deep Multi-task Learning with Relational Attention (DMLRA) module.

2.1.3 Results

- **Performance:** DMLRA outperformed baseline MTL and single-task models in F1 scores, accuracy, and mean squared error.
- **Relational Attention:** The module effectively identified task interdependencies, enhancing feature sharing and interpretability.
- **Generalization:** The model had better generalization on unseen data compared to traditional MTL approaches.

2.2 Prediction of Employee Turnover Using Random Forest Classifier with Intensive Optimized PCA Algorithm

2.2.1 Introduction

[1] Employee turnover is one of the biggest issues that an organization faces; it results in huge costs and disruption. Knowing who is likely to leave the company would help organizations proactively improve retention and reduce the financial impact of turnover. Over the last few years, machine learning algorithms have been widely applied to develop predictive models for employee turnover. Among these algorithms, Random Forest classifiers have been known to be particularly effective in their ability to manage large datasets and extract non-linear relationships among variables. On the other hand, high

dimensional data poses the challenge of having difficulty in model building. A common technique in this regard is PCA, as it reduces the dimensionality of the data. To predict employee turnover more accurately and efficiently, the study recommends the approach of improving it by the integration of PCA methodology with Random Forest Classifiers.

2.2.2 Methodology

Data Collection and Preprocessing

The data used in this study is on different attributes about employees like age, job role, salary, years at the company, performance metrics, and historical turnover data. Initially, the dataset is cleaned by removing missing or irrelevant data points. Outliers are also detected and properly managed. The rest of the features are then normalized to ensure they are of similar scale.

Principal Component Analysis (PCA)

PCA is a dimensionality reduction technique. This technique reduces the dimension of a set of data in order to highlight the most significant traits that explain the variance in the data. We apply an optimized PCA algorithm, which improves traditional PCA by improving the selection of principal components in this study. Optimized PCA is achieved through the application of a feature selection technique, where we find the components that most contribute to the predictions of employee turnover.

Random Forest Classifier

Random Forest Classifier is a technique of ensemble learning that involves creating multiple decision trees and combines the results from all of them for better accuracy and robustness. Here we applied the Random Forest model in predicting employee turnover using the principal components obtained from PCA as the input features. We trained the model on the training dataset with optimal number of trees and depth after hyperparameter tuning.

Model Evaluation

Several metrics, including accuracy, precision, recall, F1 score, and the Area Under the Receiver Operating Characteristic Curve (AUC-ROC), are used to assess the model's performance. By using cross-validation, the model will be guaranteed to generalize effectively for unknown data in order to prevent overfitting.

2.2.3 Results

This optimized PCA algorithm decreases the dimensionality of the data and produced an efficient model with zero sacrifice on prediction accuracy. Models that are trained using reduced features for Random Forest Classifier improved by considerable performance metrics than those that are trained on high-dimensional original data.

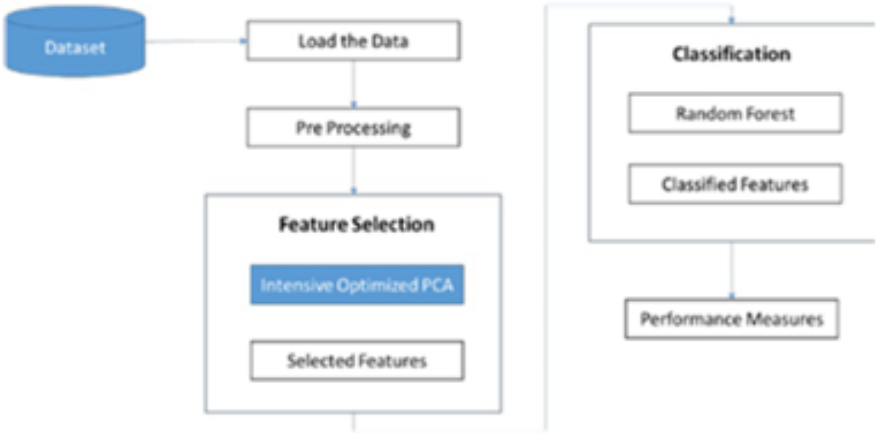


Figure 2.2: Overall flow of work

2.3 A Machine Learning, Bias-Free Approach for Predicting Business Success Using Crunchbase Data

2.3.1 Introduction

[2] Business success prediction is a critical tool in identifying promising ventures and allocating resources efficiently. However, traditional predictive methods often suffer from subjective biases, leading to inconsistent results and missed opportunities. Using information from Crunchbase, a comprehensive database of startups, their funding, and other pertinent factors, this study presents an ML-based approach for predicting the chances

of business success. This study attempts to provide an objective, data-driven solution to what was traditionally a subjective problem by focusing on developing an unbiased prediction pipeline.

2.3.2 Methodology

Data Collection and Preprocessing

The main source of Crunchbase data used for this research includes industry sectors, funding stages, geographies, and team compositions. Such data was thoroughly preprocessed: duplicates and inconsistencies were removed from the data through cleaning, and the numeric features like funding amount and team size were further normalized.

The sensitive attributes, which include the demographics of founders, are excluded or processed to remove bias. Then, a stratified sampling is applied to divide the dataset into training and test subsets in a way that preserves the class distribution.

Feature Selection and Engineering

Mutual information scores are used to feature selection to highlight the most relevant features with maximum predictability along with interpretability. Moreover, derived features including growth rate of funding and competitive density in the startup's domain are engineered so that the predictions quality is increased.

Model Training

Various machine learning algorithms are explored to build the predictive model, including Logistic Regression, Support Vector Machines (SVM), and Gradient Boosted Trees. The best-performing model is selected based on cross-validation results and evaluation metrics such as precision and recall.

Hyperparameter tuning is performed by grid search for the chosen model to maximize accuracy and generalization ability.

Fairness and Bias Mitigation

By using fairness-aware machine learning approaches, bias-free predictions are ensured. Techniques like adversarial debiasing and reweighting are used during training. To guar-

antee that performance is objective, metrics such as demographic parity, post-training, and disparate impact ratio are measured.

2.3.3 Results

This machine learning framework has a high prediction accuracy and is thus a good model for forecasting business success.

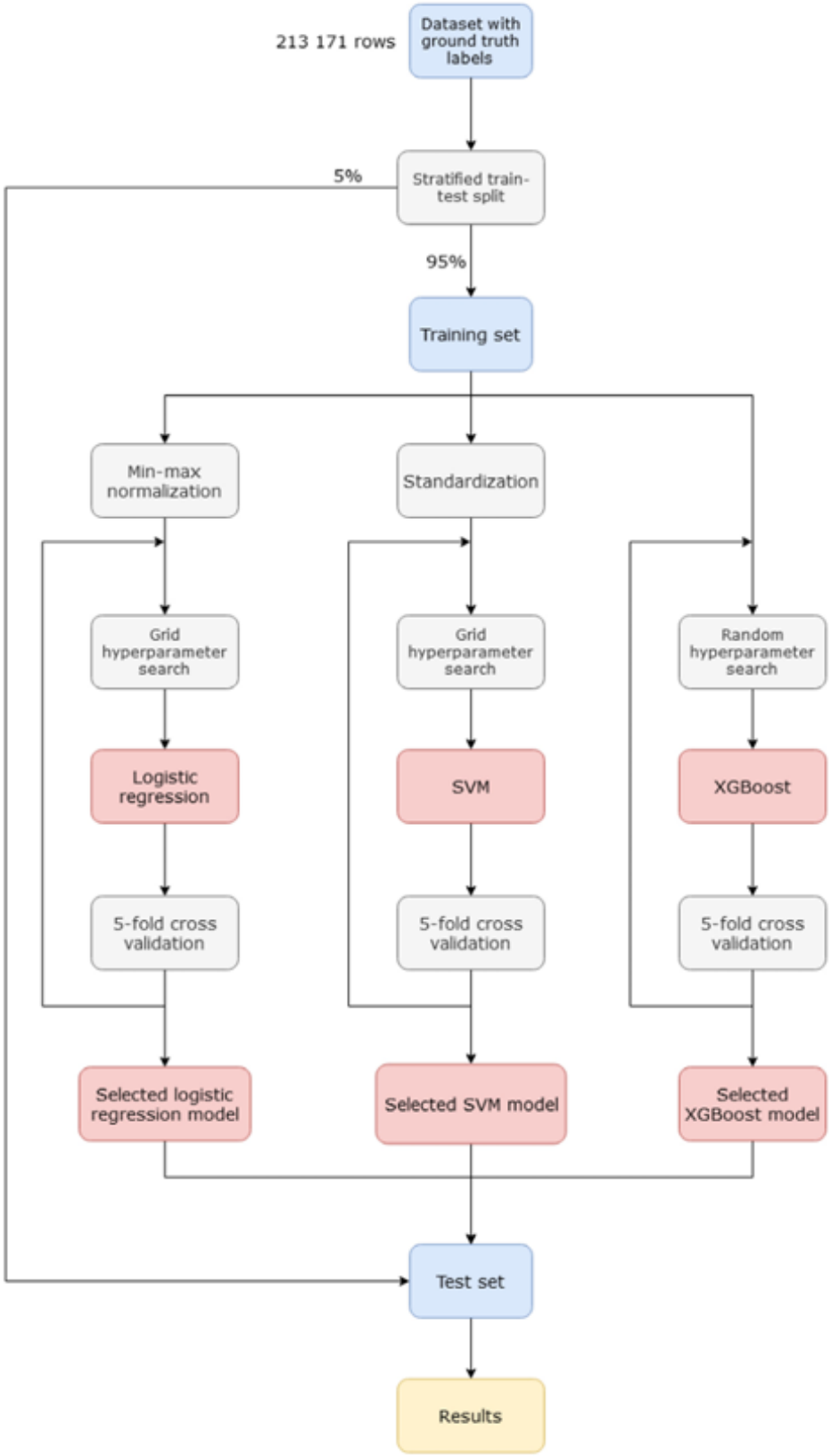


Figure 2.3: Experiment setup

2.4 The Shapley Value in Machine Learning

2.4.1 Introduction

The Shapley value [3], is a concept in cooperative game theory that has become the cornerstone in machine learning for the fair allocation of credit among contributors such as features, data points, or models. Its axiomatic qualities of efficiency, symmetry, and fairness make it a compelling contender for use in data valuation, explainability, and feature selection, among other applications. By measuring contributions, it connects theoretical strength with applications: A unifying paradigm addressing attribution issues is provided by Shapley value. This book discusses its limitations by looking at its underlying ideas, computational techniques, and applications.

2.4.2 Methodology

The Shapley value gives credit to contributors by averaging their marginal contributions across all the possible orders in which the contributors can be selected. Contributors here could be features, data points, or even ensemble models; the payoff could represent the metrics of goodness of fit and model accuracy, among others. Considering the factorial complexity of exact calculations, approximations such as Monte Carlo sampling and SHAP make Shapley values practical for large-scale tasks. Applications include feature selection, which ranks features based on their contribution to model performance, and explainability, in which they decompose predictions into feature-level attributions. Additionally, in data valuation, Shapley values assess the importance of individual data points, guiding data collection and sharing strategies.

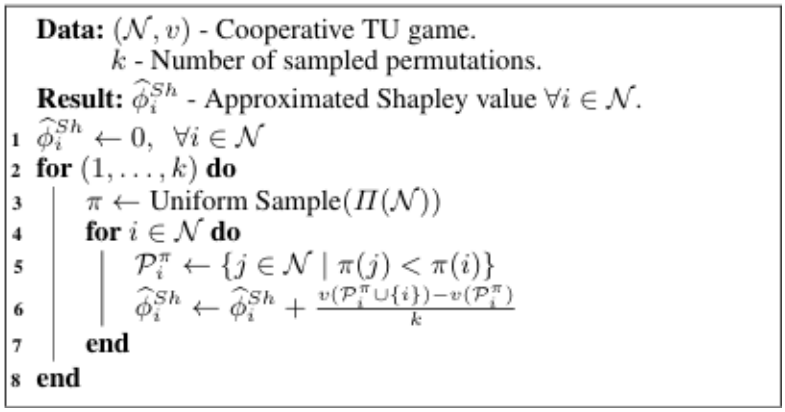


Figure 2.4: Monte Carlo permutation sampling approx imation of the Shapley value.

2.4.3 Results

The Shapley value has been effective in feature selection, providing interpretable importance scores that simplify models and enhance generalization. Tools such as SHAP improve explainability by attributing predictions to specific features, which enhances trust in AI systems. In data valuation, Shapley values equitably assess data contributions, optimizing resources for training. Challenges remain, such as high computational costs and interpretability barriers for non-experts. Still, developments in approximation methods and interpretability frameworks keep opening the scope for even greater applications. This places the Shapley value firmly in machine learning's tool kit.

2.5 Summary and Gaps Identified

Research Paper	**Advantage**	**Disadvantage**
Deep Multi-Task Learning with Relational Attention	High accuracy by learning multiple factors. Handles diverse data.	Requires large datasets. Increased complexity and computation.
Prediction of Employee Turn Over Using Random Forest Classifier with Intensive Optimized PCA Algorithm	High prediction accuracy and better generalization through PCA and Random Forest.	Increased computational complexity and reduced interpretability of PCA-transformed features.
A Machine Learning, Bias-Free Approach for Predicting Business Success Using Crunchbase Data	Promotes fairer and more accurate predictions by minimizing biases in business success analysis.	May require complex adjustments and additional data to ensure complete bias removal, impacting model simplicity.
The Explanation Game: Explaining Machine Learning Models Using Shapley Values	A uniform game formulation for Shapeley values is presented in the study, which enhances clarity and permits confidence intervals on attributes.	Shapley-value methods still face issues with selecting reference distributions, which can lead to misleading feature attributions.

Table 2.1: Comparison of Research Papers

2.5.1 Gaps Identified

1. **Interpretability Challenges**: While already-made methods such as Shapley values and The PCA-enhanced models are very robust in prediction, but often, they are not interpretable enough to help the end-users understand the reasoning behind the decisions.

2. **Bias Removal Limitations**: Bias-free methods are not completely free of all biases, particularly in historical or structural inequities within the datasets

3. **Computational Complexity**: The Random Forest with PCA and deep multi-task learning are very computationally intensive and not practical for real-time or resource-constrained applications.

