Forecasting Feasts: A Culinary Journey into Restaurant Revenue Prediction

AN INDUSTRY ORIENTED MINI REPORT

Submitted to

JAWAHARLAL NEHRU TECNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

Submitted By

SRUJAN PALAKURTHI	**(21UK1A05B7)**
SHRUTHI GUNDA	**(22UK5A0509)**
HARSHITHA PULICHERI	**(21UK1A05C4)**
BHARGAVA KOUSHIK PINDI	**(21UK1A05B9)**

Under the guidance of

Mrs. S. Anoosha

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

Affiliated to JNTUH, HYDERABAD

BOLLIKUNTA, WARANGAL (T.S) –

506005

DEPARTMENT OF

COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE(WARANGAL)

CERTIFICATE OF COMPLETION

INDUSTRY ORIENTED MINI PROJECT

This is to certify that the UG Project Phase-1 entitled "FORECASTING FEAST:A CULINARY JOURNEY INTO RESTAURANT REVENUE" is being submitted SRUJAN PALAKURTHI(21UK1A05B7), SHRUTHI GUNDA (22UK5A0509), HARSHITHA PULICHERI(21UK1A05C4), BHARAGAVA KOUSHIK PINDI (21UK1A05B9) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering to Jawaharlal Nehru Technological University Hyderabad during the academic year 2024-2025.

Project Guide	**HOD**
Mrs. S. Anoosha	**Dr. Naveenkumar**
(Assistant Professor)	(Professor)

External

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr .SYEDMUSTHAK AHAMED** , Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this UG Project Phase-1 in the institute.

We extend our heartfelt thanks to **Dr. NAVEEN KUMAR,** Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Project Phase-1.

We express heartfelt thanks to Smart Bridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the UG Project Phase-1 and for their support in completing the UG Project Phase-1.

We express heartfelt thanks to the guide, **S. ANOOSHA** , Assistant professor, Department of CSE for his constant support and giving necessary guidance for completion of
this UG Project Phase-1.

Finally, we express our sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughout the thesis.

SRUJAN PALAKURTHI	**(21UK1A05B7)**
SHRUTHI GUNDA	**(22UK5A0509)**
HARSHITHA PULICHERI	**(21UK1A05C4)**
BHARGAVA KOUSHIK PINDI	**(21UK1A05B9)**

ABSTRACT

"Forecasting Feasts: A Culinary Journey into Restaurant Revenue Prediction" is a data-driven project designed to assist restaurant owners and managers in predicting future revenue trends based on historical data and various factors. By leveraging machine learning and data analysis techniques, the system aims to provide actionable insights for optimizing business strategies and enhancing profitability in the culinary industry.

Scenario1: Demand Forecasting

One scenario involves using the system to forecast customer demand for specific dishes or menu items. By analyzing past sales data, customer preferences, seasonal trends, and external factors like weather or events, restaurant owners can make informed decisions regarding menu planning, pricing strategies, and inventory management.

Scenario 2: Resource Allocation

Another scenario focuses on optimizing resource allocation within a restaurant. By predicting revenue trends, the system can help managers allocate staff, ingredients, and operational resources more effectively, ensuring smooth operations and minimizing waste.

Scenario 3: Marketing Campaign Effectiveness

The system can also be used to evaluate the effectiveness of marketing campaigns. By correlating revenue data with marketing efforts such as promotions, advertisements, or loyalty programs, restaurant owners can assess which strategies yield the highest returns and refine their marketing strategies accordingly.

TABLE OF CONTENTS:-

1.INTRODUCTION

1.1OVERVIEW

Predicting restaurant revenue involves leveraging data analytics and statistical modeling techniques to forecast future sales or revenue for a restaurant. Here's an overview of how this process can be approached:

Data Collection: Gather relevant historical data on restaurant sales typically including daily, weekly, or monthly revenue figures. Addition a data such as customer counts, menu items sold, pricing, promotional weather conditions, and economic indicators can also be valuable.

Data Preprocessing: Clean and preprocess the data to handle missing values, outliers, and inconsistencies. Transform the data into a suitable format for analysis, such as aggregating sales data by time periods (e.g., daily totals).

Exploratory Data Analysis (EDA): Explore the data to uncover patterns, trends, and relationships. EDA helps in understanding factors influencing revenue, such as seasonal variations (e.g., holidays, weekends), correlations between sales and external factors (like weather), and customer behaviour.

Feature Engineering: Create new features from existing data that can enhance predictive accuracy. Examples include average revenue per customer, day of the week effects, special events or holidays, and marketing campaign impacts.

Model Selection: Choose appropriate predictive models based on the nature of the data and business requirements. Commonly used models for restaurant revenue prediction include:

Time Series Models: Such as ARIMA (Auto Regressive Integrated Moving Average) or SARIMA (Seasonal ARIMA) for capturing seasonal patterns and trends over time.

Machine Learning Models: Regression techniques like linear regression, decision trees, random forests, or gradient boosting models can incorporate both time-based features and additional predictors.

By following these steps, restaurant operators can effectively utilize predictive analytics to forecast revenue, optimize operations, and enhance decision-making processes in the competitive restaurant industry.

1.2 PURPOSE

effective business management and decision-making in the food service industry. Here The purpose of restaurant revenue prediction is multifaceted and crucial for are some key purposes:

1. **Financial Planning and Budgeting**: Predicting restaurant revenue helps in creating accurate financial forecasts and budgets. It allows restaurant owners and managers to plan expenditures, manage cash flow, and allocate resources effectively.
2. **Optimizing Inventory and Supply Chain Management**: Accurate revenue predictions enable better inventory management by anticipating demand for ingredients and supplies. This helps in minimizing food wastage, ensuring stock availability, and optimizing purchasing decisions.
3. **Staffing and Labor Optimization**: Forecasting revenue helps in scheduling staff according to anticipated customer traffic. It ensures optimal staffing levels to provide quality service during peak periods while avoiding overstaffing during slower times, thereby controlling labor costs.
4. **Menu Planning and Pricing Strategy**: Revenue predictions provide insights into popular menu items and their profitability. This data informs menu planning decisions, such as adjusting offerings to meet customer preferences and optimizing pricing strategies to maximize revenue.
5. **Marketing and Promotions**: Understanding revenue trends helps in strategizing marketing campaigns and promotions effectively. Restaurants can plan targeted promotions during slower periods or leverage high-demand periods to attract more customers.
6. **Operational Efficiency**: By accurately predicting revenue, restaurants can operate more efficiently. This includes optimizing seating arrangements, managing reservations, and streamlining operational processes to enhance overall customer experience and satisfaction.
7. **Risk Management**: Revenue predictions assist in identifying potential risks and challenges. For instance, forecasting can help restaurants prepare for seasonal fluctuations, economic downturns, or unexpected events that may impact revenue.
8. **Strategic Decision Making**: Revenue predictions serve as a valuable tool for strategic decision-making. Whether expanding operations, opening new locations, or investing in renovations, accurate revenue forecasts provide essential data to support these decisions.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

Predicting restaurant revenue accurately can be challenging due to several factors:

- **Seasonality and Trends**: Restaurants often experience fluctuating demand based on seasons, holidays, and local events. Predicting these patterns accurately requires historical data and understanding local factors.
- **Menu Changes and Specials**: Introducing new menu items or running promotions can significantly impact revenue. Predictive models need to account for these changes and their effects on customer behavior.
- **External Factors**: Economic conditions, competition, and even weather can influence restaurant traffic and revenue. These external variables are crucial to consider in predictive models.
- **Customer Preferences**: Understanding customer preferences, demographics, and behavior patterns can enhance revenue predictions. This involves analyzing customer data and feedback effectively.
- **Operational Challenges**: Efficiency in operations, staffing levels, and supply chain management can affect revenue directly. Predictive models should integrate operational data to improve accuracy.
- **Data Quality and Integration**: Integrating data from various sources (POS systems, reservations, social media, etc.) while ensuring data quality is crucial for reliable predictions.

2.2 PROPOSED SOLUTION

To address the challenges in predicting restaurant revenue, here’s a proposed solution framework:

1. Data Collection and Integration

- **Data Sources**: Gather data from various sources such as POS systems, reservation platforms, weather APIs, local event calendars, and social media.
- **Data Quality**: Ensure data accuracy, completeness, and consistency through data cleaning and validation processes.

2. Feature Engineering

- **Time-Series Features**: Extract and engineer features like day of the week, month, seasonality, holidays, and special events.
- **Menu and Promotion Impact**: Incorporate features related to menu changes, promotions, and discounts to understand their impact on revenue.

3. Predictive Modeling Techniques

- **Time-Series Forecasting**: Utilize techniques such as ARIMA (Auto Regressive Integrated Moving Average) or SARIMA (Seasonal ARIMA) models to forecast revenue based on historical data.
- **Machine Learning Models**: Implement regression models (e.g., linear regression, random forest) to predict revenue considering both time-series features and external factors.

4. Customer and Operational Insights

- **Customer Segmentation**: Analyze customer data to segment based on demographics, preferences, and spending patterns.
- **Feedback Analysis**: Use sentiment analysis on customer reviews and feedback to understand customer satisfaction and its impact on revenue.

