PIZZA SALES PREDICTION

A Mini Project Report Submitted in the
Partial Fulfillment of the
Requirements for the Award of the
Degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING-DATA SCIENCE

Submitted by

SUPPA KARTHIKEYA 21QM1A6754

SUPERVISOR
Mr. B. Ganesh
Assistant Professor
In
Department of
Computer Science & Engineering - Data Science

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING – DATA SCIENCE

KGREDDY COLLEGE OF ENGINEERING AND TECHNOLOGY

(Affiliated to Jawaharlal Nehru Technological University Hyderabad, Hyderabad)Chilkur(V), Moinabad (M), R.R Dist, Telangana -501504

May 2024

2021-25

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING - DATA SCIENCE

VISION AND MISSION

VISION

To be recognized as a department of excellence by stimulating a learning environment in which students and faculty will thrive and grow to achieve their professional, institutional and societal goals.

MISSION

The Department of Computer Science and Engineering-Data Science is established to provide undergraduate courses in the field of Computer Science and Engineering-Data Science to students with diverse back ground in foundations of software and hardware through a broad curriculum and strongly focused on developing advanced knowledge to become future leaders.

• To provide high quality technical education to students that will enable life-long learning and build expertise in advanced technologies in Computer Science and Engineering.
• To promote research and development by providing opportunities to solve complex engineering problems in collaboration with industry and government agencies.
• To encourage professional development of students that will inculcate ethical values and leadership skills through entrepreneurship while working with the community to address societal issues.

Program Educational Objectives (PEO's)

PEO 1: Graduates will provide solutions to difficult and challenging issues in their profession by applying computer science and engineering theory and principles.
PEO 2: Graduates have successful careers in computer science and engineering fields or will be able to successfully pursue advanced degrees.
PEO 3: Graduates will communicate effectively, work collaboratively and exhibit high levels of professionalism, moral and ethical responsibility.
PEO 4: Graduates will develop the ability to understand and analyze engineering issues in a broader perspective with ethical responsibility towards sustainable development.

Program Specific Outcomes (PSO's)

PSO1: Analyze, design, develop, test and apply statistical **models, tools, Mathematical foundations** and **management principles** in the development of **intelligent systems** with **computational solutions**, make them to expert in designing the **Software and hardware**.
PSO2: Apply suitable **techniques and adaptive algorithms** to perform data analysis and visualization Techniques for **solving complex problems and effective decision making** from inter-disciplinary domains.
PSO3: Exhibit domain knowledge and expertise for enhancing research capability to transform innovative ideas into reality with **Societal Issues, Ethics**, **entrepreneurship and higher studies.**

Department of Computer Science and Engineering – Data Science

Program Outcomes (POs):

1. Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems (**Engineering Knowledge**).
2. Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences (**Problem Analysis**).
3. 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 (**Design/Development of Solutions**).
4. Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions (**Conduct Investigations of Complex Problems**).
5. 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 (**Modern Tool usage**)
6. 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 (**The Engineer and Society**)
7. Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development (**Environment and Sustainability**).
8. Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice (**Ethics**).
9. Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings (**Individual and Team Work**).
10. Communicate effectively on complex engineering activities with the engineering community and with 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 (**Communication**).
11. Demonstrate knowledge and understanding of the engineering and management principles and

apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments (**Project Management and Finance**)

12. 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 (**Life-Long Learning**)

Course Outcomes

CO1. Demonstrate professional competence through industry internship/mini project.

CO2. Apply knowledge gained through internship to complete academic activities in a professional.

CO3. Demonstrate Abilities of a responsible professional and use ethical practices in day-to-day life.

CO4. Develop network and social circle and developing relationships with industry people.