4. **Scalability Issues**: Current models often struggle with scaling efficiently to larger datasets or more diverse data sources, thus reducing their effectiveness for broader industrial applications.

5. **Generalization Across Domains**: Most are optimized for a particular dataset or application, making it difficult to apply them in an industry or society without major reconfiguration.

Chapter 3

Requirements

3.1 Hardware and Software Requirements

3.1.1 Hardware Requirements

- **Processor:** Intel i5 or above / Apple M1 or higher
- **RAM:** Minimum 8 GB (16 GB recommended for model training and parallel data collection)
- **Storage:** At least 5 GB free space (for datasets, intermediate files, and cache)
- **Graphics:** GPU (optional, for faster model training – NVIDIA CUDA enabled preferred)

3.1.2 Software Requirements

- **Operating System:** Windows 10 / macOS / Linux
- **Programming Language:** Python 3.8 or above
- **Python Libraries:**
  - `pandas, numpy, scikit-learn` – for data preprocessing and standard ML utilities
  - `lightgbm, xgboost` – for building and training high-performance ML models
  - `googlemaps, requests` – for data collection using Google APIs
  - `joblib / pickle` – for model serialization and deserialization
- **Frontend:**

  - React.js (with integration of mapping tools such as Leaflet or Google Maps JavaScript API)

- **Backend:**
  - Node.js or Python (Flask / FastAPI) – for handling API requests and ML inference
  - Integration with serialized ML model (using `joblib` or `pickle`)
  - Dynamic data fetching from Google APIs upon user request

- **APIs and Services:**
  - Google Places API – to fetch restaurant and location details
  - Google Roads API – to calculate proximity to major roads
  - Google Routes API – to estimate traffic levels

- **Other Tools:**
  - Git – for version control and team collaboration
  - Jupyter Notebook / VS Code – for development and model experimentation

Chapter 4

System Architecture

The structure of the project is established using a System Architecture diagram which specifies the front-end, back-end, and APIs involved. Module details are included in Component Design. Algorithm Design details key algorithms where applicable. Data movement is depicted through Use Case Diagrams. Software and hardware requirements are described in this section. Dataset identified, module divisions, key deliverables break down the data and tasks and expected results. Finally, a project timeline schedules phases such as design, implementation, and testing.

4.1 System Overview

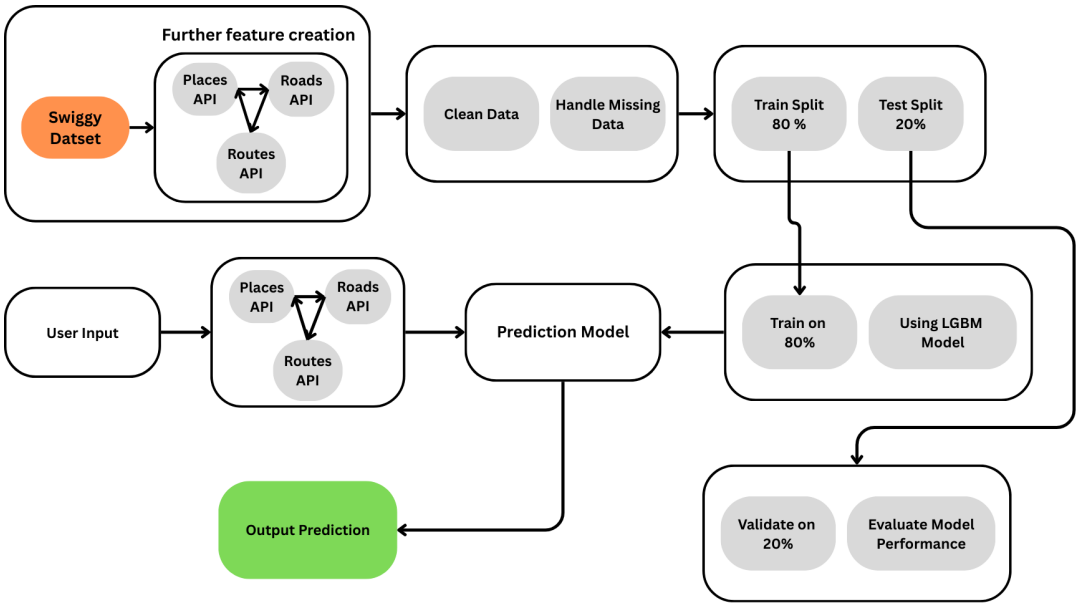


Figure 4.1: System Architecture

The architecture of our system can be broadly divided into two main components: (1) Dataset Collection and Model Training, and (2) Real-time Success Prediction via User

Input. Below is a step-by-step breakdown of each stage.

4.1.1 Part 1: Dataset Collection and Model Training

1. **Base Dataset:** We started with a publicly available Swiggy dataset from Kaggle, containing details of over 1.5 lakh restaurants across India. Each entry included information such as restaurant name, city, cuisine type, and average pricing.

2. **Parallel Data Enrichment:** Since enriching data using Google APIs is time-consuming, the dataset was split into 100 smaller subsets (approximately 1500 entries each) to allow parallel processing. Each subset was assigned for independent data fetching.

3. **Google API Integration:** For each restaurant, we used the name and city to fetch:

   - Ratings and rating count using the **Google Places API**
   - Latitude and longitude coordinates
   - Nearby buildings to estimate population density
   - Distance to the nearest road using the **Roads API**
   - Traffic conditions from the **Routes API**
   - Average price level of nearby restaurants

4. **Model Training:** After preprocessing the enriched data (handling missing values, normalizing features), we trained multiple models including **Random Forest**, **XGBoost**, and **LightGBM**. Based on evaluation metrics like Mean Squared Error (MSE), **LightGBM** emerged as the best-performing model.

4.1.2 Part 2: Real-Time Success Prediction via User Input

1. **Frontend Interface:** The user accesses a React-based frontend with an interactive map interface. They input:

   - The location (via map selection)
   - Desired cuisine type

- Estimated average pricing

2. **Backend Data Retrieval:** Once the user selects a location, the backend system:
   - Uses the latitude and longitude to call Google APIs
   - Fetches nearby places for population estimation
   - Retrieves traffic levels and road proximity
   - Computes average price levels of nearby restaurants
   - Calculates a competition score
3. **Prediction Generation:** This features is fed into the trained LightGBM model, which outputs a numerical **success score**. This score reflects the predicted success likelihood of opening a restaurant at the specified location with the chosen parameters.

This modular architecture allows for both robust offline model training and fast, real-time prediction based on dynamic user inputs.

4.2 Architectural Design

4.2.1 Sequence Diagram

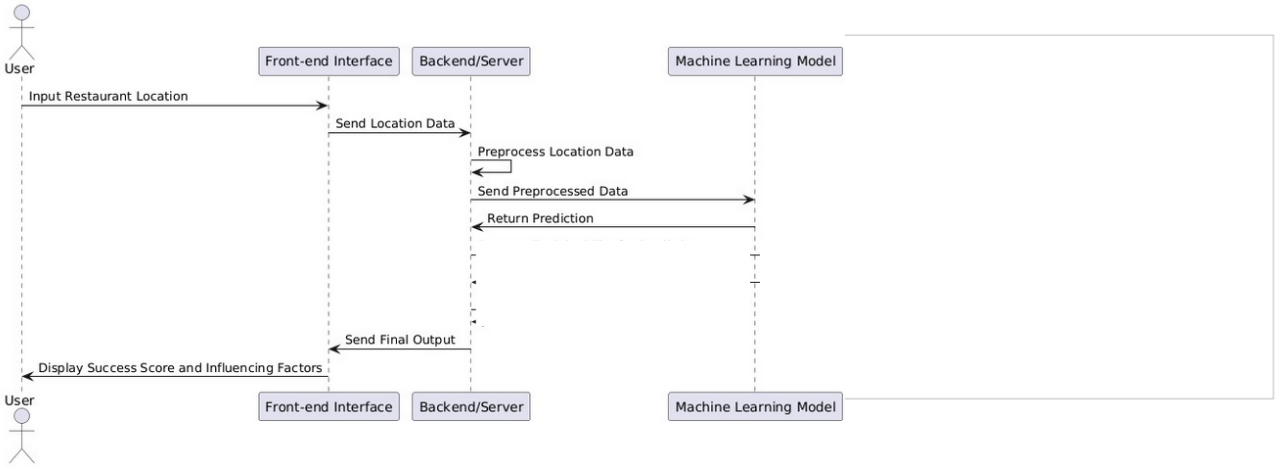


Figure 4.2: Sequence Diagram

4.3 Module Divisions

4.3.1 Module Divisions

Our system is organized into five major modules that work together to deliver accurate and real-time restaurant success predictions. Each module is responsible for a specific task, from collecting data to presenting the final prediction to the user.

1. Data Collection Module

This module gathers data from external sources. Google APIs such as Places, Roads, and Routes are used to fetch details like population indicators, road proximity, traffic conditions, nearby restaurant density, and price levels. To obtain ratings and coordinates, the city and restaurant name are added to this data.

2. Data Preprocessing Module

By removing duplicates and handling missing values, the acquired data is cleaned. Relevant features are engineered. The data is then normalized and structured for input into machine learning models.

3. Model Training Module

This module splits the data into training and testing sets. Multiple models, including Random Forest, XGBoost, and LightGBM, were trained and evaluated. LightGBM was selected as the final model due to its superior performance in predicting success scores based on features like location, cuisine, and local competition.

4. Prediction Module

When the user selects a location and inputs cuisine and price preferences, this module gathers real-time data using APIs. It then prepares the feature vector and uses the trained model to generate a success score for the chosen location.

5. Output Module

The final output, including the predicted success score and details like the number of nearby competitors, is displayed to the user. This helps in evaluating the feasibility of opening a restaurant at the selected spot.

4.4 Work Schedule-Gantt Chart

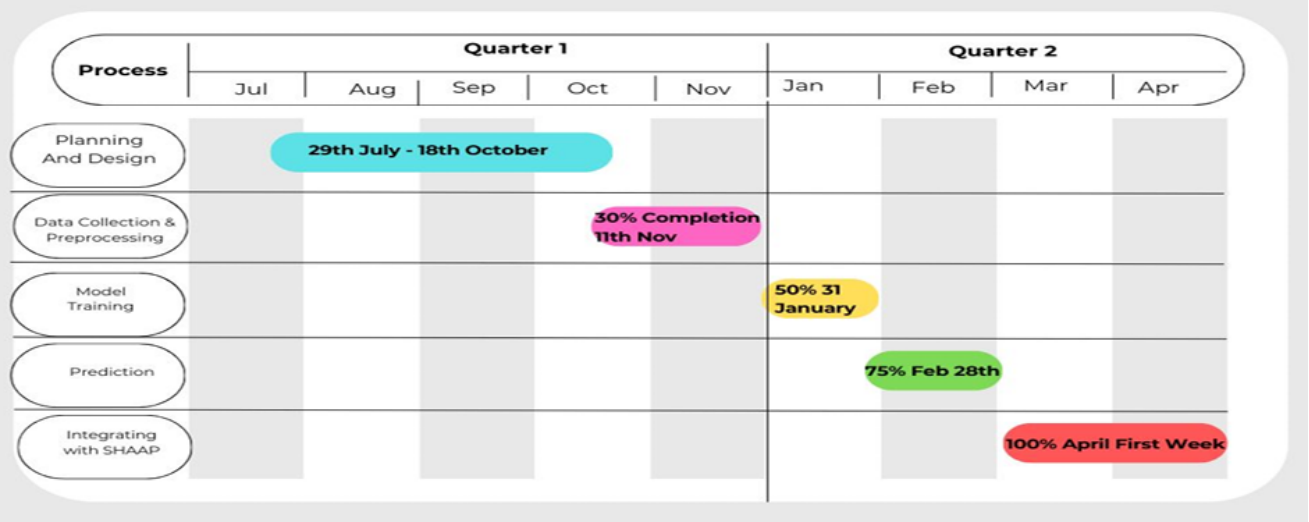


Figure 4.3: Gantt Chart

Chapter 5

System Implementation

5.1 Dataset Identified

The dataset was built using restaurant names and their cities to fetch key details like rating, rating count, latitude, and longitude through the Google Places API. With the coordinates, a nearby search was done to find surrounding buildings and estimate local population density. Additional data points like distance to the nearest road (via the Roads API), traffic levels (via the Routes API), and average price level of nearby restaurants were also collected. All these values were then stored to create a detailed dataset for analysis.

5.2 Proposed Methodology

5.2.1 Data Collection Module

The data acquisition process initiates with a structured dataset sourced from Swiggy via Kaggle, comprising extensive metadata on restaurants across India. Third-party APIs, namely Google's (Places, Roads, and Routes), are used to retrieve variables like geolocation locations, local mean price points, estimated population density, and traffic counts in order to give contextual depth to this foundation data. The data set is separated into smaller batches and processed in parallel to increase throughput and process the number of processes, resulting in more scalable and effective data enrichment.

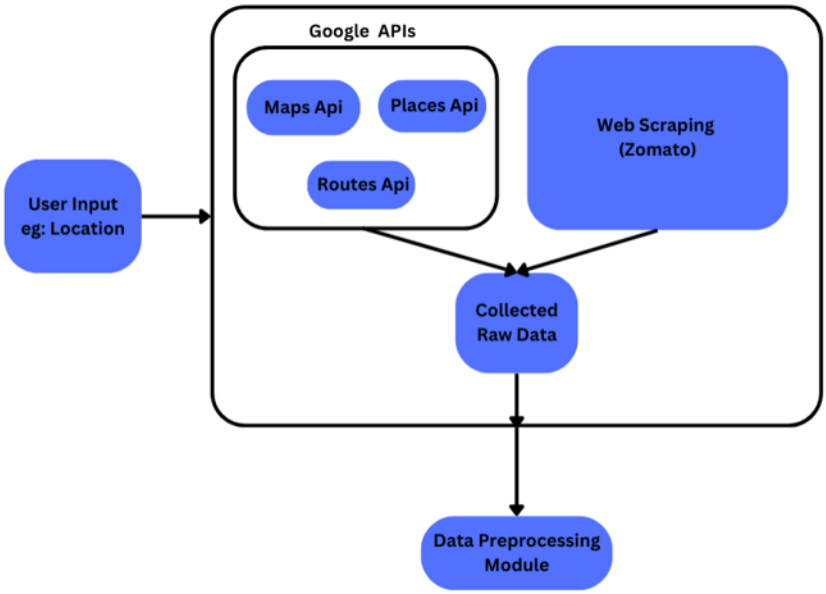


Figure 5.1: Data Collection

5.2.2 Data Preprocessing Module

Following collection, the data is cleaned and ready for model training. Missing values are appropriately handled, and records that are invalid or incomplete are removed. Important aspects are engineered, including competition scores, road distance, traffic level, and predicted population. To align the data with the model's input format for optimal performance during training and prediction, normalization and encoding are carried out.