5. Model Evaluation and Refinement

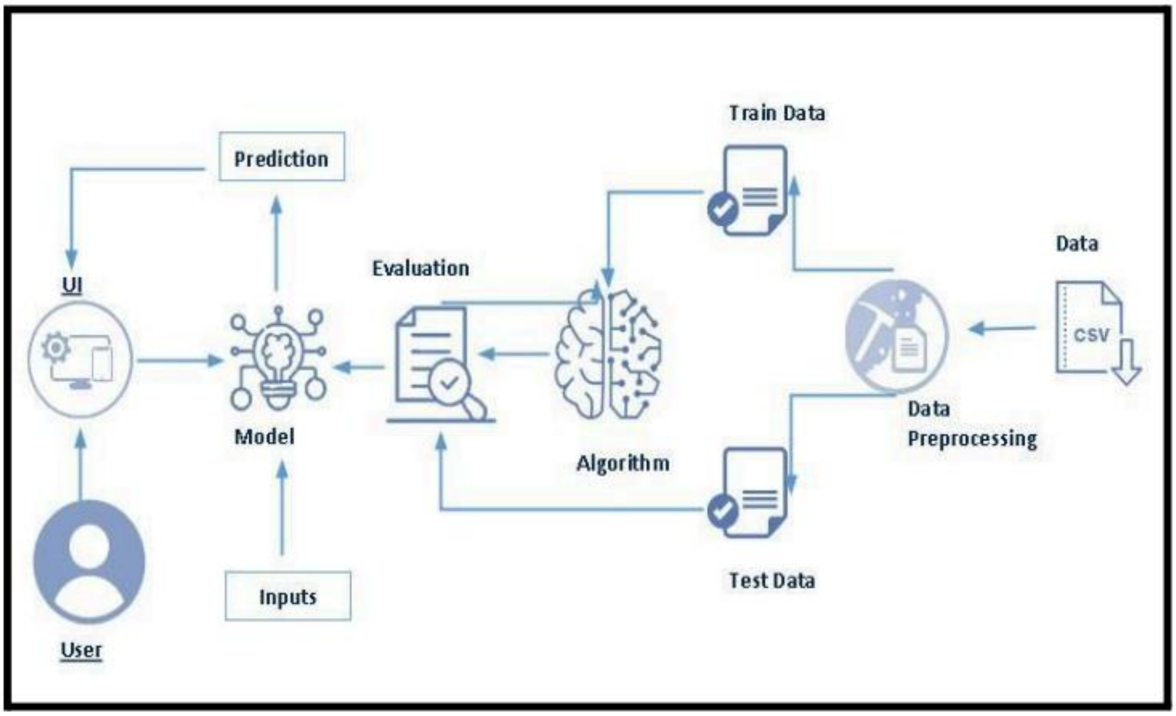
- **Cross-Validation**: Validate models using techniques like k-fold cross-validation to ensure robustness and generalizability.
- **Continuous Learning**: Implement feedback loops to update models with new data and refine predictions over time.

6. Visualization and Reporting

- **Dashboard Development**: Create interactive dashboards to visualize predicted vs. actual revenue, key performance indicators (KPIs), and trends over time.

3.THEORITICAL ANALYSIS

3.1 BLOCK DIAGRAM



3.2. SOFTWARE DESIGNING

The following is the Software required to complete this project:

➢ **Google Colab:** Google Colab will serve as the development and execution environment for your predictive modelling data preprocessing, and model training tasks. It provides a cloud-based Jupyter Notebook environment with access to Python libraries and hardware acceleration.

➢ **Dataset (CSV File):** The dataset in CSV format is essential for training and testing your predictive model. It should include historical air quality data, weather information, pollutant levels, and other relevant features.

➢ **Data Preprocessing Tools**: Python libraries like NumPy, Pandas, and Scikit-learn will be used to preprocess the dataset. This includes handling missing data, feature scaling, and data cleaning.

➢ **Feature Selection/Drop:** Feature selection or dropping unnecessary features from the dataset can be done using Scikit-learn or custom Python code to enhance the model's efficiency.

4. Model Selection:

- **Choose Suitable Models:** Select predictive models based on the nature of the problem (regression for continuous prediction of revenue).
- **Consideration of Algorithms:** Evaluate algorithms like linear regression, decision trees, random forests, or more sophisticated methods like neural networks depending on the complexity and size of data.

5. Model Training and Evaluation:

- **Training:** Train the selected models using historical data, typically splitting data into training and validation sets.
- **Evaluation:** Evaluate models using metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), or Mean Absolute Percentage Error (MAPE).

6. Deployment:

- **Implementation:** Implement the predictive model into a software application.
- **User Interface:** Develop a user-friendly interface for stakeholders to input parameters (e.g., date range, special events) and view predictions.
- **Integration:** Integrate with existing restaurant management systems if applicable.

7. Monitoring and Maintenance:

- **Monitoring:** Regularly monitor model performance and recalibrate if necessary (e.g., retraining with new data, updating feature engineering).
- **Feedback Loop:** Incorporate feedback from users to improve the accuracy and usability of the prediction software.

Additional Considerations:

- **Scalability:** Ensure the software can handle increasing amounts of data as the restaurant grows.
- **Interpretability:** Consider the interpretability of the model outputs to explain predictions to stakeholders.
- **Compliance:** Ensure compliance with relevant data privacy regulations (e.g., GDPR, CCPA).

4.EXPERIMENTAL INVESTIGATION

Conducting an experiment for investigating restaurant revenue prediction involves systematically designing and executing a study to evaluate the effectiveness of different prediction models or strategies. Here's a structured approach you can follow:

1. Define Objectives:

- **Specific Goals:** Clearly define what you aim to achieve with the experiment (e.g., comparing the accuracy of different prediction models, evaluating the impact of specific features).

2. Hypothesis Formulation:

- **Formulate Hypotheses:** Based on your objectives, formulate hypotheses that can be tested during the experiment (e.g., "Model A will provide more accurate revenue predictions compared to Model B").

3. Experimental Design:

- **Selection of Variables:** Identify the independent variables (e.g., different prediction models, feature sets) and dependent variables (e.g., prediction accuracy metrics such as RMSE, MAE).

4. Data Collection:

- **Data Sources:** Gather historical data on restaurant sales, menu items, weather conditions, special events, and any other relevant factors.

5. Model Selection and Preparation:

- **Model Selection:** Choose the prediction models to be evaluated based on the experiment's objectives (e.g., linear regression, decision trees, neural networks).

6. Experiment Execution:

- **Implementation:** Implement each prediction model according to the experimental design.
- **Comparison:** Compare the performance of different models based on the evaluation metrics.

7. Analysis and Interpretation:

- **Comparison:** Compare the performance of different models based on the evaluation metrics.
- **Statistical Analysis:** Conduct statistical tests (if applicable) to determine if observed differences in performance are statistically significant.

8. Documentation and Reporting:

- **Documentation:** Document the experiment design, methodology, data sources, models used, and results thoroughly.

9. Iterative Improvement:

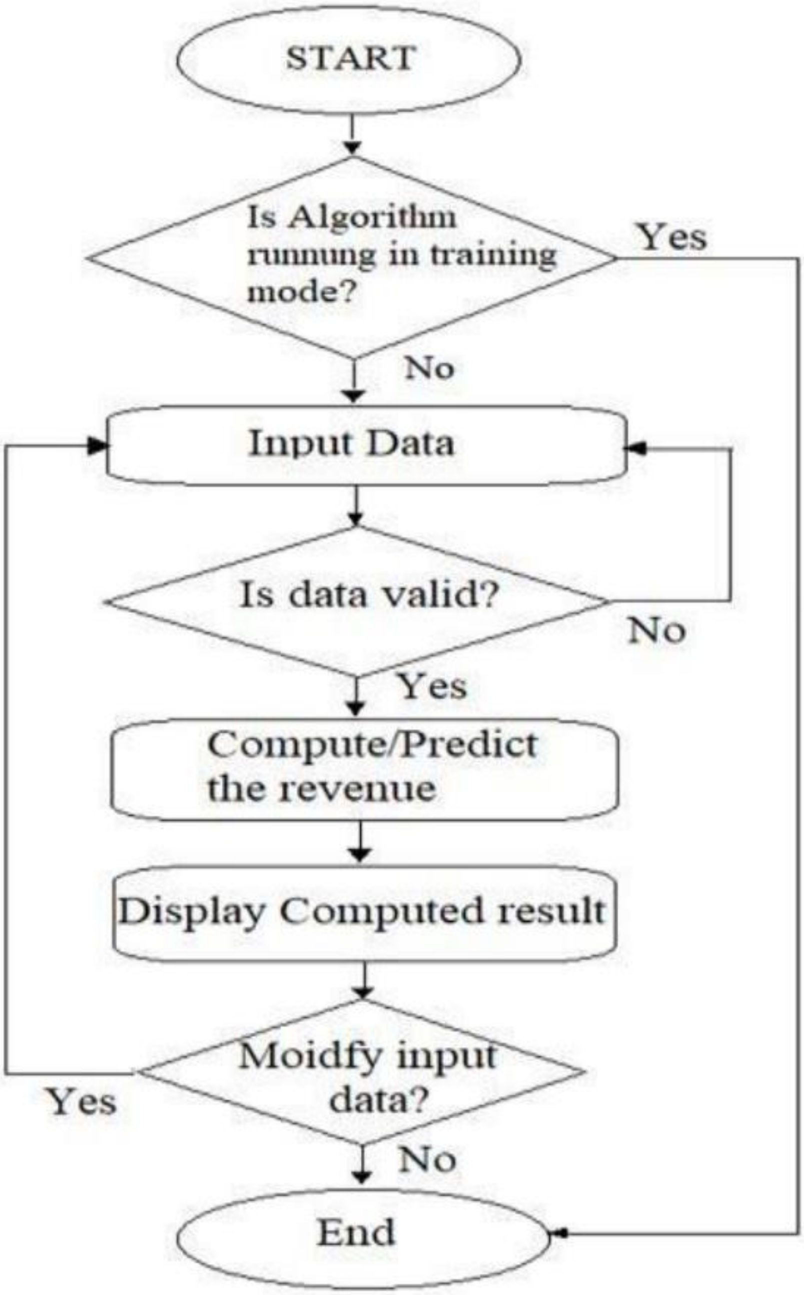
- **Feedback Loop:** Incorporate feedback from stakeholders and potentially refine the experiment design or models based on insights gained.

Additional Considerations:

- **Sample Size:** Ensure the dataset used is sufficiently large to provide reliable results and avoid biases.
- **Ethical Considerations:** Respect data privacy and confidentiality throughout the experiment.
- **Practical Application:** Consider how the findings from the experiment can be applied practically in real-world scenarios.