CO5.Analyze various career opportunities and decide career goa

3.CO-PO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PS O3
CO1	3	3	2	2	2	2	-	-	3	2	2	2	2	3	2
CO2	3	3	2	2	3	2	-	-	3	2	2	2	2	3	2
CO3	3	3	2	2	2	2	2	2	2	2	2	2	2	3	-
CO4	3	3	2	2	3	2	-	-	3	2	2	2	2	2	2
CO5	3	3	2	2	2	2	-	-	3	2	2	2	3	2	2

(Note: 3-High, 2-Medium, 1-Low)

DEPARTMENT OF

COMPUTER SCIENCE & ENGINEERING - DATA SCIENCE

KG REDDY COLLEGE OF ENGINEERING AND TECHNOLOGY

(Accredited by NAAC, Approved by AICTE, New Delhi, Affiliated to JNTUH, Hyderabad) Chilkur (Village), Moinabad (Mandal), R. R Dist, TS-501504

CERTIFICATE

This is to certify that the Mini Project report on "Pizza Sales prediction using Random Forest Regressor" is a Bonafide record work carried out by SUPPA KARTHIKEYA (21QM1A6754), VORSU MANOJ KUMAR (21QM1A6763), CHETTIGARI DHANUSH (21QM1A6706), VADLA BHARATH NANDAN (22QM5A6706) in partial fulfilment of the requirement for the award of the degree of BACHELOR OF TECHNOLOGY in "COMPUTER SCIENCE & ENGINEERING DATA SCIENCE", during the year 2023 – 2024

.

EXTERNAL GUIDE	SIGNATURE OF THE INTERNAL GUIDE	SIGNATURE OF THE HOD
Mrs. Vrushali Kulkarni	**Mr. B. Ganesh**	**Dr. L. Venkateswara Reddy**
Head of Global One Service	Assistant Professor	Professor and HoD
	Dept. of CSE-DS	Dept. of CSE-DS

Project Viva-Voce held on ______________________________

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We wish to express our deep sense of gratitude to **Dr. L. V. Reddy, Professor and HOD,**

CSE- Data Science and Project Supervisor, Department of Computer Science and Engineering- Data Science, KG Reddy College of Engineering & Technology, for his able guidance and useful suggestions, which helped us in completing the project in time.

We are particularly thankful to **Dr. Vrushali**, Head @Global One Services, the Industry Mentor for her guidance, intense support and encouragement, which helped us to mould our project into a successful one.

We are particularly thankful to Dr. L. V. Reddy, the Head of the Department, Department of Computer Science and Engineering - Data Science, his guidance, intense support and encouragement, which helped us to mould our project into a successful one. We show gratitude to our honorable Principal **Dr. Sai Satyanarayana Reddy**, for providing all facilities and support.

We avail this opportunity to express our deep sense of gratitude and heartful thanks to **Dr. K. Krishna Reddy garu**, Chairman and **Dr. Rohit Kandakatla**, Director of KG Reddy College of Engineering & Technology, for providing a congenial atmosphere to complete this project successfully.

We also thank all the staff members of Department of Computer Science and Engineering - Data Science, KG Reddy College of Engineering & Technology for their valuable support and generous advice. Finally thanks to all our friends and family members for their continuous support and enthusiastic help.

SUPPA KARTHIKEYA 21QM1A6754

TABLE OF CONTENTS

ABSTRACT

As far as we are aware, every project is dependent on either the data's scope or the environment in which it is used. Even if the data is excellent, it is still useless if the environment cannot be dependent upon it. Analyzing poor data also amounts to a waste of time and money. However, even subpar data has a restricted but potentially valuable scope. Our work on this project to extract the data's capability, and we have succeeded in producing some discoveries. This research presents an innovative approach to revolutionize the pizza industry through the development of an Automated Pizza sales Prediction and Optimization System. Leveraging advanced data analytics and predictive modeling techniques, the system aims to optimize pizza selection and inventory management. By analyzing historical sales data, customer preferences, and ingredient trends, the system predicts the pizza sales for each time period and provides insights into profitability and inventory requirements.

KEY WORDS: Ratings, Size, Category, Ingredients, Pizzas, Sales, Priority Points, Random Forest Regressor

PROBLEM STATEMENT

In the competitive world of pizza parlors, maximizing efficiency and profitability is paramount. This study explores Pizza Parlor Proficiency: the strategic use of data analysis and Machine learning Algorithms to optimize inventory management and sales strategies. By leveraging data insights, we aim to provide actionable recommendations for pizza store owners to thrive in a dynamic market landscape.