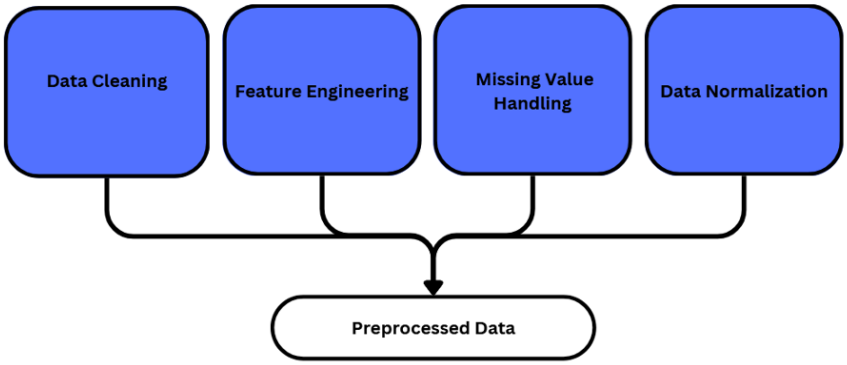


Figure 5.2: Data Preprocessing

5.2.3 Model Training Module

To determine which machine learning model performs the best, several models are tested, including Random Forest, XGBoost, and LightGBM. To obtain an objective estimate, the data is divided into training (80%) and testing (20%) sets. The models are trained to transfer input parameters, such as pricing, competition score, traffic rate, and population density, to a success score derived from actual restaurant ratings. Metrics like Mean Squared Error (MSE) and R-squared (R^2) are used to quantify performance. Out of all the models that were tested, LightGBM was chosen for final deployment because of its excellent accuracy, speed, and efficiency as well as its capacity to handle big datasets.

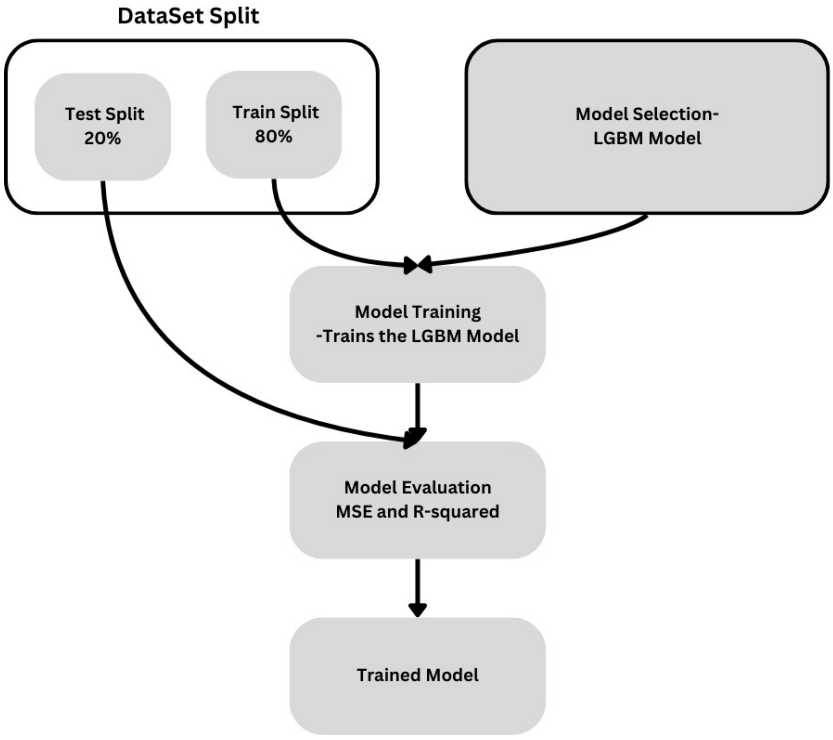


Figure 5.3: Model Training

5.2.4 Prediction Module

The prediction module allows users to estimate the success score of a new restaurant location in real time. When the user inputs a location, cuisine type, and average pricing, the system fetches live data using Google APIs—such as population density, traffic level, road proximity, and nearby competition. These values are processed and scaled to match the model's input format. The trained LightGBM model then predicts a success score,

which is displayed to the user along with nearby competitors.

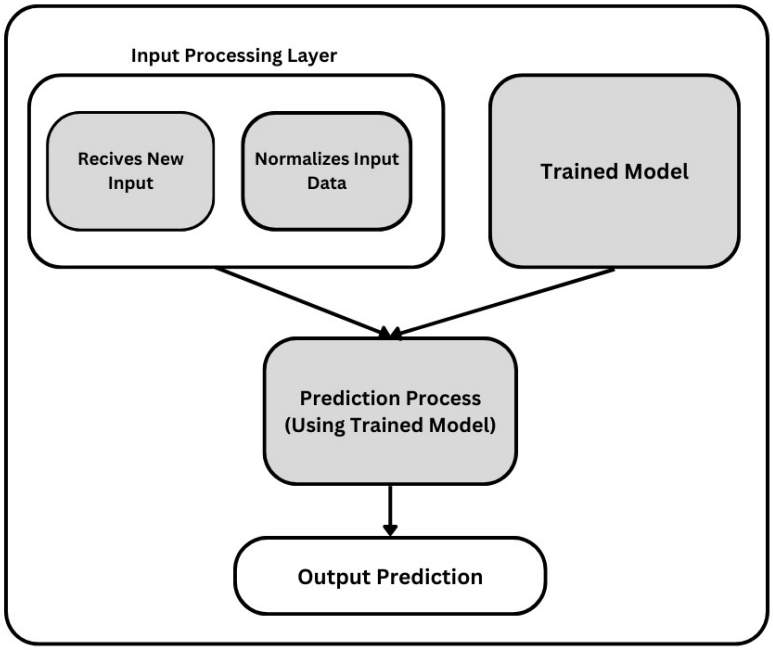


Figure 5.4: Prediction

5.3 User Interface Design

The MapMySuccess user interface is designed to be intuitive, informative, and interactive, allowing users to easily assess the success potential of new restaurant locations.

- **Layout:** The interface uses a two-column layout:
  - **Left Panel:**
    - Displays the predicted success score
    - Contains input fields for cuisine type and expected price range
    - Shows the clicked coordinates (latitude and longitude) when a user selects a location on the map
    - Lists nearby restaurants serving the selected cuisines, including their names and types
  - **Right Panel:**
    - Displays an interactive Google Map with a map/satellite toggle

* Allows users to select any location by clicking on the map
* Triggers real-time data collection and prediction based on the selected location

This interface supports a smooth user experience, helping restaurant owners make well-informed decisions based on data and visualization.

5.4 Implementation Strategies

The implementation of our restaurant success prediction system involves a combination of powerful Python libraries, APIs, and modern web development frameworks. The entire process is structured into two main parts: model development and deployment via an interactive frontend.

5.4.1 Dataset Collection

To enhance the restaurant dataset sourced from Kaggle (Swiggy data), we integrated multiple contextual features using the Google Places and Routes APIs.
The script iterates over each restaurant entry, retrieving missing location coordinates, user ratings, and rating counts. For deeper insight into the restaurant's surroundings, we calculated metrics such as average nearby population density, traffic severity, proximity to main roads, and average competitor pricing levels.
These enriched features were derived using real-time API responses, processed row-by-row, and saved into an updated dataset for further modeling. Care was taken to handle rate limits and ensure accurate data mapping.

swiggy.csv (45.66 MB)

Detail Compact Column 10 of 11 columns

id	name	city	rating	rating_count	cost	cuisine
567335	AB FOODS POINT	Abohar	--	Too Few Ratings	₹ 200	Beverages,Pizzas
531342	Janta Sweet House	Abohar	4.4	50+ ratings	₹ 200	Sweets,Bakery

Figure 5.5: Kaggle Dataset Card

To ensure consistent internet connectivity and seamless integration with the Google APIs, our data collection scripts were executed on the Google Colab platform. This setup allowed for smooth access to external services and parallelized data enrichment using multiple API calls.

	name	city	rating	ating_cour	cost	cuisine	address	latitude	longitude	pop_density	traffic_rte	visibility	avg_price_level
2	AB FOODS	Abohar	4.1	23	200	Beverages,Pizzas	AB FOODS	30.14049	74.206	3.65	3.425	6.962416	0.5
3	Janta Swe	Abohar	4	1958	200	Sweets,Bakery	Janta Swe	30.14257	74.19521	3.35	4.170833333	9.738254	2
4	theka coff	Abohar	4.1	1037	100	Beverages	theka coff	30.14963	74.20314	3.583333333	5.275	6.649747	1
5	Singh Hut	Abohar	4	31	250	Fast Food,Indian	Singh Hut,	30.14639	74.20388	3.55	4.283333333	13.24738	0.8
6	GRILL MA	Abohar	3.8	65	250	Italian-American,F	GRILL MA	30.14953	74.20299	3.6	3.5	1.883244	1
7	Sam Uncle	Abohar	4.3	26	200	Continental	Sam Uncle	30.14328	74.19929	3.25	3.983333333	9.795981	1
8	shere pun	Abohar	4	487	150	North Indian	shere pun	30.14075	74.19954	3.25	3.258333333	2.251434	1
9	Shri Balaji	Abohar	4.2	206	100	North Indian	Shri Balaji	30.21902	74.93992	3.533333333	4.033333333	11.17743	0
10	Hinglaj Ka	Abohar	4.6	7	100	Snacks,Chaat	Hinglaj Ka	30.1426	74.19503	3.366666667	3.9	5.096526	2
11	yummy hu	Abohar	3.5	6	200	Indian	yummy hu	30.14289	74.19577	3.3	4.1125	6.031842	2
12	CHAWLA S	Abohar	4.1	1037	300	Juices,Beverages	CHAWLA S	30.14963	74.20314	3.6	5.2	6.649747	1
13	Sethi Milk	Abohar	4.3	23	100	Sweets,Desserts	Sethi Milk	30.14574	74.21085	3.25	3.45	11.7363	2

Figure 5.6: Features after Data collection using Google APIs

On this we find a competition score for each restaurant by analyzing nearby restaurants within a 500-meter radius. It considers both total competitors and cuisine-specific competitors to generate a weighted score.

5.4.2 Data Pre Processing And True score calculation

To ensure data quality, we removed rows with invalid ratings (e.g., '–') and handled empty rating counts. Additionally, rating counts with 'K' suffixes were converted to numeric values for consistency and analysis.

True Score Calculation

To ensure high-quality input for our model, we first removed any rows with missing `rating` or `rating_count` values. After cleaning the data, we introduced a new feature called the **True Score** — an adjusted score that considers both the rating value and the number of users who rated the restaurant. This helps prevent skewed results from restaurants with very few ratings. The adjustment is done using an exponential confidence function, where the confidence increases as the number of ratings increases. The final results were saved into a new Excel file for use in modeling. After the cleaning and enrichment process, our final dataset contains a rich set of features that combine both internal characteristics and external contextual information of each restaurant. These include **cost** (average meal cost), **pop_density** (estimated nearby population density), **traffic_rate** (traffic severity in the area), and **visibility** (distance to the nearest main road). In addition, we derived **avg_price_level** from surrounding restaurants, along with **Same_type_restaurants_no** and **Total_no_restaurants** to capture local competition. From these, we computed a weighted **Comp_Score** to reflect competitive pressure. Finally, we calculated a **true_score**, which adjusts the raw rating using an exponential confidence formula based on the number of ratings. Together, these features provide a comprehensive view for modeling restaurant success.

Data Augmentation for Low True Score Samples

To address potential data imbalance and enhance model robustness, we applied data augmentation specifically targeting restaurants with a low predicted `true_score` (less than 3). These underperforming samples may be underrepresented in the dataset and yet crucial for learning the characteristics of low-success regions.

For each low-scoring record, we synthetically generated three additional samples by introducing controlled random noise to key numerical features. Features such as population density, traffic rate, visibility, and competition score were varied slightly using uniform multiplicative noise (within 10%), while the average price level was nudged within a 5% range. The cost feature was also perturbed using a small random integer shift. This approach simulates plausible variations in real-world conditions without altering the core profile of the restaurant.

The augmented samples were then concatenated back with the original dataset, resulting in a more balanced and enriched dataset. This augmentation technique helps the model generalize better, especially in low-success prediction scenarios.

5.4.3 Model Training

Once the dataset was enriched with contextual and competitive features, we proceeded with model training to predict restaurant success. The target variable used was the **true_score**, which reflects a confidence-weighted version of user ratings. We experimented with multiple machine learning models including **Random Forest**, **XGBoost**, and **LightGBM**. These models were trained using the cleaned dataset with features such as cost, population density, traffic rate, visibility, competition scores, and price levels. After extensive experimentation and hyperparameter tuning, **LightGBM** outperformed the others in terms of both accuracy and training time, making it our final model of choice. The model was evaluated using standard metrics such as RMSE and R^2, and it demonstrated strong predictive performance on the test set.

Random Forest Training

We trained a Random Forest Regressor using Optuna for hyperparameter tuning. Key features were scaled using MinMaxScaler, and the model was evaluated using RMSE and R^2 score. The best-performing model and scaler were saved using `joblib` for future use.

XGBoost Training

Then we employed XGBoost for regression, leveraging Optuna to tune hyperparameters like learning rate, max depth, and regularization terms. MinMaxScaler was used for feature scaling, and early stopping was enabled to avoid overfitting. .