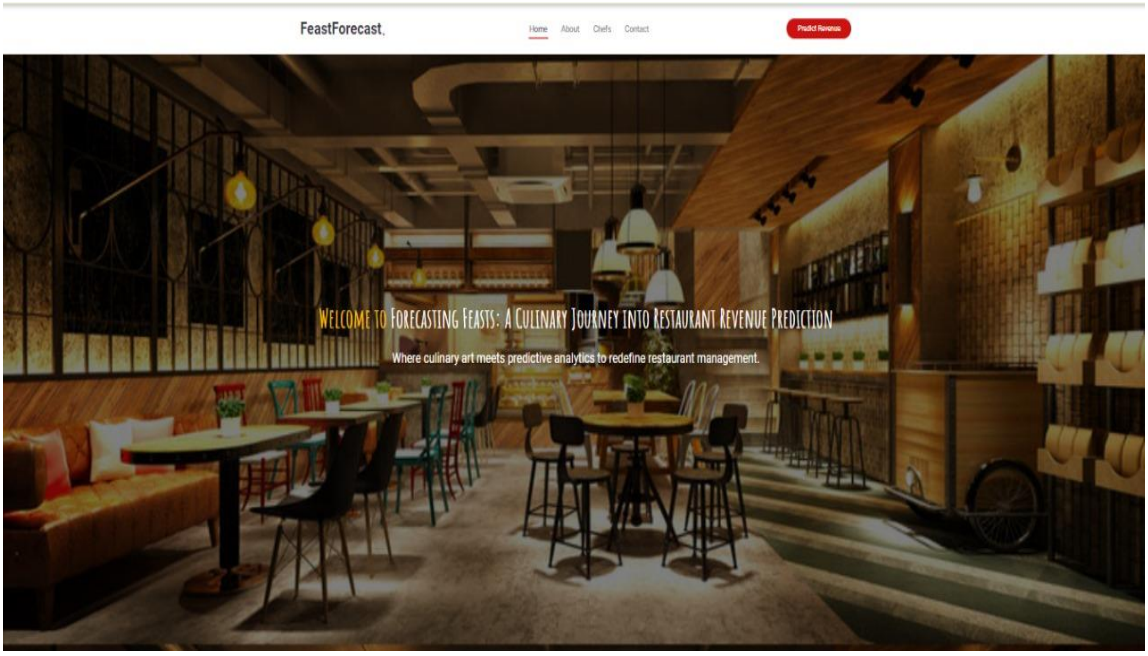
By following this structured approach, you can effectively conduct an experiment to investigate restaurant revenue prediction, leading to insights that can inform decision-making and potentially improve prediction accuracy in practical applications.

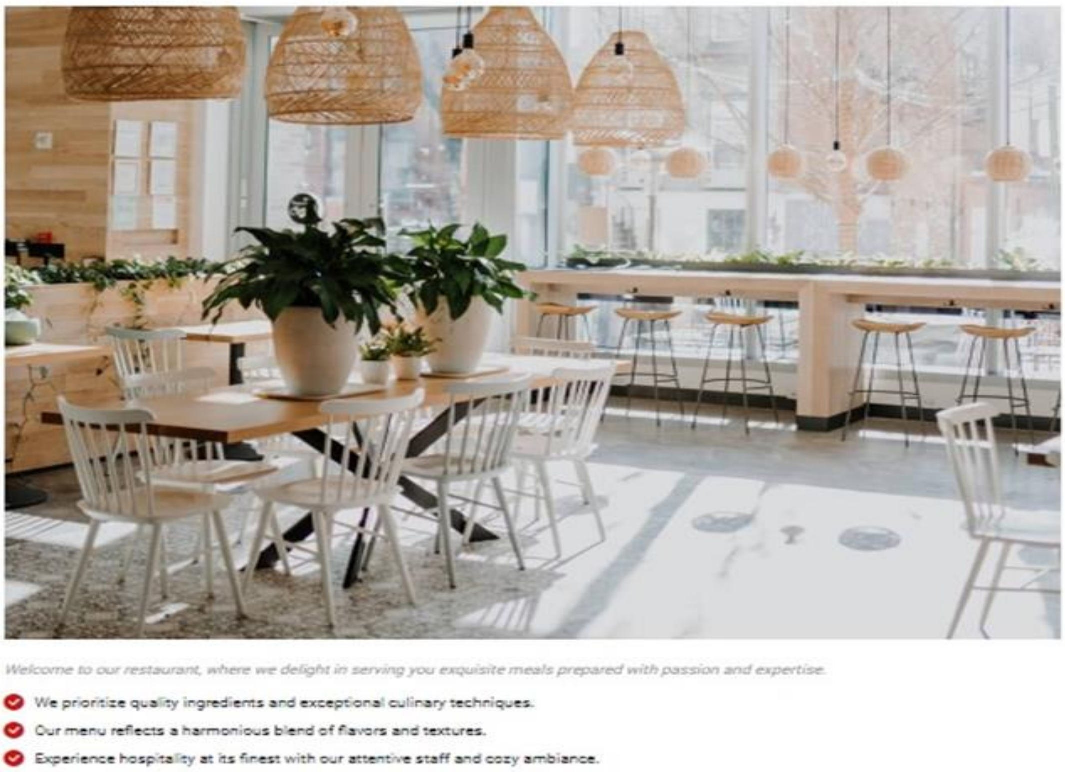
5.FLOWCHART


START
Is Algorithm runnung in training mode?
Yes
No
Input Data
Is data valid?
No
Yes
Compute/Predict the revenue
Display Computed result
Moidfy input data?
Yes
No
End

6.RESULT

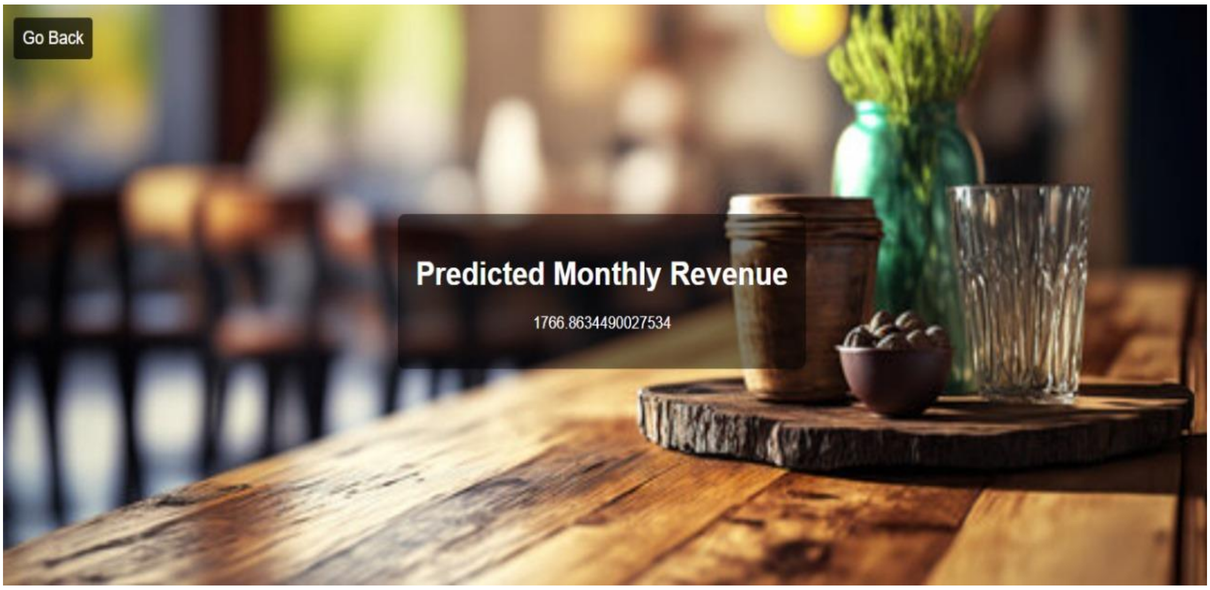
HOME PAGE






MEET OUR PROFESSIONAL CHEFS
Walter White
Master Chef
Known for his innovative approach, Chef Walter White blends traditional flavors with modern techniques to create memorable dining experiences.
Sarah Johnson
Patissier
Pastry aficionado Sarah Johnson brings sweetness to our dishes with her mastery of desserts, ensuring every bite is a delight.


Revenue Prediction
Number of Customers:
Menu Price:
Marketing Spend:
Cuisine Type:
Average Customer Spending:
Promotions:
Reviews:
Predict Revenue


Go Back
Predicted Monthly Revenue
1766.8634490027534

7.ADVANTAGES AND DISADVANTAGES

Advantages:

Better Resource Planning:

- Predicting revenue helps in planning resources such as inventory, staffing levels, and kitchen preparation.

Optimized Menu and Pricing Strategies:

- Insight into revenue trends allows restaurants to adjust menu offerings and pricing strategies to maximize profitability.

Improved Financial Management:

- Accurate revenue prediction facilitates better financial management by forecasting cash flow and revenue streams.

Enhanced Customer Experience:

- Predicting demand enables restaurants to provide better customer service by ensuring timely service and availability of popular menu items.

Disadvantages:

Uncertainty and Variability:

Revenue prediction in the restaurant industry can be challenging due to fluctuating factors such as seasonality, weather conditions, and economic shifts.

Data Dependency and Quality:

Predictive models rely heavily on historical data and the quality of that data.

Over-Reliance on Predictions:

Depending too heavily on revenue predictions without considering qualitative factors or real-time changes in customer.

Complexity in Model Development:

Developing and maintaining effective revenue prediction models requires expertise in data science.

8.APPLICATIONS

1. Demand Forecasting: Predicting future demand for menu items and services.

2.Dynamic Pricing Strategies: Adjusting menu prices based on predicted demand and market conditions.

3.Menu Engineering: Analyzing the performance of menu items and optimizing the menu based on predicted revenue potential.

4. Staffing Optimization: Forecasting customer traffic to optimize staffing levels.

5. Marketing and Promotions: Predicting revenue impacts of marketing campaigns and promotional activities.

9.CONCLUSION

While restaurant revenue prediction offers significant advantages in terms of resource planning, strategic decision-making, and customer satisfaction, it also comes with challenges related to data quality, complexity, and potential risks. Restaurants should approach revenue prediction as a tool that complements qualitative insights and real-time observations to make informed decisions and maintain operational efficiency.

However, the implementation of revenue prediction systems requires careful consideration of data quality, model accuracy, and ethical considerations regarding customer privacy. Restaurants must ensure robust data integration, appropriate model selection, and ongoing validation to maintain reliability and relevance in dynamic market conditions.

Ultimately, effective utilization of revenue prediction empowers restaurants to adapt swiftly to market changes, optimize customer experiences, and sustain competitive advantage in the increasingly complex and competitive food service industry.

10. FUTURE SCOPE

The future scope of restaurant revenue prediction holds tremendous potential, driven by advancements in technology, data analytics, and changing consumer behaviours. Here are several areas where we can expect significant developments
and innovations:

1.Integration of AI and Machine Learning:

- **Enhanced Predictive Models:** Future advancements will likely lead to more sophisticated machine learning algorithms capable of processing vast amounts of data in real-time. These models can adapt dynamically to changing consumer preferences, economic conditions, and external factors like weather patterns.