LightGBM Training

We trained a LightGBM regressor to predict restaurant success using the engineered features. Hyperparameter tuning was performed using Optuna, optimizing parameters such as learning rate, max depth, and regularization terms. Feature scaling was applied using MinMaxScaler. The model was trained with early stopping enabled to avoid overfitting.

After training, the optimized model and scaler were saved. LightGBM provided competitive results with low RMSE and a high R^2 score.

5.4.4 Model Prediction via Interactive Map Interface

When a user clicks on a point on the interactive React-based map, inputs such as the latitude, longitude, selected cuisine type, and the average price are captured from the frontend. These geographic coordinates are sent to the backend, where they serve as the starting point for feature extraction using the Google Places API. Specifically, we retrieve contextual features such as nearby population density, traffic routes, visibility score (proximity to main roads), number of nearby restaurants of the same type, total restaurant count in the vicinity, and a competition score derived from the ratings and review counts of neighboring competitors.

The collected data is then merged with static user inputs like average price and cuisine type, and passed through the same preprocessing pipeline used during model training (e.g., feature scaling with the previously fitted `MinMaxScaler`). Once the final feature vector is prepared, it is fed into the best-performing model—`LightGBM`, which was previously trained and optimized using Optuna for hyperparameter tuning and saved as a pickle file. The model returns a predicted **True Score**, which reflects the potential success of opening a restaurant at that specific location, considering both intrinsic restaurant features and external environmental factors. This forecast is then transmitted back to the frontend and visually presented to the user in real-time.

Chapter 6

Results and Discussions

This section shows the outcome of our machine learning approach to local restaurant success prediction. Determining how well our system can comprehend real location values and translate them into a successful rating was the primary objective.

Using an improved dataset that included real-time data on things like population density, traffic, visibility, price range, and competition proximity, we tested three distinct models: Random Forest, XGBoost, and LightGBM. After much fine-tuning, LightGBM emerged as the best, surpassing the others in accuracy and speed.

Performance measures like Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2) were utilized to gauge model performance. The tuned LightGBM model showed low error predictions and a high R^2 value, which verified its efficacy to detect underlying patterns in data. Besides learning, the model was also incorporated in an internet-based system with which users can input a place, kind of food, and price. In return, the system dynamically fetches local information in real-time using Google APIs, processes them, and forecasts a success score within seconds. Such real-time feedback provides a valuable tool for business planning and risk analysis.

6.1 Overview

The backend of this application was developed using **Flask**, a lightweight Python web framework ideal for building REST APIs and serving machine learning models. On the frontend, **ReactJS** was used to create a dynamic, interactive user interface that allows users to select locations on a map and input restaurant details.

Some of the technologies used include:

- **LightGBM**: A gradient boosting framework that provided the most accurate predictions during model evaluation.

- **pandas & numpy**: Used for data cleaning, manipulation, and numerical computations.
- **Flask**: Handled the backend logic, model inference, and API integration.
- **ReactJS**: Powered the frontend with an interactive map interface for selecting restaurant locations.
- **Google Maps APIs**: Used to fetch real-time data such as traffic, nearby restaurants, and population density.

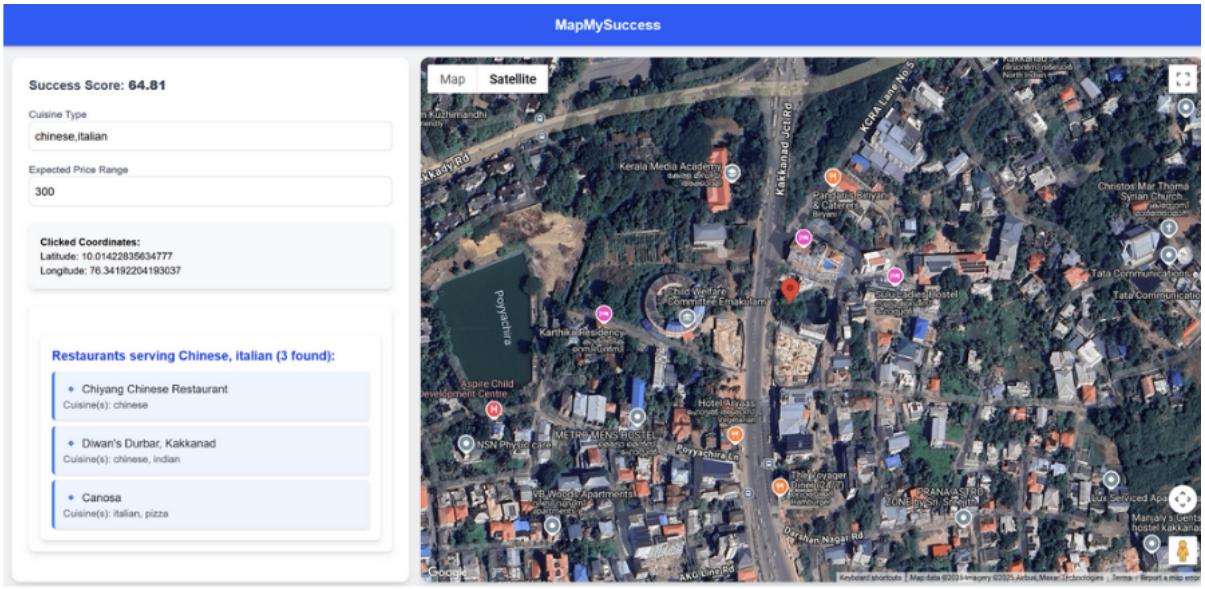


Figure 6.1: User Interface

6.2 Quantitative Result

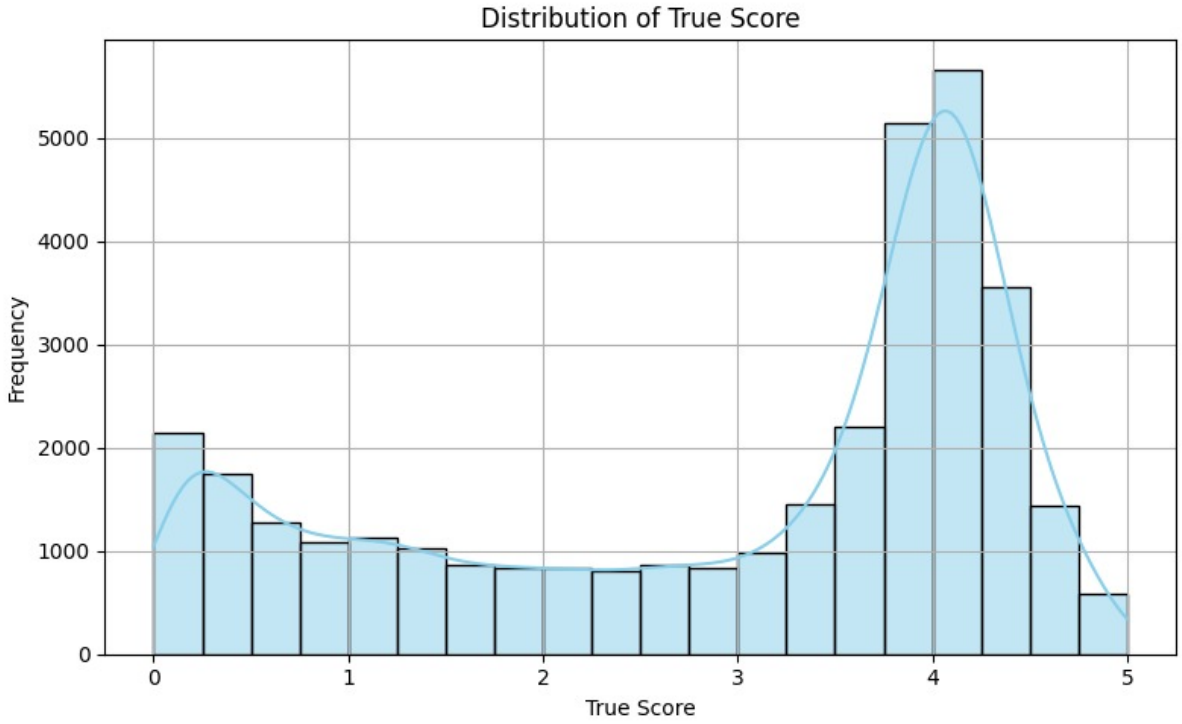


Figure 6.2: Before Augmentation

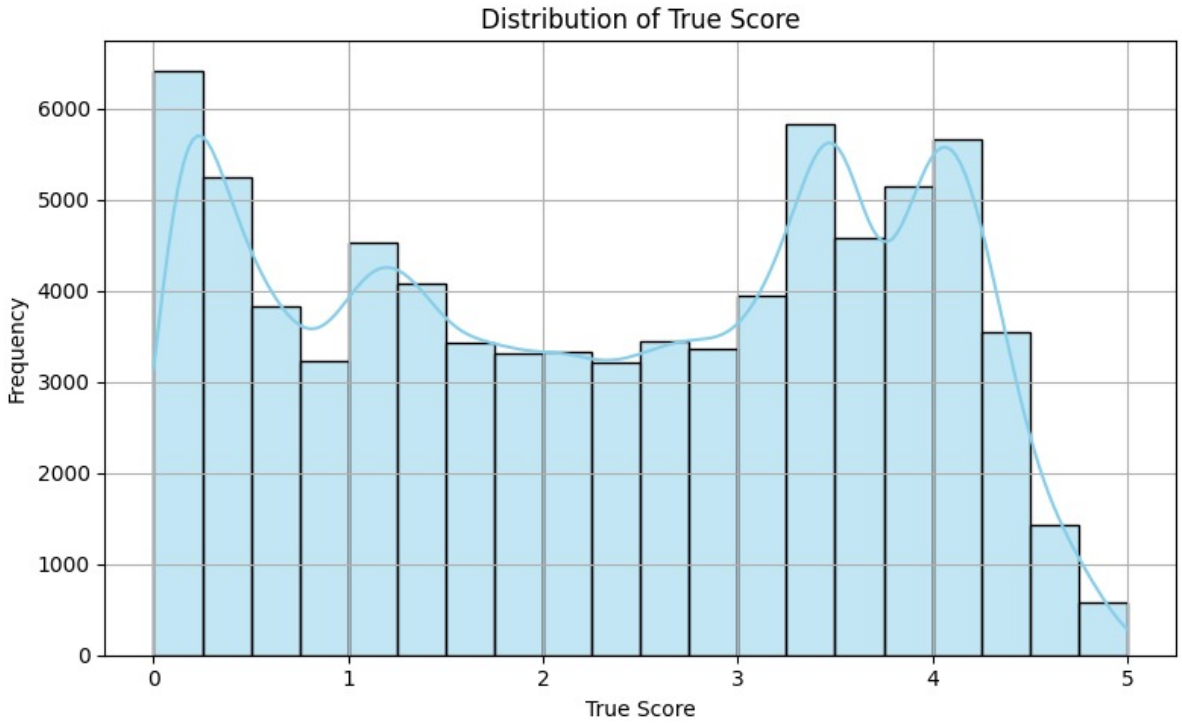


Figure 6.3: After Augmentation

6.2.1 LightGBM

The LGBM model was trained on a dataset with the number of features such as cost, population density, traffic rate, visibility, average price level, and competition score to predict the actual success score for a restaurant location. The dataset was MinMax scaled before being used to train the model, and Optuna was used to optimize the hyperparameters in 50 optimization trials.

A satisfactory fit between actual and projected values is indicated by the best model's Mean Squared Error (MSE) of 1.3539, Root Mean Squared Error (RMSE) of 1.1636, and R-squared value of 0.3827 during testing. These metrics show that, particularly after feature improvement, the model provided more accurate prediction values and did a very good job of identifying the underlying patterns in the data. The model's ability to explain a significant portion of the variance in success ratings from the input features is demonstrated by its R-squared of 38.27%. This experiment demonstrated that the predictive accuracy of the model for location-based success analysis was greatly improved by Optuna-based auto hyperparameter tuning in conjunction with feature scaling and augmentation.

```
Optimized LGBM model and scaler saved.
Optimized Mean Squared Error: 2.0012992721116314
Optimized Root Mean Squared Error: 1.4146728498531493
Optimized R-squared Score: 0.07569887176723156
```

Figure 6.4: Performance

6.2.2 XGBoost

Additionally, we used the same location-based data to train the XGBoost model, using competition score, average price level, visibility, traffic rate, population density, and cost as features. 50 trials were carried out using Optuna with the aim of hyperparameter search, and input features were scaled to the normal feature space using MinMaxScaler.

After optimization, the model's Mean Squared Error (MSE) was 1.3651, its Root Mean Squared Error (RMSE) was 1.1684, and its R-squared was 0.3775. All the above outcomes indicate that the model performed well in estimating the true scores. Although the RMSE and MSE values were close to the optimized LGBM model, the XGBoost model achieved a very low R-squared value, indicating that it predicted 37.75% variance in the target

variable.

This type of performance shows that tree models can assess multifactorial interactions in restaurant success prediction, especially when well optimized using automated software like Optuna. The XGBoost model generalized and stabilized well, even though it did not have the best R-squared score, especially when not using augmented data.

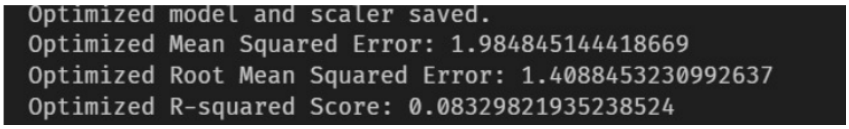


Figure 6.5: Performance

6.2.3 LightGBM with Augmented Data

A dataset containing the most crucial characteristics—cost, population density, traffic rate, visibility, average price level, and competition score—was used to train the LightGBM regression model. Optuna was utilized to adjust hyperparameters across 50 trials in order to minimize mean squared error after data preparation using MinMaxScaler and an 80-20 train-test split. The final model was trained using the optimal hyperparameters that Optuna had chosen, and it performed well in terms of prediction. It obtained an R-squared score of 0.7811, a mean squared error of 0.4836, and a root mean squared error of 0.6954. The R^2 score shows that the model explains about 78.11% of the true score variance effectively, which is an indicator of high model reliability. These results validate that the combination of the LightGBM algorithm, hyperparameter optimization, and data augmentation significantly enhances the ability of the model to predict restaurant success potential from location-based features.