2.Big Data and IoT Integration:

- **Real-Time Data Analytics:** Integration of IoT devices (e.g., smart appliances, POS systems, customer feedback systems) with big data analytics will enable restaurants to gather and analyzer real-time data more
effectively. This can lead to more accurate predictions and immediate adjustments in operational strategies.

3.Enhanced Customer Insights:

- **Sentiment Analysis:** Advanced sentiment analysis techniques can extract valuable insights from customer reviews, social media interactions, and online feedback platforms. This can help restaurants understand customer sentiment, preferences, and emerging trends more comprehensively.

4.Predictive Analytics for Sustainable Practices:

- **Resource Optimization:** Predictive analytics can assist restaurants in optimizing resource usage, such as energy consumption and food waste management. This contributes to sustainability goals and reduces operational costs.

11. BIBILOGRAPHY

[1] A. Lasek, N. Cercone, and J. Saunders, “Restaurant Sales and Customer Demand Forecasting: Literature Survey and Categorization of Methods,” in Smart City 360°, A. Leon-Garcia, R. Lenort, D.
Holman.

[2] V. Cerqueira, L. Torgo, and C. Soares, “Machine Learning vs Statistical Methods for Time Series Forecasting: Size Matters,” Sep. 2019, arXiv:1909.13316 [cs, stat].

[3] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, “Statistical and Machine Learning forecasting methods: Concerns and ways forward,” PLOS ONE, vol. 13, no. 3, p. e0194889, Mar. 2018.

[4] C. M. Galanakis, M. Rizou, T. M. S. Aldawoud, I. Ucak, and N. J. Rowan, “Innovations and technology disruptions in the food sector within the COVID-19 pandemic and post-lockdown era,” Trends in Food Science & Technology, vol. 110, pp. 193–200, Apr. 2021

[5] C. Strotmann, V. Baur, N. Börnert, and P. Gerwin, “Generation and prevention of food waste in the German food service sector in the COVID-19 pandemic – Digital approaches to encounter the pandemic related crisis,” Socio-Economic Planning Sciences, vol. 82, p. 101104, Aug. 2022

[6] V. Gudivada, A. Apon, and J. Ding, “Data Quality Considerations for Big Data and Machine Learning: Going Beyond Data Cleaning and Transformations,” International Journal on Advances in Software, vol. 10, pp. 1–20, Jul. 2017.

[7] T. Westmeier, D. Botache, M. Bieshaar, and B. Sick, “Generating Synthetic Time Series for Machine-Learning-Empowered Monitoring of Electric Motor Test Benches,” in 2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA), Oct. 2022, pp. 1–10.

[8] Q. Wen, L. Sun, F. Yang, X. Song, J. Gao, X. Wang, and H. Xu, “Time Series Data Augmentation for Deep Learning: A Survey,” in Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, Aug. 2021, pp. 4653–4660, arXiv:2002.

[9] xgboost developers, “Xgboost documentation,” 2022. [Online].

[10] B. F. Canziani, B. Almanza, R. E. Frash, M. J. McKeig, and C. Sullivan-Reid, “Classifying restaurants to improve usability of restaurant research,” International Journal of Contemporary Hospitality Management, vol. 28, no. 7, pp. 1467–1483, Jan. 2016, publisher: Emerald Group Publishing Limited.

[11] A. Debnath, G. Waghmare, H. Wadhwa, S. Asthana, and A. Ankur, “Exploring generative data augmentation in multivariate time series forecasting: Opportunities and challenges.” Mile TS ’21: 7th KDD Workshop on Mining and Learning from Time Series, no. 7, p. 7, 2021.

[12] S. Yang, W. Xiao, M. Zhang, S. Guo, J. Zhao, and F. Shen, “Image Data Augmentation for Deep Learning: A Survey,” Apr. 2022, arXiv:2204.08610.

[13] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, “Domain randomization fortransferring deep neural networks from simulation to the real world,” in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sep. 2017, pp. 23–30, iSSN: 2153-0866.

[14] Q. Wen, J. Gao, X. Song, L. Sun, H. Xu, and S. Zhu, “RobustSTL: A Robust Seasonal-Trend Decomposition Algorithm for Long Time Series,” Dec. 2018, arXiv:1812.01767 [cs, eess, stat].

12.APPENDIX

Model building:

1)Dataset

2)Google colab and VS code Application Building

1. HTML file (Index file, Input file, output file)
1. CSS file
2. Models in pickle format

SOURCE CODE:

Index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">
  <title>Forecasting Feasts: A Culinary Journey into Restaurant Revenue
Prediction</title>
  <meta content="" name="description">
  <meta content="" name="keywords">

  <!-- Favicons -->
  <link href="/static/assets/img/favicon.png" rel="icon">
  <link href="/static/assets/img/apple-touch-icon.png" rel="apple-touch-icon">

  <!-- Fonts -->
  <link href="https://fonts.googleapis.com" rel="preconnect">
  <link href="https://fonts.gstatic.com" rel="preconnect" crossorigin>
  <link
href="https://fonts.googleapis.com/css2?family=Roboto:wght@100;300;400;500;700;
900&family=Inter:wght@100;200;300;400;500;600;700;800;900&family=Amatic+S
C:wght@400;700&display=swap" rel="stylesheet">

  <!-- Vendor CSS Files -->
  <link href="/static/assets/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <link href="/static/assets/vendor/bootstrap-icons/bootstrap-icons.css"
rel="stylesheet">
```

```
  <link href="/static/assets/vendor/aos/aos.css" rel="stylesheet">
  <link href="/static/assets/vendor/glightbox/css/glightbox.min.css" rel="stylesheet">
  <link href="/static/assets/vendor/swiper/swiper-bundle.min.css" rel="stylesheet">

  <!-- Main CSS File -->
  <link href="/static/assets/css/main.css" rel="stylesheet">

  <!-- Custom CSS for Background Image -->
  <style>
    body {
      background-image: url('/static/assets/img/home.jpg');
      background-size: cover;
      background-position: center;
      height: 100vh;
    }
    .hero {
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
      text-align: center;
      color: white;
      background-color: rgba(0, 0, 0, 0.5);
    }
    .hero h1 {
      font-size: 3em;
      margin-bottom: 0.5em;
    }
    .hero p {
      font-size: 1.5em;
    }
  </style>
</head>

<body class="index-page">

  <header id="header" class="header d-flex align-items-center sticky-top">
    <div class="container position-relative d-flex align-items-center justify-content-
between">
```

```html
      <a href="/" class="logo d-flex align-items-center me-auto me-xl-0">
        <h1 class="sitename">FeastForecast</h1>
        <span>.</span>
      </a>
      <nav id="navmenu" class="navmenu">
        <ul>
          <li><a href="#hero" class="active">Home</a></li>
          <li><a href="#about">About</a></li>
          <li><a href="#chefs">Chefs</a></li>
          <li><a href="#contact">Contact</a></li>
        </ul>
        <i class="mobile-nav-toggle d-xl-none bi bi-list"></i>
      </nav>
      <a class="btn-getstarted" href="/predict">Predict Revenue</a>
    </div>
  </header>

  <main class="main">
    <!-- Hero Section -->
<section id="hero" class="hero">
  <div class="container text-center">
    <h1><span style="color: #ffc107;">Welcome to</span> <span style="color:
#ffffff;">Forecasting Feasts: A Culinary Journey into Restaurant Revenue
Prediction</span></h1>
    <p class="hero-description" style="color: #ffffff;">Where culinary art meets
predictive analytics to redefine restaurant management.</p>
  </div>
</section>

<!-- /Hero Section -->


    <!-- About Section -->
   <section id="about" class="about section">
  <div class="container section-title" data-aos="fade-up">
    <h2>About Our Restaurant</h2>
    <p><span>Learn More</span> <span class="description-title">About
Us</span></p>
  </div>
  <div class="container">
```

```
  <div class="row gy-4">
    <div class="col-lg-12" data-aos="fade-up" data-aos-delay="100">
      <img src="/static/assets/img/about.jpg" class="img-fluid mb-4" alt="Restaurant Interior">
      <div class="content">
        <p class="fst-italic">
          Welcome to our restaurant, where we delight in serving you exquisite meals prepared with passion and expertise.
        </p>
        <ul>
          <li><i class="bi bi-check-circle-fill"></i> We prioritize quality ingredients and exceptional culinary techniques.</li>
          <li><i class="bi bi-check-circle-fill"></i> Our menu reflects a harmonious blend of flavors and textures.</li>
          <li><i class="bi bi-check-circle-fill"></i> Experience hospitality at its finest with our attentive staff and cozy ambiance.</li>
        </ul>
        <p>Indulge in a dining experience that celebrates food as an art form, where every dish tells a story of craftsmanship and dedication to gastronomy.</p>
      </div>
    </div>
  </div>
 </div>
</section>
<!-- /About Section -->

  <!-- Chefs Section -->
  <section id="chefs" class="chefs section">
 <div class="container section-title" data-aos="fade-up">
  <h2>Our Chefs</h2>
  <p><span>Meet Our</span> <span class="description-title">Professional Chefs</span></p>
 </div>
 <div class="container">
  <div class="row gy-4">
    <div class="col-lg-6 d-flex align-items-stretch" data-aos="fade-up" data-aos-delay="100">
      <div class="team-member">
        <div class="member-img">
```

```
            <img src="/static/assets/img/chefs/chefs-1.jpg" class="img-fluid" alt="Chef
Walter White">
            <div class="social">
              <a href="#"><i class="bi bi-twitter"></i></a>
              <a href="#"><i class="bi bi-facebook"></i></a>
              <a href="#"><i class="bi bi-instagram"></i></a>
              <a href="#"><i class="bi bi-linkedin"></i></a>
            </div>
          </div>
          <div class="member-info">
            <h4>Walter White</h4>
            <span>Master Chef</span>
            <p>Known for his innovative approach, Chef Walter White blends traditional
flavors with modern techniques to create memorable dining experiences.</p>
          </div>
        </div>
      </div>
      <div class="col-lg-6 d-flex align-items-stretch" data-aos="fade-up" data-aos-
delay="200">
        <div class="team-member">
          <div class="member-img">
            <img src="/static/assets/img/chefs/chefs-2.jpg" class="img-fluid" alt="Chef
Sarah Johnson">
            <div class="social">
              <a href="#"><i class="bi bi-twitter"></i></a>
              <a href="#"><i class="bi bi-facebook"></i></a>
              <a href="#"><i class="bi bi-instagram"></i></a>
              <a href="#"><i class="bi bi-linkedin"></i></a>
            </div>
          </div>
          <div class="member-info">
            <h4>Sarah Johnson</h4>
            <span>Patissier</span>
            <p>Pastry aficionado Sarah Johnson brings sweetness to our dishes with her
mastery of desserts, ensuring every bite is a delight.</p>
          </div>
        </div>
      </div>
    </div>
```