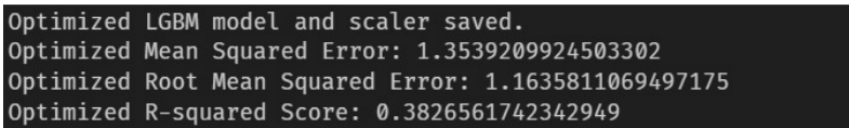


Figure 6.6: Performance

6.2.4 XGBoost with Augmented Data

An enhanced version of the original data set was used to train the XGBoost model, which increased the number of data points and enhanced the model's capacity for generalization.

Cost, population density, traffic rate, visibility, average price level, and competition score were among the data set's characteristics. MinMaxScaler was utilized for feature scaling, while Optuna was utilized to adjust the model's hyperparameters across 50 trials. The model produced an R-squared score of 0.8192, a mean squared error of 0.4139, and a root mean squared error of 0.6434 after training. The results demonstrate improved performance over earlier models.Actually, the model's R2 score of 81.92% indicates that it performs exceptionally well in terms of prediction and can explain a significant portion of the variance in the target variable. This demonstrates that the best way to accurately forecast the success potential of restaurant locations is to employ data augmentation using XGBoost in conjunction with Optuna-based hyperparameter adjustment.

```
Optimized model and scaler saved.
Optimized Mean Squared Error: 1.3651221858185985
Optimized Root Mean Squared Error: 1.16838443408777
Optimized R-squared Score: 0.37754879529145613
```

Figure 6.7: Performance

6.2.5 Random Forest with Augmented Data

Cost, population density, traffic rate, visibility, average price level, and competition score were among the features that were used to train the Random Forest model on the original data set. Optuna was used to adjust the model's hyperparameters, such as the number of estimators, maximum depth, and minimum samples per split and leaf, after MinMaxScaler was used to normalize the input data and split it into training and test sets in 80-20 ratios. In order to determine the optimal set of parameters, Optuna minimized the negative R-squared score for over 50 optimization runs. Following training on the scaled training data, the optimal Random Forest model produced an R-squared score of 0.7103, a mean squared error of 0.6219, and a root mean squared error of 0.7886. This indicates that the model has a fairly good predictive power to evaluate the viability of restaurant locations, explaining roughly 71.03% of the variance in the genuine success scores.

```
Optimized RF model and scaler saved.
Optimized Mean Squared Error: 1.9684610008679293
Optimized Root Mean Squared Error: 1.4030185319046677
Optimized R-squared Score: 0.09086524472440638
```

Figure 6.8: Performance

6.2.6 Summary

In this chapter, we used Optuna for hyperparameter tweaking and compared the effectiveness of the Random Forest, LightGBM, and XGBoost models for predicting restaurant site success. LightGBM performed better in terms of prediction than the other models, as evidenced by its highest R2 score. While both Random Forest and XGBoost performed well, their accuracy and efficiency were not as well-balanced. The training and test scores were nearly the same, and all of the models had very little overfitting. These outcomes demonstrate that LightGBM is the best model for this job due to its high accuracy and good generalization.

Chapter 7

Conclusions & Future Scope

The capacity of machine learning algorithms to determine the feasibility of possible restaurant locations based on important factors including cost, population density, traffic, visibility, competition, and price is demonstrated by this project. Creating a system that is accurate, comprehensible, and actionable for real-world decision-making in the food service industry was the primary goal. After extensive testing and hyperparameter adjustment using Optuna, LightGBM outperformed Random Forest and XGBoost as the most accurate and well-balanced model. Most significantly, all models showed excellent generalization with very little overfitting, demonstrating their dependability with unknown data.

The project's scope can be increased in the future by including more specific data, including foot traffic statistics, social media sentiment, or current market trends. Enhancing the model's interpretability through explainable AI tools like SHAP will help restaurant managers better understand the reasoning behind each prediction. Moreover, deploying this system as a user-friendly web application could empower business owners to make data-driven location decisions without needing deep technical knowledge. Future development could also focus on incorporating geospatial analysis and mapping tools to visually guide users in identifying strategic areas for expansion based on location-specific insights and competition clustering.

References

[1] A. B. Wild Ali, "Prediction of employee turn over using random forest classifier with intensive optimized pca algorithm," *Wireless Personal Communications*, vol. 119, no. 4, pp. 3365–3382, 2021. [Online]. Available: https://doi.org/10.1007/s11277-021-08408-0

[2] K. Żbikowski and P. Antosiuk, "A machine learning, bias-free approach for predicting business success using crunchbase data," *Information Processing Management*, vol. 58, no. 4, p. 102555, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457321000595

[3] B. Rozemberczki, L. Watson, P. Bayer, H.-T. Yang, O. Kiss, S. Nilsson, and R. Sarkar, "The shapley value in machine learning," 2022. [Online]. Available: https://arxiv.org/abs/2202.05594

Appendix A: Presentation

MapMySuccess

Guide: Ms. Ann Grace
Asst. Professor
Dept. of CSE

Rahul Varghese U2103169
Rebecca Liz Punnoose U2103171
Richu Kurian U2103175
Rohan Joseph Arun U2103180

CONTENTS

PROBLEM DEFINITION

Choosing the right location is critical for a restaurant's success, but many owners struggle to predict how well a spot will perform. This leads to poor choices, wasted investments, and business failures. A tool is needed to help restaurant owners make data-driven decisions about location potential and success.

PURPOSE AND NEED

- Provide restaurant owners with a tool to predict the success potential of new locations.
- Evaluate key factors such as population density, price level and nearby competitors.
- Help owners make informed, data-driven decisions to increase profitability.

OBJECTIVE

To develop a machine learning model that predicts the success rate and expected revenue of potential restaurant locations by analyzing key factors such as population density, parking availability, visibility, and nearby competitors, thereby enabling restaurant owners to make informed, data-driven decisions for optimal site selection.

LITERATURE SURVEY

Research Paper	Advantage	Disadvantage
The Explanation Game: Explaining Machine Learning Models using Shapley values; L Merrick et al.	The paper introduces a unified game formulation for Shapley values, improving clarity and allowing for confidence intervals on attributions.	Shapley-value methods still face issues with selecting reference distributions, which can lead to misleading feature attributions.
A machine learning, bias-free approach for predicting business success using Crunchbase data; K Żbikowski et al.	The paper uses comprehensive data preprocessing ensuring that the data is well-prepared for model training.	Using methods like cross-validation on large datasets, while thorough, can be computationally expensive and time-consuming, potentially limiting scalability.

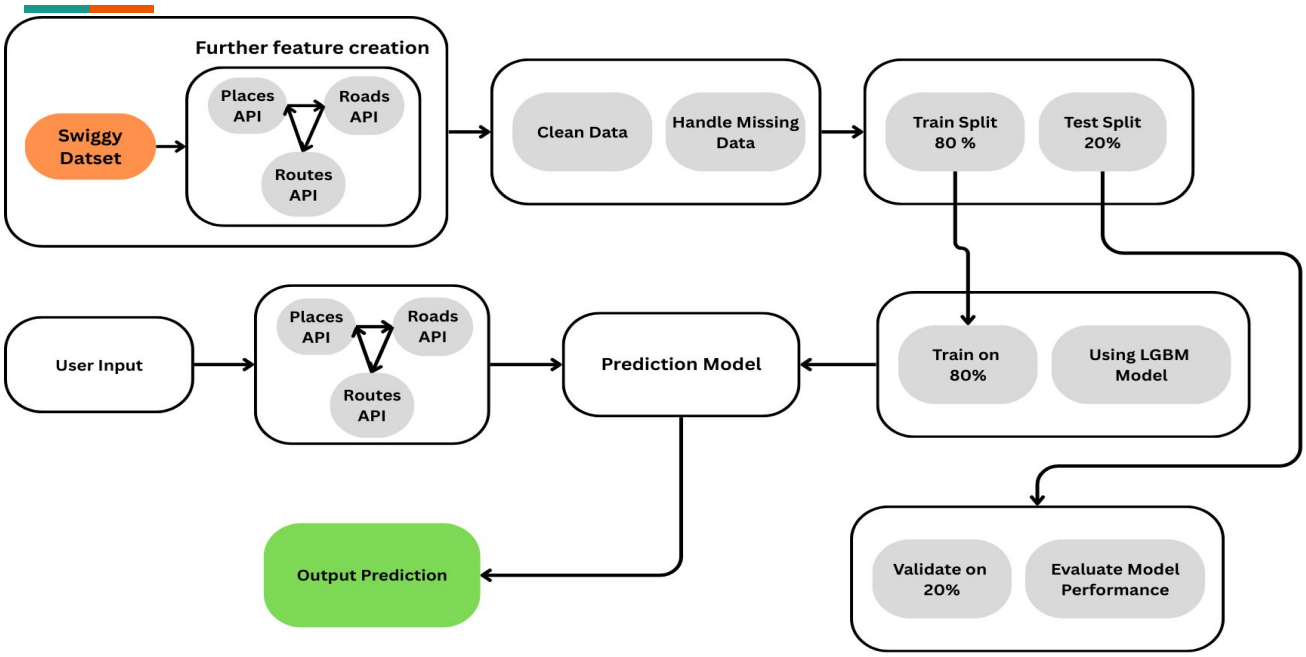
LITERATURE SURVEY

Research Paper	Advantage	Disadvantage
Prediction of Employee Turn Over Using Random Forest Classifier with Intensive Optimized Pca Algorithm;AB Wild Ali et al.	High prediction accuracy and better generalization through PCA and Random Forest.	Increased computational complexity and reduced interpretability of PCA-transformed features.
Deep Multi-Task Learning with Relational Attention;J Zhao et al.	High accuracy by learning multiple factors. Handles diverse data.	Requires large datasets. Increased complexity and computation.

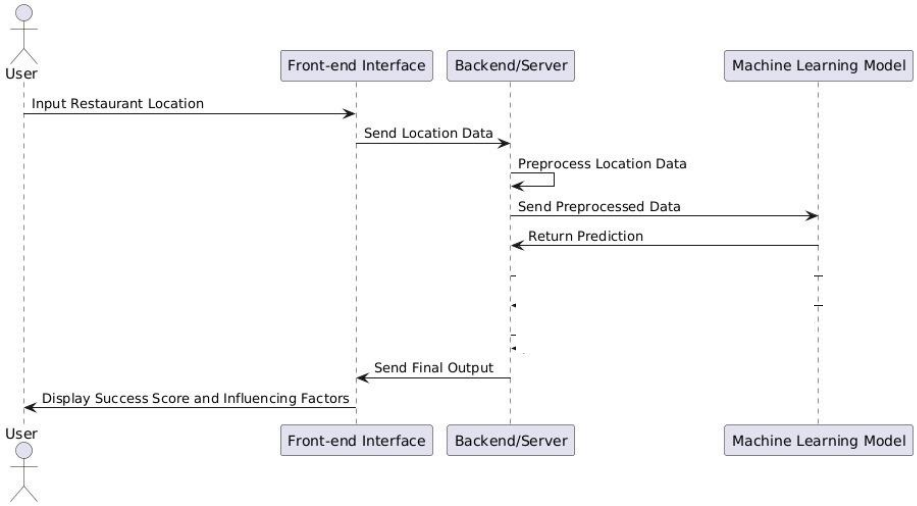
PROPOSED METHOD

- Data Collection and Preprocessing
- Feature Extraction
- Model Training and Validation
- Prediction and Analysis

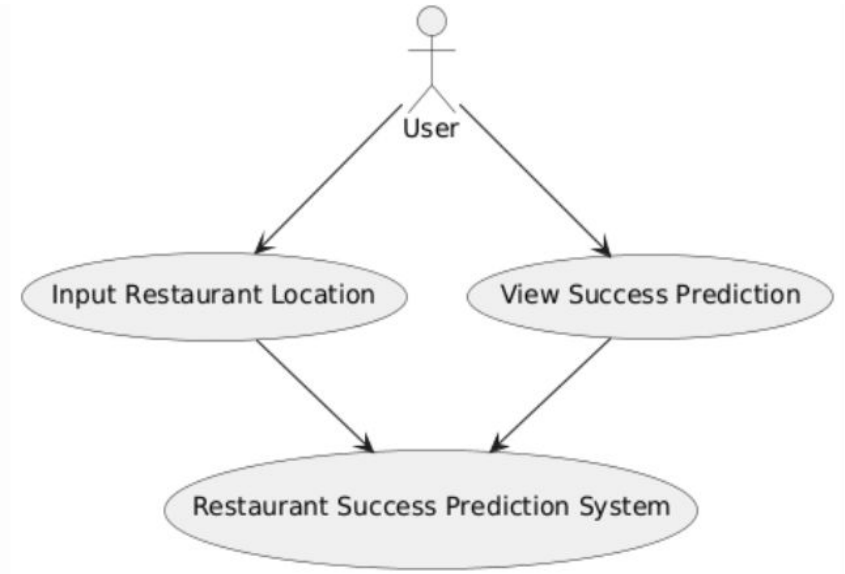
ARCHITECTURAL DIAGRAM



SEQUENCE DIAGRAM



USE CASE DIAGRAM



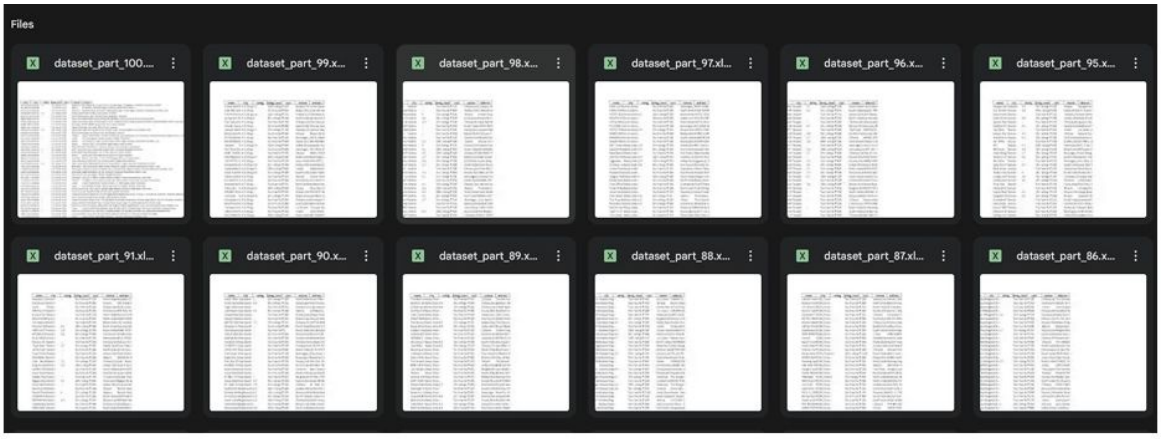
METHODOLOGY

1. Data Collection Module
2. Data Preprocessing Module
3. Model Training Module
4. Prediction Module

DATA COLLECTION MODULE

1.Collected Swiggy data with details on 1.5 lakh restaurants (cuisine, avg. price).