```html
  </div>
</section>
<!-- /Chefs Section -->

    <!-- Contact Section -->
    <section id="contact" class="contact section">
      <div class="container section-title" data-aos="fade-up">
        <h2>Contact Us</h2>
        <p><span>Get In</span> <span class="description-title">Touch</span></p>
      </div>
      <div class="container">
        <div class="row gy-3">
          <div class="col-lg-3 col-md-6 d-flex">
            <i class="bi bi-geo-alt icon"></i>
            <div class="address">
              <h4>Address</h4>
              <p>A108 Adam Street, New York, NY 535022</p>
            </div>
          </div>
          <div class="col-lg-3 col-md-6 d-flex">
            <i class="bi bi-telephone icon"></i>
            <div>
              <h4>Contact</h4>
              <p>
                <strong>Phone:</strong> +1 5589 55488 55<br>
                <strong>Email:</strong> info@example.com<br>
              </p>
            </div>
          </div>
          <


        <div class="col-lg-3 col-md-6 d-flex">
          <i class="bi bi-clock icon"></i>
          <div>
            <h4>Opening Hours</h4>
            <p>
              <strong>Mon-Sat:</strong> <span>11AM - 23PM</span><br>
              <strong>Sunday</strong>: <span>Closed</span>
```

```
          </p>
        </div>
      </div>

      <div class="col-lg-3 col-md-6">
        <h4>Follow Us</h4>
        <div class="social-links d-flex">
          <a href="#" class="twitter"><i class="bi bi-twitter-x"></i></a>
          <a href="#" class="facebook"><i class="bi bi-facebook"></i></a>
          <a href="#" class="instagram"><i class="bi bi-instagram"></i></a>
          <a href="#" class="linkedin"><i class="bi bi-linkedin"></i></a>
        </div>
      </div>

    </div>
  </div>

  <div class="container copyright text-center mt-4">
    <p>© <span>Copyright</span> <strong class="px-1 sitename">Yummy</strong>
<span>All Rights Reserved</span></p>
    <div class="credits">
      <!-- All the links in the footer should remain intact. -->
      <!-- You can delete the links only if you've purchased the pro version. -->
      <!-- Licensing information: https://bootstrapmade.com/license/ -->
      <!-- Purchase the pro version with working PHP/AJAX contact form: [buy-url] --
>
      Designed by <a href="https://bootstrapmade.com/">BootstrapMade</a>
    </div>
  </div>

</footer>

<!-- Scroll Top -->
<a href="#" id="scroll-top" class="scroll-top d-flex align-items-center justify-
content-center"><i class="bi bi-arrow-up-short"></i></a>

<!-- Preloader -->
<div id="preloader"></div>
```

```
  <!-- Vendor JS Files -->
  <script src="/static/assets/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
  <script src="/static/assets/vendor/php-email-form/validate.js"></script>
  <script src="/static/assets/vendor/aos/aos.js"></script>
  <script src="/static/assets/vendor/glightbox/js/glightbox.min.js"></script>
  <script src="/static/assets/vendor/purecounter/purecounter_vanilla.js"></script>
  <script src="/static/assets/vendor/swiper/swiper-bundle.min.js"></script>

  <!-- Main JS File -->
  <script src="/static/assets/js/main.js"></script>

</body>

</html>
```

Input.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Revenue Prediction Input</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            margin: 0;
            background-color: #f0f0f0;
        }
        .form-container {
            background-color: #fff;
            padding: 20px;
            border-radius: 8px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
            width: 300px;
```

```
        }
        h1 {
            text-align: center;
            margin-bottom: 20px;
        }
        label {
            display: block;
            margin: 10px 0 5px;
        }
        input {
            width: 100%;
            padding: 8px;
            margin-bottom: 10px;
            border: 1px solid #ccc;
            border-radius: 4px;
        }
        button {
            width: 100%;
            padding: 10px;
            background-color: #4CAF50;
            color: white;
            border: none;
            border-radius: 4px;
            cursor: pointer;
        }
        button:hover {
            background-color: #45a049;
        }
    </style>
</head>
<body>
    <div class="form-container">
        <h1>Revenue Prediction</h1>
        <form action="/predict" method="post">
            <label for="number_of_customers">Number of Customers:</label>
            <input type="number" id="number_of_customers"
name="number_of_customers" required>
            <br>
            <label for="menu_price">Menu Price:</label>
```

```
            <input type="number" id="menu_price" name="menu_price" required>
            <br>
            <label for="marketing_spend">Marketing Spend:</label>
            <input type="number" id="marketing_spend" name="marketing_spend"
required>
            <br>
            <label for="cuisine_type">Cuisine Type:</label>
            <input type="text" id="cuisine_type" name="cuisine_type" required>
            <br>
            <label for="average_customer_spending">Average Customer
Spending:</label>
            <input type="number" id="average_customer_spending"
name="average_customer_spending" required>
            <br>
            <label for="promotions">Promotions:</label>
            <input type="number" id="promotions" name="promotions" required>
            <br>
            <label for="reviews">Reviews:</label>
            <input type="number" id="reviews" name="reviews" required>
            <br>
            <button type="submit">Predict Revenue</button>
        </form>
    </div>
</body>
</html>
```

Output.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Revenue Prediction Output</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            display: flex;
            justify-content: center;
            align-items: center;
```

```
            height: 100vh;
            margin: 0;
            background: url('/static/assets/img/background.jpg') no-repeat center center
fixed;
            background-size: cover;
            color: white;
            text-align: center;
        }
        .container {
            background-color: rgba(0, 0, 0, 0.5);
            padding: 20px;
            border-radius: 8px;
        }
        a {
            position: absolute;
            top: 20px;
            left: 20px;
            color: white;
            text-decoration: none;
            font-size: 18px;
            background-color: rgba(0, 0, 0, 0.7);
            padding: 10px;
            border-radius: 4px;
        }
        a:hover {
            background-color: rgba(0, 0, 0, 0.9);
        }
    </style>
</head>
<body>
    <a href="/">Go Back</a>
    <div class="container">
        <h1>Predicted Monthly Revenue</h1>
        <p id="prediction">{{ prediction }}</p>
    </div>
</body>
</html>
```

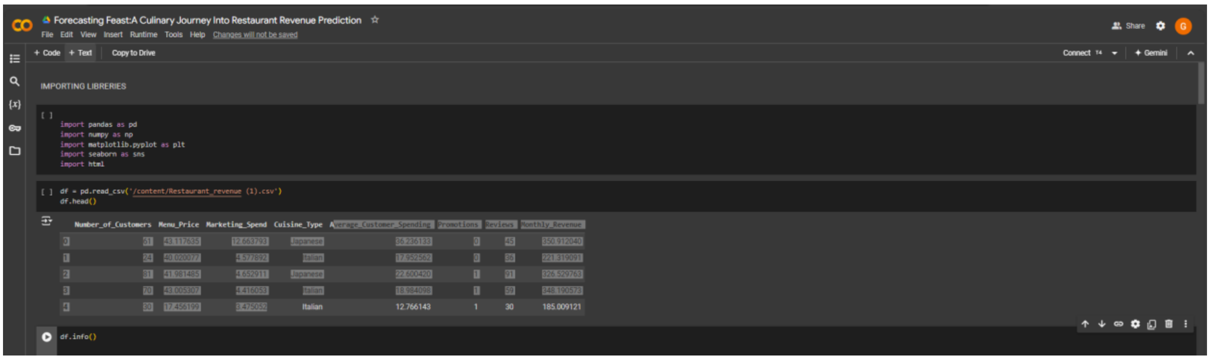
App.py

```
import numpy as np
import pickle
from flask import Flask, request, render_template

app = Flask(_name_)

# Load the pre-trained model
with open('model.pkl', 'rb') as file:
    model = pickle.load(file)

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/predict', methods=['GET', 'POST'])
def predict():
    if request.method == 'POST':
        # Get form data
        number_of_customers = float(request.form['number_of_customers'])
        menu_price = float(request.form['menu_price'])
        marketing_spend = float(request.form['marketing_spend'])
        cuisine_type = request.form['cuisine_type']
        average_customer_spending = float(request.form['average_customer_spending'])
        promotions = float(request.form['promotions'])
        reviews = float(request.form['reviews'])

        # Prepare the input data for prediction
        input_data = np.array([[number_of_customers, menu_price, marketing_spend,
average_customer_spending, promotions, reviews]])

        # Dummy example: map cuisine type to a numerical value (e.g., 0, 1, 2)
        cuisine_mapping = {'Italian': 0, 'Chinese': 1, 'Indian': 2}  # Example mapping
        if cuisine_type in cuisine_mapping:
            input_data = np.append(input_data, cuisine_mapping[cuisine_type])
        else:
            input_data = np.append(input_data, -1)  # Unknown cuisine type
```

```
        input_data = input_data.reshape(1, -1)

        # Make prediction
        prediction = model.predict(input_data)[0]

        # Render the output template with the prediction
        return render_template('output.html', prediction=prediction)

    return render_template('input.html')

if _name_ == '_main_':
    app.run(debug=True)
```

CODE SNIPPETS

MODEL BUILDING



Forecasting Feast:A Culinary Journey Into Restaurant Revenue Prediction

File Edit View Insert Runtime Tools Help Changes will not be saved

+ Code + Text Copy to Drive

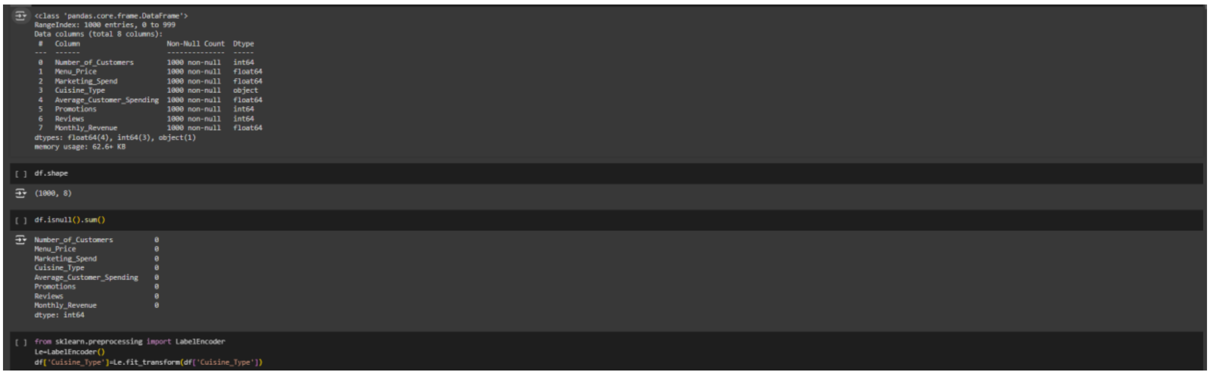
IMPORTING LIBRERIES

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import html
```

```
df = pd.read_csv('/content/Restaurant_revenue (1).csv')
df.head()
```