2.Split into 100 smaller sets for faster, parallel processing.



DATA COLLECTION MODULE

3. Used Google Places API to get ratings, review counts, and exact location.

	A	B	C	D	E	F	G
1	Name	Address	Rating	r Ratings T	Latitude	Longitude	
2	Dessi Cuppa Cusat	Unversity of Science	4	1357	10.04655	76.32177	
3	Milma Shoppee	28RH+MF3, South Ka	4.2	58	10.04164	76.32867	
4	Nosh Haus | Metro Pi	Thomas Aykareth Sc	4.4	1480	10.08896	76.34355	
5	Navya Bakers, Edapal	Edappally Toll, Junct	4	323	10.02829	76.31042	
6	The Coffee Cup	Lulu Shopping Mall,	4.3	484	10.02765	76.30826	
7	Costa Coffee	Lulu Mall, Junction, I	4.2	2007	10.027	76.30814	
8	GVQ Time Cafe	GVQ Time Cafe, Kuzl	4.5	518	10.06482	76.32296	
9	Krishnapooja thattuk	SGR Tower, 2nd Floc	5	10	10.06339	76.32255	
10	GRAND CAFE	5/257A,Pallathu Squ	5	4	10.06555	76.32115	
11	CHAI CUP	Puthiya Road Signal	4.4	5	10.06729	76.3228	
12	OCB Cafe	388C+WXP, East Eloc	3.6	17	10.06733	76.32245	
13	Quality Bakers	Municipal complex,	N/A	N/A	10.06181	76.32093	
14	Foodju	386C+3WH, opposit	3.7	32	10.06018	76.32236	
15	CAFE 3.0	Metro pillar No : 255	5	7	10.0598	76.32223	
16	CP FRUITS SHAKE PAL	Metro Pillar 256, Koc	4.4	13	10.05969	76.32214	
17	K Town Classic Cafe	3, Bisho Nagar Aven	4.2	145	10.05829	76.32158	
18	Chai Truck	389H+HW, Choornik	3.5	13	10.06894	76.32978	
19	Tea Talk	389J+H56, Choornikl	4.5	16	10.06901	76.33035	

DATA COLLECTION MODULE

4.Identified nearby buildings to estimate population in the area.

5.Calculated distance to nearest road using Roads API.

6.Assessed local traffic using Routes API.

7.Found average price levels of nearby restaurants.

8.Computed competition score based on nearby restaurants of the same cuisine.

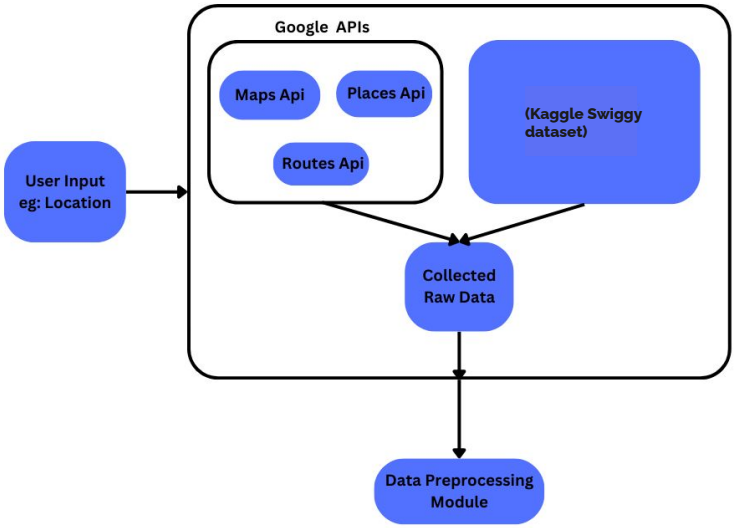
```
Unique Place Categories within 1 km: ['locality', 'point_of_interest', 'atm', 'restaurant', 'lo
dging', 'parking', 'supermarket', 'pharmacy', 'store', 'meal_takeaway', 'clothing_store', 'phys
iotherapist', 'university', 'hair_care', 'car_repair', 'cafe', 'laundry', 'insurance_agency']
Length of places list: 18
List of numeric values (population densities) for nearby places: [4, 3, 4, 3, 3, 3, 3, 3, 4, 3,
 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 5, 4, 3, 3, 3, 4, 3, 4, 3, 3, 3, 3, 5, 3, 3, 4, 3, 4, 3, 3, 4, 3
, 3, 3, 4, 4, 3, 4, 4, 4, 3, 4, 3, 3, 4, 3, 3, 3, 3, 3]

---------------

Average Population Density: 3.4
Traffic Severity (1-5): 1
Distance to Nearest Main Road (meters): 35.80120001234243
Competitor Presence for South indian restaurant Restaurants: 1/5

_____________________________________________________________________

['Aryaas', 9.99810742938937, 76.3614687596429, '1-200', 3.4, 1, 35.80120001234243, 1]
```


DATA COLLECTION MODULE

Dataset structure:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
name	city	rating	rating_count	cost	cuisine	address	latitude	longitude	pop_density	traffic_rte	visibility	avg_price_level	Same_type_restaurants_no	Total_no_restaurants	Comp_Score	true_score
AB FOODS	Abohar	4.1	23	200	Beverages	AB FOODS	30.14049	74.206	3.65	3.425	6.962416	0.5	1	10	6.913	1.5117371
Janta Swe	Abohar	4	1958	200	Sweets,Ba	Janta Swe	30.14257	74.19521	3.35	4.170833	9.738254	2	1	17	11.512	4
theka coff	Abohar	4.1	1037	100	Beverages	theka coff	30.14963	74.20314	3.58333333	5.275	6.649747	1	1	12	8.227	4.1
Singh Hut	Abohar	4	31	250	Fast Food,	Singh Hut,	30.14639	74.20388	3.55	4.283333	13.24738	0.8	1	14	9.541	1.8482222
GRILL MAS	Abohar	3.8	65	250	Italian-Am	GRILL MAS	30.14953	74.20299	3.6	3.5	1.883244	1	1	12	8.227	2.7643792
Sam Uncle	Abohar	4.3	26	200	Continenta	Sam Uncle	30.14328	74.19929	3.25	3.983333	9.795981	1	1	13	8.884	1.7435616
shere punj	Abohar	4	487	150	North Indi	shere punj	30.14075	74.19954	3.25	3.258333	2.251434	1	3	12	8.913	3.9997645
Shri Balaji	Abohar	4.2	206	100	North Indi	Shri Balaji	30.21902	74.93992	3.53333333	4.033333	11.17743	0	1	1	1	4.131773
Hinglaj Kac	Abohar	4.6	7	100	Snacks,Cha	Hinglaj Kac	30.1426	74.19503	3.36666667	3.9	5.096526	2	1	17	11.512	0.6009521
yummy hu	Abohar	3.5	6	200	Indian	yummy hu	30.14289	74.19577	3.3	4.1125	6.031842	2	1	16	10.855	0.3957785
CHAWLA S	Abohar	4.1	1037	300	Juices,Bev	CHAWLA S	30.14963	74.20314	3.6	5.2	6.649747	1	1	12	8.227	4.1
Sethi Milk	Abohar	4.3	23	100	Sweets,De	Sethi Milk	30.14574	74.21085	3.25	3.45	11.7363	2	1	2	1.657	1.5854803
Swastik Dł	Abohar	4.1	1476	200	North Indi	Swastik Dł	30.14037	74.19507	3.35	3.408333	11.23514	1.333333333	3	14	10.227	4.1
Jodhpuri K	Abohar	2	2	100	Snacks	Jodhpuri K	30.14277	74.1959	3.3	3.991667	5.759013	2	1	16	10.855	0.0784211
Bharawan	Abohar	4.1	1476	300	Indian	Bharawan	30.13224	74.206	3.53333333	1.9625	11.26197	1	2	7	5.285	4.1
Tandoori N	Abohar	3.8	89	300	Tandoor	Tandoori N	30.14621	74.20378	3.5	4.225	3.686665	0.8	1	14	9.541	3.159175
Roll Expre	Abohar	4.4	249	200	Fast Food	Roll Expre	30.13293	74.20571	3.55	2.066667	12.94085	1.2	1	8	5.599	4.3697541
wah ji waa	Abohar	3.9	104	200	North Indi	wah ji waa	30.14078	74.20626	3.55	2.408333	3.198268	0.285714286	1	10	6.913	3.4127722
FOODY M(	Abohar	4.1	23	300	Fast Food,	FOODY M(	30.14049	74.206	3.61666667	3.425	6.962416	0.5	1	10	6.913	1.5117371
PUNJABI T	Abohar	4.1	1476	700	Punjabi,Nc	PUNJABI T	30.13224	74.206	3.56666667	1.9625	11.26197	1	1	7	4.942	4.1
PUNJABI T	Abohar	4	1476	650	North Indi	PUNJABI T	30.14075	74.19954	3.25	3.270833	2.251434	1	1	12	8.227	4

DATA COLLECTION MODULE



DATA PREPROCESSING MODULE

1. Data Cleaning: Remove irrelevant or duplicate entries and outliers from the collected data.
2. Feature Engineering: Create new features from the raw data.
3. Missing Value Handling: Address incomplete data using methods like imputation or removal.
4. Data Normalization: Scale features such as population and parking to ensure consistent data ranges for improved model performance.
5. Combine Processed Data: The cleaned and processed data is then passed to the Model Training Module for further analysis.

DATA PREPROCESSING MODULE

Encountered duplicate values in the data, highlighting the need for a data-cleaning process.

148	Jibin Coffee House	4963+3H	N/A		N/A	10.1101	76.3539
149	Tely Cafe	489V+4(		4.5	31	10.1178	76.3438
150	Thoppi Salims,cool palace	Market, F	N/A		N/A	10.1102	76.355
151	De Cafe Canopy	near Chi		4.1	2305	10.1023	76.3488
152	Gopu's Coolbar	Kochi, B		4.5	74	10.1131	76.3547
153	Cinnamor	Aysha B		4.7	51	10.1183	76.3499
154	Hotel Coffee House	India	N/A		N/A	10.1054	76.3544
155	Vigneswara sweets and co	Manappı		4	6	10.117	76.3533
156	Baby's Bakery, Nazreth Ro	4933+QM		4.6	9	10.1044	76.3542
157	Rappichas Tea Stall	KSRTC,		4.8	4	10.1066	76.3558
158	Tea Shop	opposite	N/A		N/A	10.1071	76.3562
159	Tea Trolly	482X+27	N/A		N/A	10.1001	76.3482
160	Coffee shop	48CR+2F		5	3	10.12	76.3412
161	Hot n Cold	Sub Jail		3	1	10.1062	76.3574
162	TEA N ME	Mulanga		4	5	10.099	76.3477
163	Amana's Coffee House	4955+46	N/A		N/A	10.1078	76.3581
164	Thaavalam cafe	38XW+F		3.9	9	10.0986	76.3474
165	Kerala Cafe	487M+8(		3.7	21	10.1133	76.3345
166	Milma Booth	39X3+R(		4.6	9	10.0995	76.3536
167	Kuzhivelipady	39X2+C(	N/A		N/A	10.0985	76.3519
168	Canteen, WaterTreatment	4956+R4		5	1	10.1096	76.3603

DATA PREPROCESSING MODULE

Target Variable:

- Bayesian average approach
- If rating count is low, score is reduced.
- If rating count is high, score closer to actual rating.

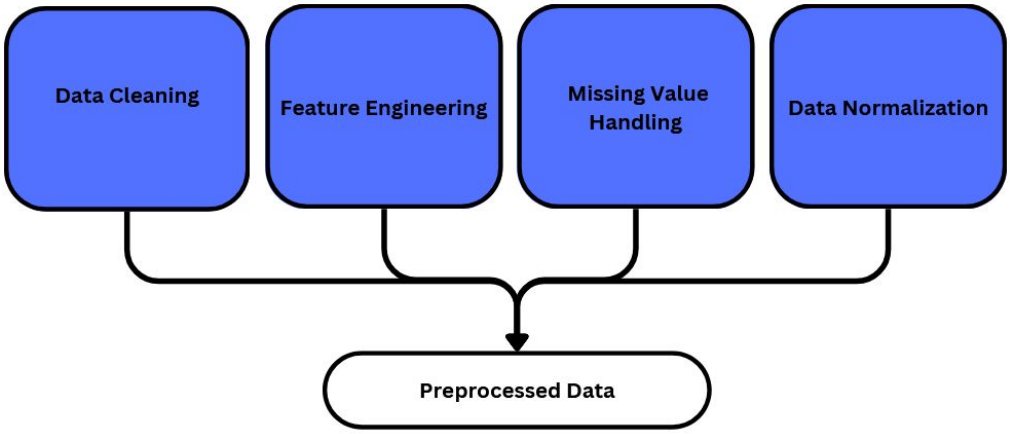
$$\text{True Score} = \text{rating} \times \left(1 - e^{-\frac{\text{rating count}}{C}}\right)$$

where:

- **rating** = the restaurant's rating (1 to 5)
- **rating count** = number of reviews
- **C** = a scaling factor (e.g., 100) that controls how quickly the count influences the score
- $e^{-\frac{\text{rating count}}{C}}$ = confidence factor that reduces uncertainty for low counts

rating	rating_count	true_score
4.1	23	1.511737053
4	1958	4
4.1	1037	4.099999996
4	31	1.84822225
3.8	65	2.764379186
4.3	26	1.743561644
4	487	3.999764478
4.2	206	4.131773039

DATA PREPROCESSING MODULE



MODEL TRAINING MODULE

LightGBM (Light Gradient Boosting Machine)

LightGBM is a fast, effective, and high-performance method that works well with structured or tabular data. It can pursue error minimization more vigorously than standard boosting techniques since it expands decision trees leaf-wise. As a result, it works best when the data includes a variety of numerical features, such in our study where the inputs are values for visibility, traffic, and population density. It was ideal for figuring out the restaurant location's success rate because of its capacity to work with large datasets, identify intricate patterns, and train quickly with little memory usage.