	Number_of_Customers	Menu_Price	Marketing_Spend	Cuisine_Type	Average_Customer_Spending	Promotions	Reviews	Monthly_Revenue
0	61	43.117635	12.663793	Japanese	36.236133	0	45	350.912040
1	24	40.020077	4.577892	Italian	17.952562	0	36	221.319091
2	81	41.981485	4.652911	Japanese	22.600420	1	91	326.529763
3	70	43.005307	4.416053	Italian	18.984098	1	59	348.190573
4	30	17.456199	3.475052	Italian	12.766143	1	30	185.009121

```
df.info()
```



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 8 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Number_of_Customers        1000 non-null   int64
 1   Menu_Price                 1000 non-null   float64
 2   Marketing_Spend            1000 non-null   float64
 3   Cuisine_Type               1000 non-null   object
 4   Average_Customer_Spending  1000 non-null   float64
 5   Promotions                 1000 non-null   int64
 6   Reviews                    1000 non-null   int64
 7   Monthly_Revenue            1000 non-null   float64
dtypes: float64(4), int64(3), object(1)
memory usage: 62.6+ KB
```

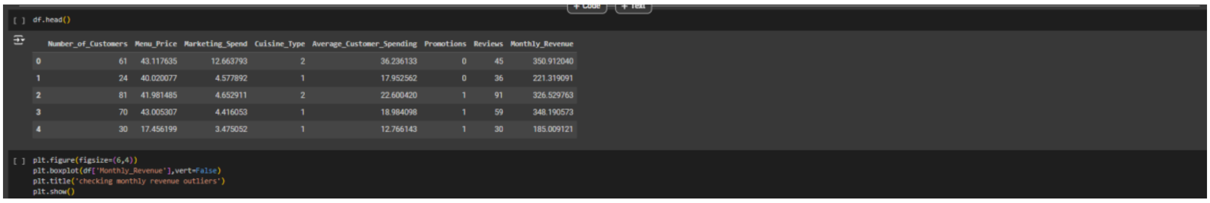
```
df.shape
```

```
(1000, 8)
```

```
df.isnull().sum()
```

```
Number_of_Customers          0
Menu_Price                   0
Marketing_Spend              0
Cuisine_Type                 0
Average_Customer_Spending    0
Promotions                   0
Reviews                      0
Monthly_Revenue              0
dtype: int64
```

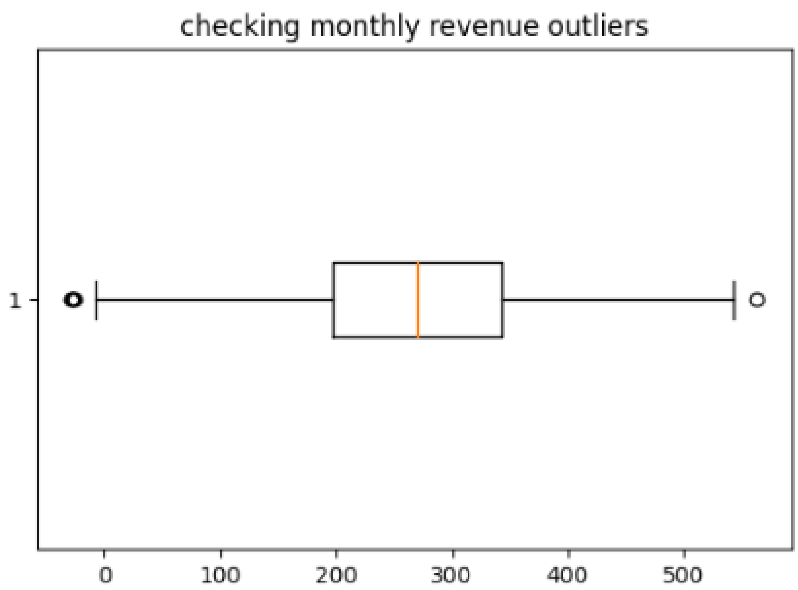
```
from sklearn.preprocessing import LabelEncoder
Le=LabelEncoder()
df['Cuisine_Type']=Le.fit_transform(df['Cuisine_Type'])
```



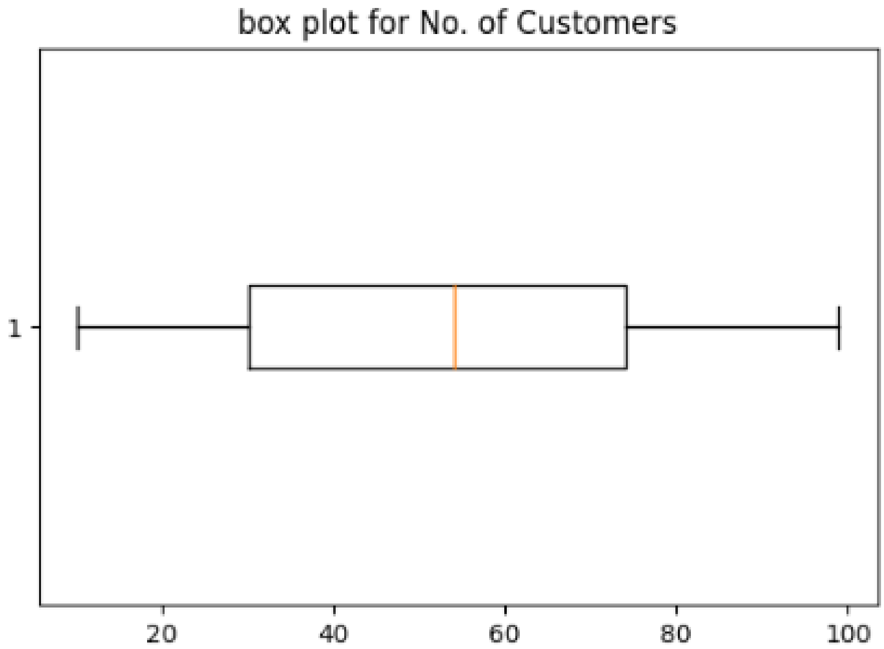
```
df.head()
```

	Number_of_Customers	Menu_Price	Marketing_Spend	Cuisine_Type	Average_Customer_Spending	Promotions	Reviews	Monthly_Revenue
0	61	43.117635	12.663793	2	36.236133	0	45	350.912040
1	24	40.020077	4.577892	1	17.952562	0	36	221.319091
2	81	41.981485	4.652911	2	22.600420	1	91	326.529763
3	70	43.005307	4.416053	1	18.984098	1	59	348.190573
4	30	17.456199	3.475052	1	12.766143	1	30	185.009121

```
plt.figure(figsize=(6,4))
plt.boxplot(df['Monthly_Revenue'],vert=False)
plt.title('checking monthly revenue outliers')
plt.show()
```



```
[ ] plt.figure(figsize=(6,4))
    plt.boxplot(df['Number_of_Customers'],vert=False)
    plt.title('box plot for No. of Customers')
    plt.show()
```




```
[ ] #dealing with outliers using winsorize method
    from scipy.stats import mstats
    df['Monthly_Revenue']=mstats.winsorize(df['Monthly_Revenue'],limits=[0.1,0.1])
```

```
[ ] df['Monthly_Revenue'].max()
```

```
404.6011165527136
```

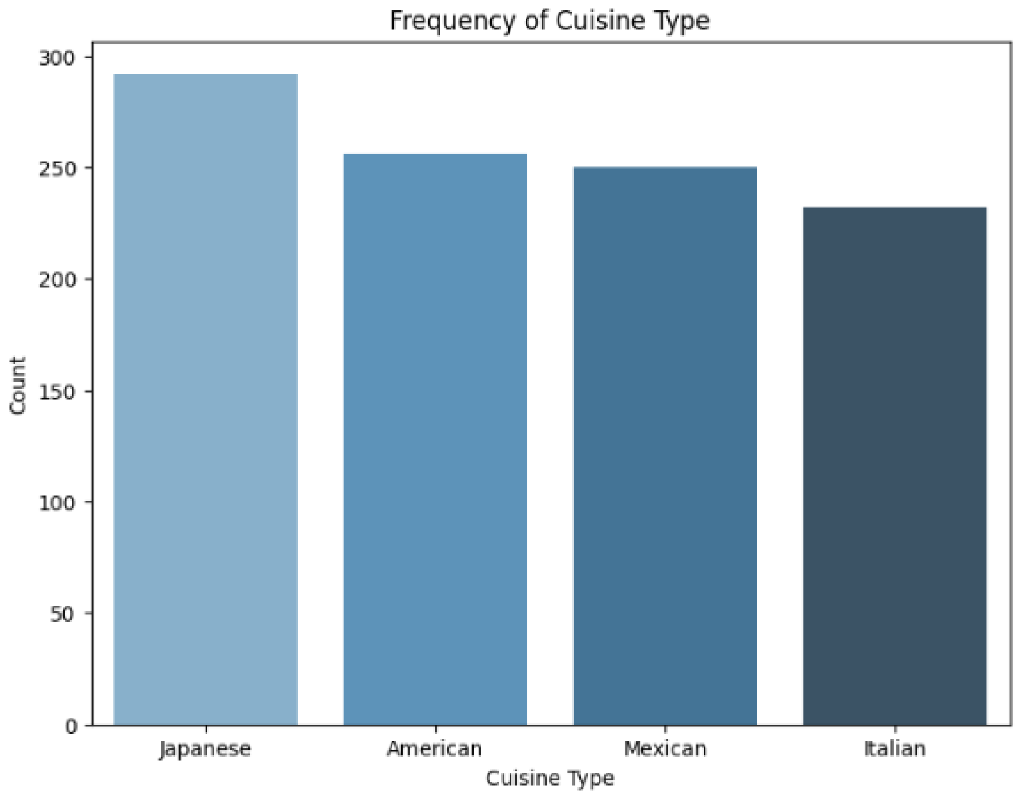
```
[ ] #checking mathematical statistics
    df.describe()
```