MODEL TRAINING MODULE

XGBoost (Extreme Gradient Boosting)

Extreme Gradient Boosting, or XGBoost, is a gradient boosting family member and a quick machine learning technique. When working with tabular data, which consists of rows and columns containing numerical or categorical attributes, XGBoost performs exceptionally well.

It is particularly adept at handling non-linear connections, handles missing values automatically, and is highly performance and speed optimized. Regularization, another aspect of XGBoost, prevents overfitting, especially when the dataset is small or features are noisy.

XGBoost is an excellent option for regression tasks because to its strength and capacity to identify minute patterns in ordered data.

MODEL TRAINING MODULE

Random Forest

It is a well-liked, all-purpose ensemble learning machine learning algorithm. To provide more reliable and accurate forecasts, it builds a large number of decision trees and aggregates their predictions. This improves generalization to fresh data and lowers the chance of single decision trees overfitting.

Random Forest can find non-linear correlations between features and performs as well on numerical and categorical data. It is also resilient to missing values and outliers, and provides feature importance scores, making it useful for understanding which inputs influence predictions most. Due to its reliability, scalability, and ease of use, Random Forest is a strong choice for both classification and regression problems, especially when interpretability and robustness are important.

MODEL TRAINING MODULE

1. Data Input: Preprocessed data from the Data Preprocessing Module is fed into the Model Training Module.

2. Data Split: The dataset is divided into training and test sets, typically in an 80/20 ratio, where the training set is used for model training, and the test set is for evaluation.

3. Model Selection:We chose LightGBM for our model because it's fast, efficient, and works exceptionally well with structured data like the kind we're using. It's great at handling complex, non-linear relationships—like how visibility or population density might affect a restaurant's success—without needing a ton of manual tweaking.

4. Hyperparameter Tuning: We fine-tuned the model's settings using Optuna. Instead of manually guessing what works, Optuna intelligently searches through different combinations to find the most effective setup for strong performance and accurate predictions.

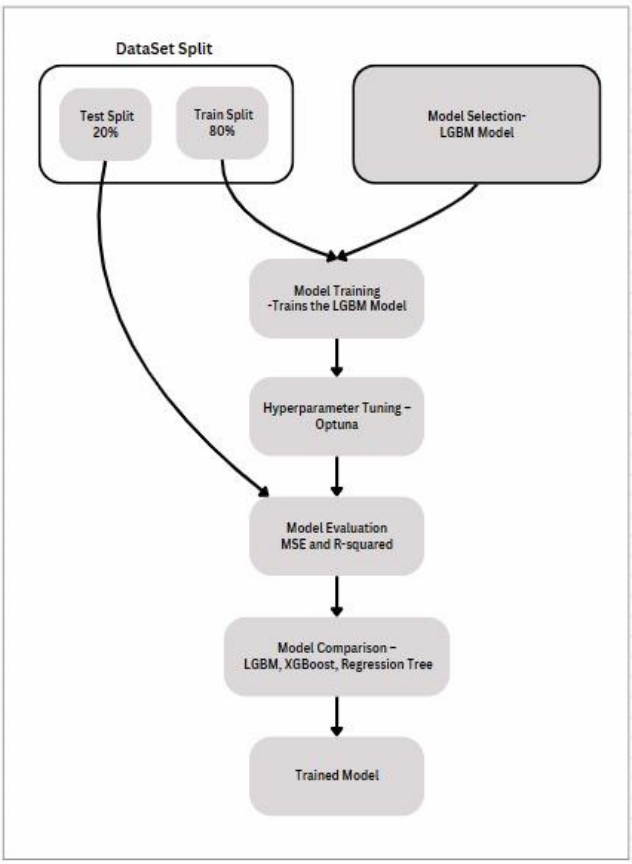
MODEL TRAINING MODULE

5.Model Training: The LightGBM model is trained on the dataset, using boosted decision trees that learn from each other to improve prediction accuracy.

6.Model Evaluation: We evaluate how well the model performs using metrics like Mean Squared Error (MSE) and R-squared on the test data.

7.Trained Model: After training and validation, the final model is saved and used by the Prediction Module to generate success scores for new inputs.

MODEL TRAINING MODULE



PREDICTION MODULE

1.Get Location Input: The user selects a location on the map.

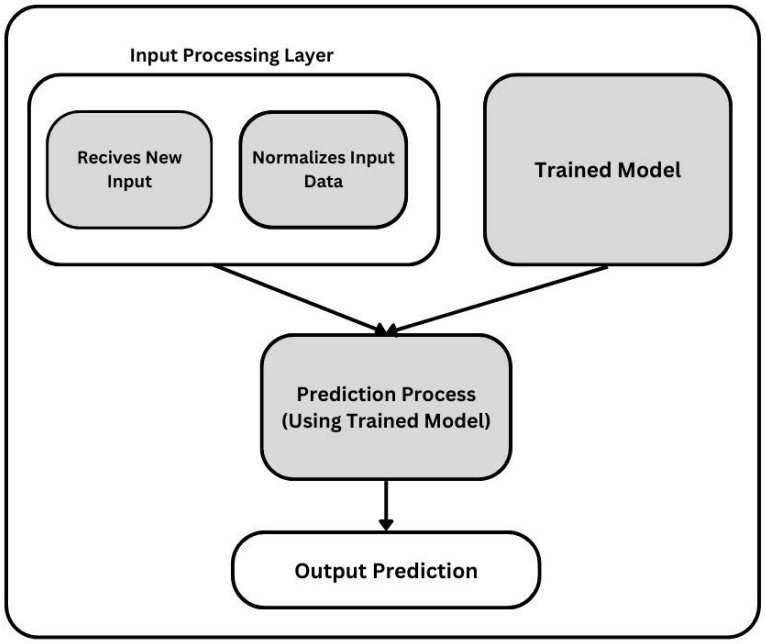
2.Extract Features: The system fetches relevant data for that location—like population density, visibility, road access, traffic, and competition.

3.Feed into Model: These features are then passed to the trained LightGBM model.

4.Generate Prediction: The model calculates a success rate for setting up a restaurant at that location.

5.Display Results: Both the predicted success rate and feature contributions are shown on the map for easy interpretation.

PREDICTION MODULE



ASSUMPTIONS

Data Quality: The input data (population density, parking availability, etc.) is accurate and collected from reliable sources, mostly reputable APIs.

User Input Accuracy: Users (restaurant owners or managers) provide accurate information regarding their restaurant's characteristics, such as cuisine type, and operating hours.

Static Conditions: The external factors influencing restaurant success (e.g., local competition, economic conditions) remain relatively stable during the prediction period.

WORK BREAKDOWN AND RESPONSIBILITIES

1. Rahul Varghese
Data Collection

2. Rebecca Liz Punnoose
Preprocessing,Feature Engineering

3. Richu Kurian, Rohan Joseph
Model Development and Tuning

REQUIREMENTS

Hardware Requirements

- CPU: Intel i5/Ryzen 5 or higher.
- RAM: 16 GB (32 GB recommended).
- Storage: 512GB SSD.
- GPU: NVIDIA GTX 1660, RTX 2060, or higher.

REQUIREMENTS

Software Requirements

Programming Language:

- Python: Main language for development and model training.

Machine Learning Libraries:

- Scikit-learn
- Xgboost
- lightGBM.

Data Handling Libraries:

- pandas: For data manipulation and preprocessing.
- numpy: For numerical computations and handling arrays.

REQUIREMENTS

Mapping & API Integration:

- requests: For making API requests to Google Maps (Geocoding, Places, Distance Matrix APIs).
- Google Maps API: For embedding maps and user interaction on the frontend.
- Google Static Maps API: Display a static map on your website.
- Google Geocoding API: Convert coordinates into addresses and addresses into coordinates
- Google Maps API Key: Required for accessing Google Maps services.

REQUIREMENTS

Web Development:

- Flask: For building the backend of the web application.
- REACT: For frontend development and map integration.

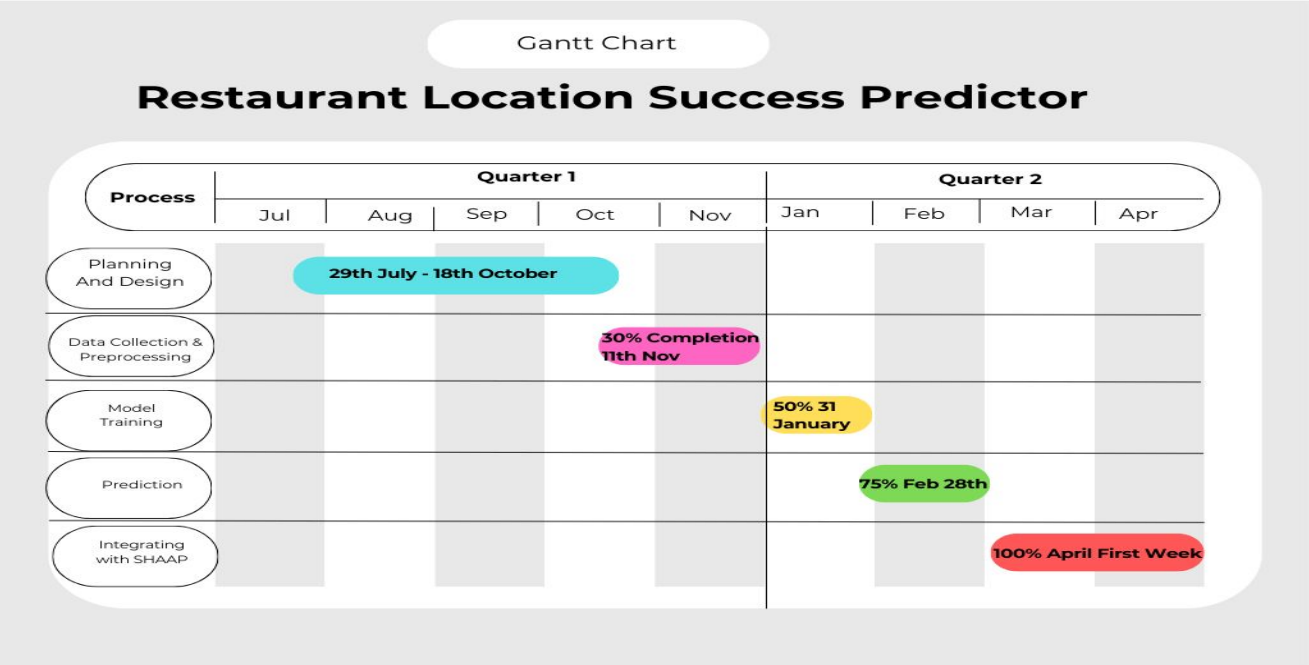
Development Tools:

- Jupyter Notebook: For experimenting with models and data processing interactively.
- VS Code : For structured code development and debugging.

Version Control:

- Git: For version control and collaboration.
- GitHub: For code storage and sharing.

Gantt Chart



RISKS & CHALLENGES

1. Data from external APIs like Google Maps may be outdated or incomplete, affecting model accuracy.
2. Google Maps APIs have request limits, leading to potential cost or scaling issues.
3. The model may overfit if the dataset lacks variation, reducing real-world performance.
4. Combining data sources and normalizing them (e.g., 1-5 scale) is complex and needs careful feature engineering.

RESULTS

The results highlight a clear improvement in model performance with data augmentation.

LightGBM with augmentation performed the best overall, showing the lowest error values and the highest R-squared score, indicating strong predictive accuracy and better generalization. XGBoost with augmentation was close behind, also showing solid performance.

Without augmentation, both models had noticeably higher errors and lower R-squared values, suggesting weaker predictive reliability. Random Forest, even with augmentation, struggled to match the accuracy of the boosted models, indicating it's less suited for this task.

Overall, LightGBM with augmentation stands out as the most reliable model for predicting location success.

RESULTS

MODEL COMPARISONS

XGBOOST

```
Optimized model and scaler saved.
Optimized Mean Squared Error: 1.984845144418669
Optimized Root Mean Squared Error: 1.4088453230992637
Optimized R-squared Score: 0.08329821935238524
```

LIGHTGBM

```
Optimized LGBM model and scaler saved.
Optimized Mean Squared Error: 2.0012992721116314
Optimized Root Mean Squared Error: 1.4146728498531493
Optimized R-squared Score: 0.07569887176723156
```


RESULTS

MODEL COMPARISONS

LIGHTGBM WITH AUGMENTATION

```
Optimized LGBM model and scaler saved.
Optimized Mean Squared Error: 1.3539209924503302
Optimized Root Mean Squared Error: 1.1635811069497175
Optimized R-squared Score: 0.3826561742342949
```

XGBOOST WITH AUGMENTATION

```
Optimized model and scaler saved.
Optimized Mean Squared Error: 1.3651221858185985
Optimized Root Mean Squared Error: 1.16838443408777
Optimized R-squared Score: 0.37754879529145613
```

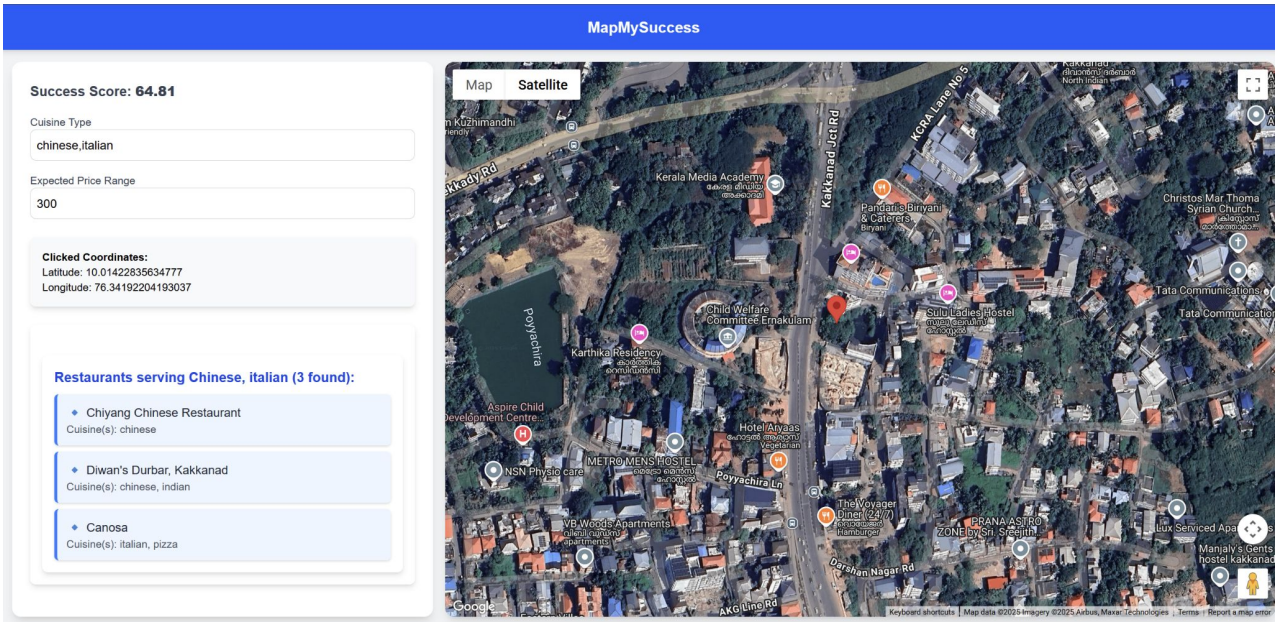
RESULTS

MODEL COMPARISONS

RANDOM FOREST WITH AUGMENTATION

```
Optimized RF model and scaler saved.
Optimized Mean Squared Error: 1.9684610008679293
Optimized Root Mean Squared Error: 1.4030185319046677
Optimized R-squared Score: 0.09086524472440638
```


RESULTS



RESULTS

The model did a great job of identifying general patterns and producing reliable predictions of success. The results demonstrate that it can provide useful guidance to aid in decision-making, even while it never aims to forecast every possible event with laser-like accuracy. The system can be a useful tool in helping choose possible sites, and the forecasts are rather accurate. The accuracy and dependability of the model can be further improved with more thorough data and further modification.

CONCLUSION

In conclusion, this research offers a data-driven solution for restaurant performance prediction in a particular location using machine learning techniques. The algorithm generates reliable success scores that can help restaurants make the right choices based on the identification of key factors like population density, price level, visibility, competition proximity, etc. This technology has the potential to revolutionize restaurant location selection, which would eventually improve business outcomes.

FUTURE SCOPE

More information, such as market trends and social media sentiment, can improve forecasts and make them more accurate and pertinent.

Providing clear and understandable explanations for each forecast might help businesspeople feel more at ease with the results.

For someone starting a new restaurant branch, having this webpage easily accessible with maps and general recommendations would be quite helpful.

REFERENCES

1. Deep Multi-Task Learning with Relational Attention - Jiejie Zhao, Bowen Du, Leilei Sun, Weifeng Lv, Yanchi Liu, Hui Xiong

2. The Explanation Game: Explaining Machine Learning Models using Shapley values. - Luke Merrick, Ankur Taly

3. Prediction of Employee Turn Over Using Random Forest Classifier with Intensive Optimized Pca Algorithm. - Alaeldeen Bader Wild Ali

4. A machine learning, bias-free approach for predicting business success using Crunchbase data. - Kamil Żbikowski, Piotr Antosiuk

THANK YOU

Appendix B: Vision, Mission, Programme Outcomes and Course Outcomes

Vision, Mission, Programme Outcomes and Course Outcomes

Institute Vision

To evolve into a premier technological institution, moulding eminent professionals with creative minds, innovative ideas and sound practical skill, and to shape a future where technology works for the enrichment of mankind.

Institute Mission

To impart state-of-the-art knowledge to individuals in various technological disciplines and to inculcate in them a high degree of social consciousness and human values, thereby enabling them to face the challenges of life with courage and conviction.

Department Vision

To become a centre of excellence in Computer Science and Engineering, moulding professionals catering to the research and professional needs of national and international organizations.

Department Mission

To inspire and nurture students, with up-to-date knowledge in Computer Science and Engineering, ethics, team spirit, leadership abilities, innovation and creativity to come out with solutions meeting societal needs.

Programme Outcomes (PO)

Engineering Graduates will be able to:

1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and Team work: Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.

10. Communication: Communicate effectively with the engineering community and with society at large. Be able to comprehend and write effective reports documentation. Make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

Programme Specific Outcomes (PSO)

A graduate of the Computer Science and Engineering Program will demonstrate:

PSO1: Computer Science Specific Skills
The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas by understanding the core principles and concepts of computer science and thereby engage in national grand challenges.

PSO2: Programming and Software Development Skills
The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry.

PSO3: Professional Skills
The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur.

Course Outcomes (CO)

Course Outcome 1: Model and solve real world problems by applying knowledge across domains (Cognitive knowledge level: Apply).

Course Outcome 2: Develop products, processes or technologies for sustainable and socially relevant applications (Cognitive knowledge level: Apply).

Course Outcome 3: Function effectively as an individual and as a leader in diverse teams and to comprehend and execute designated tasks (Cognitive knowledge level: Apply).

Course Outcome 4: Plan and execute tasks utilizing available resources within timelines, following ethical and professional norms (Cognitive knowledge level: Apply).

Course Outcome 5: Identify technology/research gaps and propose innovative/creative solutions (Cognitive knowledge level: Analyze).

Course Outcome 6: Organize and communicate technical and scientific findings effectively in written and oral forms (Cognitive knowledge level: Apply).

Appendix C: CO-PO-PSO Mapping

COURSE OUTCOMES:

After completion of the course, the student will be able to:

SL.NO	DESCRIPTION	Bloom's Taxonomy Level
CO1	Model and solve real-world problems by applying knowledge across domains (Cognitive knowledge level:Apply).	Level3: Apply
CO2	Develop products, processes, or technologies for sustainable and socially relevant applications. (Cognitive knowledge level:Apply).	Level 3: Apply
CO3	Function effectively as an individual and as a leader in diverse teams and comprehend and execute designated tasks. (Cognitive knowledge level:Apply).	Level 3: Apply
CO4	Plan and execute tasks utilizing available resources within timelines, following ethical and professional norms (Cognitive knowledge level:Apply).	Level 3: Apply
CO5	Identify technology/research gaps and propose innovative/creative solutions (Cognitive knowledge level:Analyze).	Level 4: Analyze
CO6	Organize and communicate technical and scientific findings effectively in written and oral forms (Cognitive knowledge level:Apply).	Level 3: Apply

CO-PO AND CO-PSO MAPPING

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	2	2	1	2	2	2	1	1	1	1	2	3		
CO2	2	2	2		1	3	3	1	1		1	1		2	
CO3									3	2	2	1			3
CO4					2			3	2	2	3	2			3
CO5	2	3	3	1	2							1	3		
CO6					2			2	2	3	1	1			3

3/2/1: high/medium/low

JUSTIFICATIONS FOR CO-PO MAPPING

Mapping	Level	Justification
101003/CS822U.1-PO1	M	Knowledge in the area of technology for project development using various tools results in better modeling.
101003/CS822U.1-PO2	M	Knowledge acquired in the selected area of project development can be used to identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions.
101003/CS822U.1-PO3	M	Can use the acquired knowledge in designing solutions to complex problems.
101003/CS822U.1-PO4	M	Can use the acquired knowledge in designing solutions to complex problems.
101003/CS822U.1-PO5	H	Students are able to interpret, improve, and redefine technical aspects for design of experiments, analysis, and interpretation of data, and synthesis of the information to provide valid conclusions.
101003/CS822U.1-PO6	M	Students are able to interpret, improve, and redefine technical aspects by applying contextual knowledge to assess societal, health, and consequential responsibilities relevant to professional engineering practices.
101003/CS822U.1-PO7	M	Project development based on societal and environmental context solution identification is the need for sustainable development.
101003/CS822U.1-PO8	L	Project development should be based on professional ethics and responsibilities.
101003/CS822U.1-PO9	L	Project development using a systematic approach based on well-defined principles will result in teamwork.
101003/CS822U.1-PO10	M	Project brings technological changes in society.
101003/CS822U.1-PO11	H	Acquiring knowledge for project development gathers skills in design, analysis, development, and implementation of algorithms.

101003/CS822U.1-PO12	H	Knowledge for project development contributes engineering skills in computing and information gatherings.
101003/CS822U.2-PO1	H	Knowledge acquired for project development will also include systematic planning, developing, testing, and implementation in computer science solutions in various domains.
101003/CS822U.2-PO2	H	Project design and development using a systematic approach brings knowledge in mathematics and engineering fundamentals.
101003/CS822U.2-PO3	H	Identifying, formulating, and analyzing the project results in a systematic approach.
101003/CS822U.2-PO5	H	Systematic approach is the tip for solving complex problems in various domains.
101003/CS822U.2-PO6	H	Systematic approach in the technical and design aspects provides valid conclusions.
101003/CS822U.2-PO7	H	Systematic approach in the technical and design aspects demonstrates the knowledge of sustainable development.
101003/CS822U.2-PO8	M	Identification and justification of technical aspects of project development demonstrates the need for sustainable development.
101003/CS822U.2-PO9	H	Apply professional ethics and responsibilities in engineering practice of development.
101003/CS822U.2-PO11	H	Systematic approach also includes effective reporting and documentation, which gives clear instructions.
101003/CS822U.2-PO12	M	Project development using a systematic approach based on well-defined principles will result in better teamwork.
101003/CS822U.3-PO9	H	Project development as a team brings the ability to engage in independent and lifelong learning.

101003/CS822U.3-PO10	H	Identification, formulation, and justification in technical aspects will be based on acquiring skills in design and development of algorithms.
101003/CS822U.3-PO11	H	Identification, formulation, and justification in technical aspects provides the betterment of life in various domains.
101003/CS822U.3-PO12	H	Students are able to interpret, improve, and redefine technical aspects with mathematics, science, and engineering fundamentals for the solutions of complex problems.
101003/CS822U.4-PO5	H	Students are able to interpret, improve, and redefine technical aspects with identification, formulation, and analysis of complex problems.
101003/CS822U.4-PO8	H	Students are able to interpret, improve, and redefine technical aspects to meet the specified needs with appropriate consideration for public health and safety, and the cultural, societal, and environmental considerations.
101003/CS822U.4-PO9	H	Students are able to interpret, improve, and redefine technical aspects for design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
101003/CS822U.4-PO10	H	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools for better products.
101003/CS822U.4-PO11	M	Students are able to interpret, improve, and redefine technical aspects by applying contextual knowledge to assess societal, health, and consequential responsibilities relevant to professional engineering practices.
101003/CS822U.4-PO12	H	Students are able to interpret, improve, and redefine technical aspects for demonstrating the knowledge of, and need for sustainable development.

101003/CS822U.5-PO1	H	Students are able to interpret, improve, and redefine technical aspects, apply ethical principles, and commit to professional ethics and responsibilities and norms of the engineering practice.
101003/CS822U.5-PO2	M	Students are able to interpret, improve, and redefine technical aspects, communicate effectively on complex engineering activities with the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
101003/CS822U.5-PO3	H	Students are able to interpret, improve, and redefine technical aspects to demonstrate knowledge and understanding of the engineering and management principle in multidisciplinary environments.
101003/CS822U.5-PO4	H	Students are able to interpret, improve, and redefine technical aspects, recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.
101003/CS822U.5-PO5	M	Students are able to interpret, improve, and redefine technical aspects in acquiring skills to design, analyze, and develop algorithms and implement those using high-level programming languages.
101003/CS822U.5-PO12	M	Students are able to interpret, improve, and redefine technical aspects and contribute their engineering skills in computing and information engineering domains like network design and administration, database design, and knowledge engineering.
101003/CS822U.6-PO5	M	Students are able to interpret, improve, and redefine technical aspects and develop strong skills in systematic planning, developing, testing, implementing, and providing IT solutions for different domains, which helps in the betterment of life.

101003/CS822U.6-PO8	H	Students will be able to associate with a team as an effective team player for the development of technical projects by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
101003/CS822U.6-PO9	H	Students will be able to associate with a team as an effective team player to identify, formulate, review research literature, and analyze complex engineering problems.
101003/CS822U.6-PO10	M	Students will be able to associate with a team as an effective team player for designing solutions to complex engineering problems and design system components.
101003/CS822U.6-PO11	M	Students will be able to associate with a team as an effective team player, use research-based knowledge and research methods including design of experiments, analysis, and interpretation of data.
101003/CS822U.6-PO12	H	Students will be able to associate with a team as an effective team player, applying ethical principles and committing to professional ethics and responsibilities and norms of the engineering practice.
101003/CS822U.1-PSO1	H	Students are able to develop Computer Science Specific Skills by modeling and solving problems.
101003/CS822U.2-PSO2	M	Developing products, processes or technologies for sustainable and socially relevant applications can promote Programming and Software Development Skills.
101003/CS822U.3-PSO3	H	Working in a team can result in the effective development of Professional Skills.
101003/CS822U.4-PSO3	H	Planning and scheduling can result in the effective development of Professional Skills.
101003/CS822U.5-PSO1	H	Students are able to develop Computer Science Specific Skills by creating innovative solutions to problems.

101003/CS822U.6-PSO3	H	Organizing and communicating technical and scientific findings can help in the effective development of Professional Skills..