```
/usr/local/lib/python3.10/dist-packages/numpy/lib/function_base.py:4823: UserWarning: Warning: 'partition' will ignore the 'mask' of the MaskedArray.
  arr.partition(
/usr/local/lib/python3.10/dist-packages/numpy/lib/function_base.py:4823: UserWarning: Warning: 'partition' will ignore the 'mask' of the MaskedArray.
  arr.partition(
/usr/local/lib/python3.10/dist-packages/numpy/lib/function_base.py:4823: UserWarning: Warning: 'partition' will ignore the 'mask' of the MaskedArray.
  arr.partition(
/usr/local/lib/python3.10/dist-packages/numpy/lib/function_base.py:4823: UserWarning: Warning: 'partition' will ignore the 'mask' of the MaskedArray.
  arr.partition(
```

	Number_of_Customers	Menu_Price	Marketing_Spend	Cuisine_Type	Average_Customer_Spending	Promotions	Reviews	Monthly_Revenue
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	53.271000	30.219120	9.958726	1.506000	29.477085	0.497000	49.837000	269.086759
std	26.364914	11.278760	5.845586	1.123934	11.471686	0.500241	29.226334	87.789997
min	10.000000	10.009501	0.003768	0.000000	10.037177	0.000000	0.000000	133.475475
25%	30.000000	20.396828	4.690724	0.000000	19.603041	0.000000	24.000000	197.103642
50%	54.000000	30.860614	10.092047	2.000000	29.251365	0.000000	50.000000	270.213964
75%	74.000000	39.843868	14.992436	2.250000	39.553220	1.000000	76.000000	343.395793
max	99.000000	49.974140	19.994276	3.000000	49.900725	1.000000	99.000000	404.601117

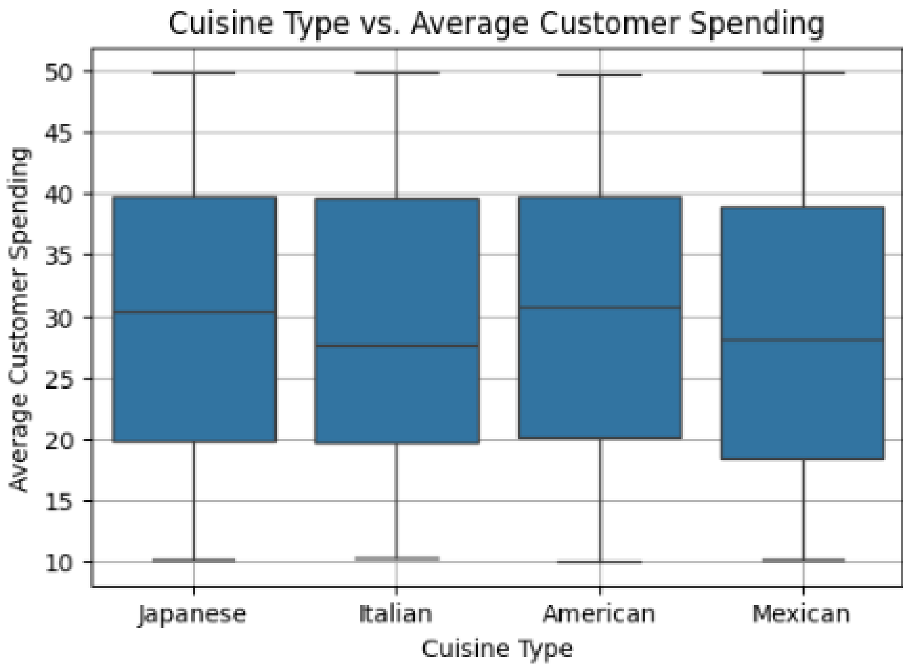
```
[ ] data = {
        'Cuisine_Type': ['Japanese', 'American', 'Mexican', 'Italian'],
        'Count': [292, 256, 250, 232]
    }
    df = pd.DataFrame(data)

    # Plot
    plt.figure(figsize=(8, 6))
    sns.barplot(x='Cuisine_Type', y='Count', data=df, palette="Blues_d", ci=None)
    plt.title('Frequency of Cuisine Type')
    plt.xlabel('Cuisine Type')
    plt.ylabel('Count')
    plt.show()
```

```
<ipython-input-140-3d4cfcdc5d28>:9: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x='Cuisine_Type', y='Count', data=df, palette="Blues_d", ci=None)
<ipython-input-140-3d4cfcdc5d28>:9: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x='Cuisine_Type', y='Count', data=df, palette="Blues_d", ci=None)
```

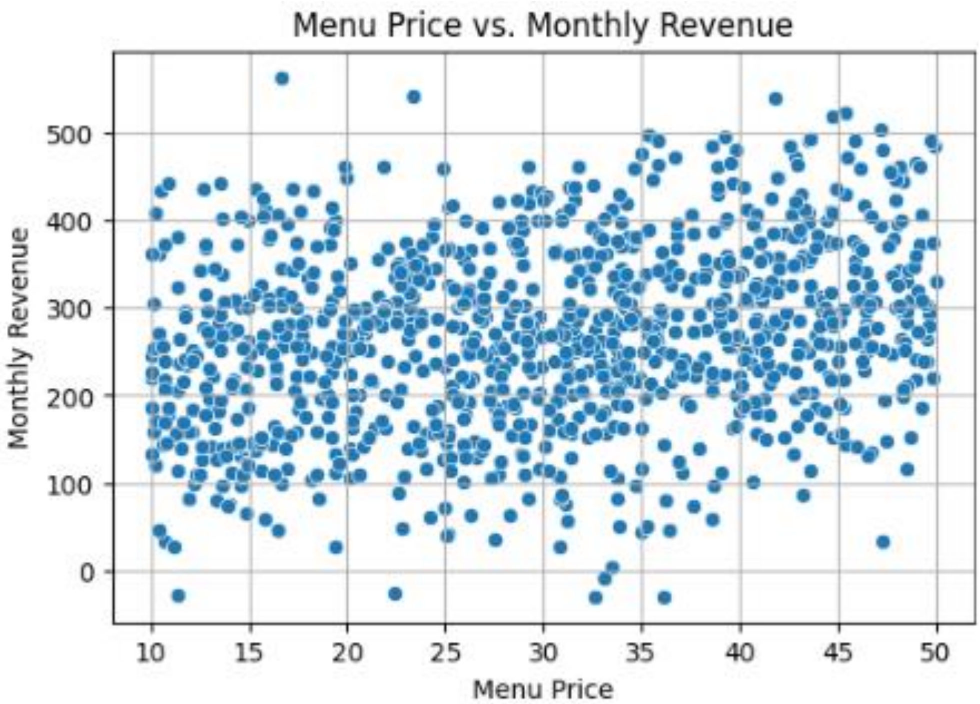


BIVARIATE ANALYSIS

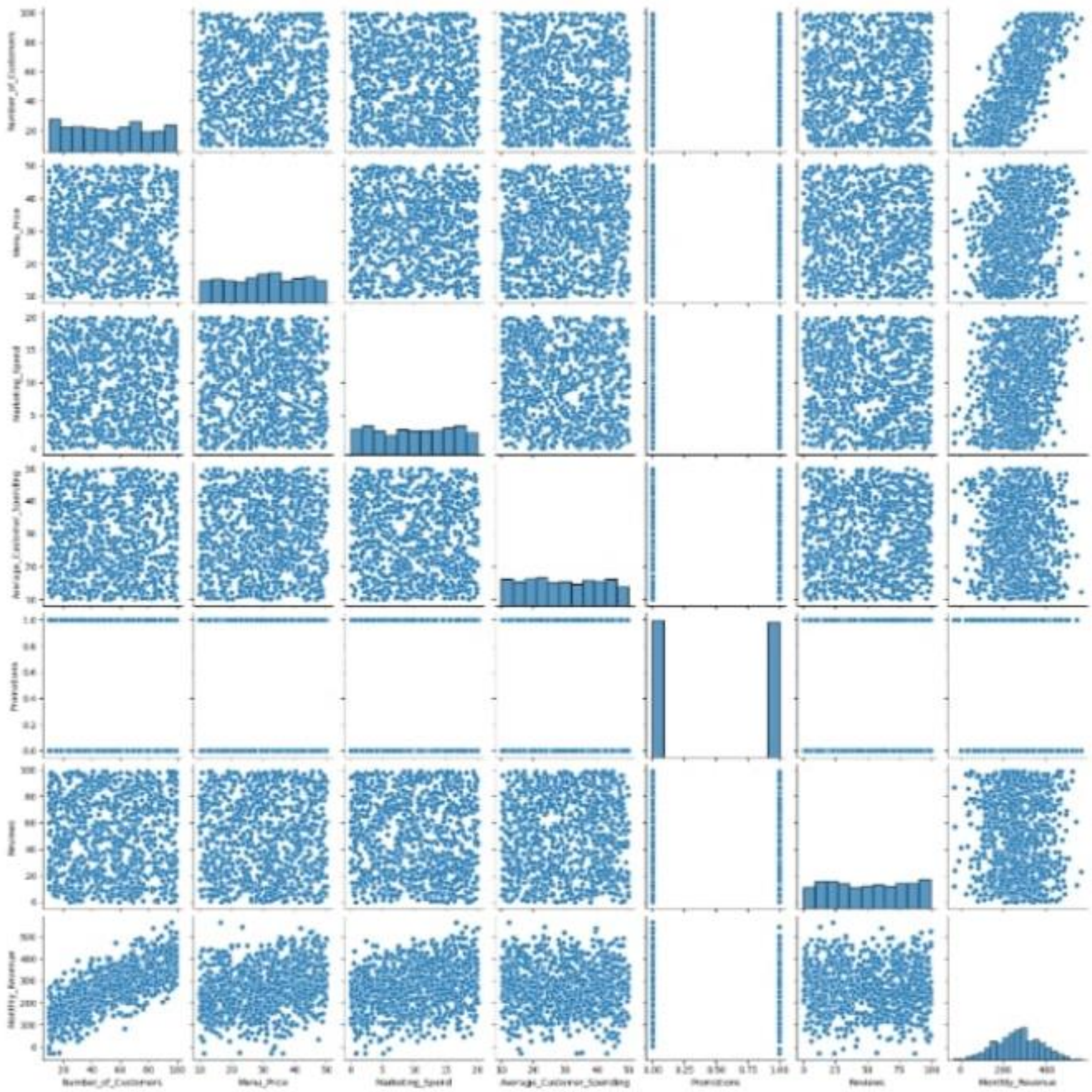
```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

# Load your data into a DataFrame and assign it to 'df_original'
df_original = pd.read_csv('/content/Restaurant_revenue (1).csv') # Replace 'your_data.csv' with the actual file path

# Or, if the DataFrame already exists, replace 'df_original' with its actual name
# df_original = existing_dataframe_name

plt.figure(figsize=(6,4))
sns.boxplot(x='Cuisine_Type', y='Average_Customer_Spending', data=df_original)
plt.title('Cuisine Type vs. Average Customer Spending')
plt.xlabel('Cuisine Type')
plt.ylabel('Average Customer Spending')
plt.grid(True)
plt.show()
```



```python
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

# Load your data into a DataFrame, ensuring it contains the 'Menu_Price' column
df = pd.read_csv('/content/Restaurant_revenue (1).csv')  # Replace 'your_data.csv' with the actual file path

# Verify if 'Menu_Price' column exists in the DataFrame
if 'Menu_Price' in df.columns:
    plt.figure(figsize=(6, 4))
    sns.scatterplot(x='Menu_Price', y='Monthly_Revenue', data=df)
    plt.title('Menu Price vs. Monthly Revenue')
    plt.xlabel('Menu Price')
    plt.ylabel('Monthly Revenue')
    plt.grid(True)
    plt.show()
else:
    print("Error: 'Menu_Price' column not found in the DataFrame.")
```



PAIRPLOT




```
[ ] X=df.drop('Monthly_Revenue',axis=1)
    X
```

	Number_of_Customers	Menu_Price	Marketing_Spend	Cuisine_Type	Average_Customer_Spending	Promotions	Reviews
0	61	43.117635	12.663793	Japanese	36.236133	0	45
1	24	40.020077	4.577892	Italian	17.952562	0	36
2	81	41.981485	4.652911	Japanese	22.600420	1	91
3	70	43.005307	4.416053	Italian	18.984098	1	59
4	30	17.456199	3.475052	Italian	12.766143	1	30
...	...	...	...	...	...	...	...
995	73	41.307842	12.122931	Japanese	19.033585	1	40
996	31	20.615496	5.822885	Mexican	17.040990	0	57
997	69	17.110656	4.141898	Japanese	44.649315	0	55
998	73	37.664722	3.046556	Japanese	27.767358	0	23
999	81	34.722067	17.989104	Italian	15.482112	1	72

1000 rows × 7 columns

```
[ ] Y=df['Monthly_Revenue']
    Y

0      350.912040
1      221.319091
2      326.529763
3      348.190573
4      185.009121
          ...
995    249.312034
996    110.228767
997    312.212552
998    272.482204
999    379.973072
Name: Monthly_Revenue, Length: 1000, dtype: float64
```

```
[ ] from sklearn.model_selection import train_test_split
```

LINEAR REGRESSION MODEL

```
[ ] from sklearn.linear_model import LinearRegression
    from sklearn.metrics import mean_squared_error
    from sklearn.metrics import r2_score

    linear_regression = LinearRegression()
    linear_regression.fit(X_train, y_train)
    y_pred = linear_regression.predict(X_test)

    mse_lr = mean_squared_error(y_test, y_pred)
    r2_lr = r2_score(y_test, y_pred)

    print('MSE: ', mse_lr)
    print('R2 score: ', r2_lr)
```

```
MSE:  3560.4388349737924
R2 score:  0.6748626487308359
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:768: UserWarning: pandas.DataFrame with sparse columns found.It will be converted to a dense numpy array.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:768: UserWarning: pandas.DataFrame with sparse columns found.It will be converted to a dense numpy array.
  warnings.warn(
```

RANDOM FOREST MODEL

```
from sklearn.ensemble import RandomForestRegressor
random_forest = RandomForestRegressor()
random_forest.fit(X_train, y_train)
y_pred = random_forest.predict(X_test)

mse_rf = mean_squared_error(y_test, y_pred)
r2_rf = r2_score(y_test, y_pred)

print('MSE: ', mse_rf)
print('R2 score: ', r2_rf)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:768: UserWarning: pandas.DataFrame with sparse columns found.It will be converted to a dense numpy array.
  warnings.warn(
MSE:  4206.9398725960655
R2 score:  0.6148792504979027
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:768: UserWarning: pandas.DataFrame with sparse columns found.It will be converted to a dense numpy array.
  warnings.warn(
```

GRADIENT BOOST MODEL

```
from sklearn.ensemble import GradientBoostingRegressor
gbr_model= GradientBoostingRegressor()
gbr_model.fit(X_train, y_train)
y_pred = gbr_model.predict(X_test)

mse_gbr = mean_squared_error(y_test, y_pred)
r2_gbr = r2_score(y_test, y_pred)

print('MSE: ', mse_gbr)
print('R2 score: ', r2_gbr)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:768: UserWarning: pandas.DataFrame with sparse columns found.It will be converted to a dense numpy array.
  warnings.warn(
MSE:  3809.6170084022474
R2 score:  0.6512518357704888
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:768: UserWarning: pandas.DataFrame with sparse columns found.It will be converted to a dense numpy array.
  warnings.warn(
```

ADABOOST MODEL

```
from sklearn.ensemble import AdaBoostRegressor
ada_model = AdaBoostRegressor()
ada_model.fit(X_train, y_train)
y_pred = ada_model.predict(X_test)

mse_ada = mean_squared_error(y_test, y_pred)
r2_ada = r2_score(y_test, y_pred)

print('MSE: ', mse_ada)
print('R2 score: ', r2_ada)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:768: UserWarning: pandas.DataFrame with sparse columns found.It will be converted to a dense numpy array.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:768: UserWarning: pandas.DataFrame with sparse columns found.It will be converted to a dense numpy array.
  warnings.warn(
MSE:  3982.1048213913766
R2 score:  0.6354616581750199
```

SUPPORT VECTOR REGRESSOR

```
from sklearn.svm import SVR
svm_model = SVR()
svm_model.fit(X_train, y_train)
y_pred = svm_model.predict(X_test)

mse_svm = mean_squared_error(y_test, y_pred)
r2_svm = r2_score(y_test, y_pred)

print('MSE: ', mse_svm)
print('R2 score: ', r2_svm)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:768: UserWarning: pandas.DataFrame with sparse columns found.It will be converted to a dense numpy array.
  warnings.warn(
MSE:  6263.137635047212
R2 score:  0.4266463668862204
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:768: UserWarning: pandas.DataFrame with sparse columns found.It will be converted to a dense numpy array.
  warnings.warn(
```

LASSO REGRESSION MODEL

```
from sklearn.linear_model import Lasso
lasso_model = Lasso()
lasso_model.fit(X_train, y_train)

mse_lasso = mean_squared_error(y_test, y_pred)
r2_lasso = r2_score(y_test, y_pred)

print('MSE: ', mse_lasso)
print('R2 score: ', r2_lasso)
```

```
MSE:  6263.137635047212
R2 score:  0.4266463668862204
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:768: UserWarning: pandas.DataFrame with sparse columns found.It will be converted to a dense numpy array.
  warnings.warn(
```

```
# Assuming 'lasso_model' is the variable holding the fitted model from previous code and
# you have data with 10 features for prediction
# Replace the example data below with your actual data with 10 features
pred = lasso_model.predict([[61, 43.117635487099967, 12.663792520269281, 0, 36.23613252393598, 0, 45, 0, 0, 0]])
print(pred)
```

```
[321.62604684]
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but Lasso was fitted with feature names
  warnings.warn(
```

```
pred
```

```
array([321.62604684])
```

```
pred = lasso_model.predict([[61, 43.117635487099967, 12.663792520269281, 0, 36.23613252393598, 0, 45, 0, 0, 0]])
print(pred)
```

```
[321.62604684]
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but Lasso was fitted with feature names
  warnings.warn(
```

```
pred
```

```
array([321.62604684])
```

```
import pandas as pd
from sklearn.metrics import r2_score, mean_squared_error

# ... (rest of your code)
# Replace these with your actual data:
y_true = [1, 2, 3, 4, 5]  # Your ground truth target values - example values
y_pred_dt = [1.1, 1.9, 3.2, 3.8, 5.1]  # Predictions from your Decision Tree Regressor - example values

r2_dt = r2_score(y_true, y_pred_dt)
mse_dt = mean_squared_error(y_true, y_pred_dt) # Calculate MSE for Decision Tree

# Make sure you have calculated r2 and mse for all other models (lr, rf, svm, gbr, ada, lasso) in a similar way before creating the DataFrame

model = pd.DataFrame({'Model':['Linear Regression','Decision Tree Regressor','RandomForestRegressor',
                              'Support vector machine','GradientBoosting Regression','AdaBoost Regression',
                               'lasso Regression'],
                      'score':[r2_lr,r2_dt,r2_rf,r2_svm,r2_gbr,r2_ada,r2_lasso],
                      'MSE':[mse_lr,mse_dt,mse_rf,mse_svm,mse_gbr,mse_ada,mse_lasso]
                      })

print(model)
```

```
                         Model     score          MSE
0            Linear Regression  0.674063  3560.438835
1      Decision Tree Regressor  0.989000     0.022000
2        RandomForestRegressor  0.614879  4206.939873
3       Support vector machine  0.426646  6263.137635
4  GradientBoosting Regression  0.651252  3809.617008
5          AdaBoost Regression  0.635462  3982.104021
6             lasso Regression  0.426646  6263.137635
```

```
model
```

	Model	score	MSE
0	Linear Regression	0.674063	3560.438835
1	Decision Tree Regressor	0.989000	0.022000
2	RandomForestRegressor	0.614879	4206.939873
3	Support vector machine	0.426646	6263.137635
4	GradientBoosting Regression	0.651252	3809.617008
5	AdaBoost Regression	0.635462	3982.104021
6	lasso Regression	0.426646	6263.137635

```
import pickle
pickle.dump(lasso_model,open('model.pkl','wb'))
```