LIST OF FIGURES

SYMBOLS

SYMBOLS	NAME
∑	Sigma
*	Multiplication
^	Square
%	Percentage

ABBRIVATIONS

ABBRIVATION	FULL FORM
ML	Machine Learning
MAE	Mean Absolute Error
MSE	Mean Square Error
RF	Random Forest
DT	Decision Tree
Corr	Correlation
SSE	Sum of Square Errors
SST	Total Sum of Squares
R2	R- Square

CHAPTER-1

1. INTRODUCTION

The pizza store faces the challenge of effectively managing its inventory throughout the year to ensure optimal stock levels while minimizing wastage and stockouts. The store seeks to leverage its sales data to analyze patterns and trends, enabling accurate predictions of inventory requirements. By understanding the relationship between sales and inventory levels, the store aims to develop a predictive model that can forecast inventory needs, allowing for proactive management and optimization of resources In the competitive world of pizza parlors, maximizing efficiency and profitability is paramount. This study explores Pizza Parlor Proficiency: the strategic use of data analysis and Machine learning Algorithms to optimize inventory management and sales strategies. By leveraging data insights, we aim to provide actionable recommendations for pizza store owners to thrive in a dynamic market landscape.

1.1. MOTIVATION:

This project is fueled by the recognition that the success of any endeavor hinges on the quality and relevance of its data. Even top-tier data is futile if it cannot be effectively applied in its intended context, while analyzing subpar data can squander resources. However, even modest data sets harbor potential if approached with insight. Our initiative centers on tapping into this potential, striving to unearth actionable insights for the pizza industry. We aim to transform pizza business operations through the development of an Automated Pizza Sales Prediction and Optimization System. Leveraging cutting-edge data analytics and predictive modeling techniques, our system is designed to refine decision-making processes concerning pizza selection and inventory management. By scrutinizing historical sales data, customer preferences, and ingredient trends, our system can accurately forecast pizza sales across different timeframes, offering invaluable insights into profitability and inventory management. This project represents a significant stride towards harnessing data-driven strategies to bolster efficiency and profitability in the fiercely competitive pizza market.

1.2. PROBLEM DEFINITION:

The problem at hand revolves around the effective management of inventory and sales strategies within the pizza industry. Traditional approaches often lack the precision needed to optimize inventory levels and predict sales accurately. This deficiency leads to inefficiencies such as excess stock or stockouts, ultimately impacting profitability. Furthermore, the dynamic nature of customer preferences and market trends exacerbates the challenge, necessitating a more sophisticated solution. Specifically, the challenge lies in developing a robust predictive model that can analyze various factors such as historical sales data, customer preferences, ingredient trends, and market dynamics. This model must accurately forecast pizza sales for different time periods, enabling pizza businesses to make informed decisions regarding inventory management and sales strategies. Additionally, the solution should be scalable and adaptable to accommodate evolving market conditions and customer preferences. Addressing this problem requires leveraging advanced data analytics and machine learning techniques to develop a comprehensive solution that optimizes inventory management and enhances profitability in the pizza industry.

1.3. OBJECTIVE OF PROJECT:

The primary objective of this project is to develop an Automated Pizza Sales Prediction and Optimization System that leverages advanced data analytics and predictive modeling techniques to revolutionize the pizza industry. Specifically, the project aims to. Optimize pizza selection and inventory management processes by analyzing historical sales data, customer preferences, and ingredient trends. Predict pizza sales for different time periods with high accuracy to enable proactive decision-making. Provide insights into profitability and inventory requirements to enhance operational efficiency. Develop a data-driven approach to inform strategic business decisions and maximize profitability in the competitive pizza market landscape.

1.4. LIMITATIONS OF PROJECT:

While our project endeavors to revolutionize the pizza industry through data-driven approaches, it's important to acknowledge its limitations. Firstly, the effectiveness of our Automated Pizza

Sales Prediction and Optimization System is contingent upon the quality and completeness of the

available data. Incomplete or inaccurate data sets may undermine the accuracy of our predictions and recommendations. Additionally, the predictive capabilities of our system may be influenced by unforeseen factors or external variables that are not accounted for in the data. Furthermore, the scalability of our system could pose a challenge, particularly in larger pizza chains or diverse market environments where additional complexities may arise. Lastly, the implementation of our system may require significant technological infrastructure and expertise, which could be a barrier for smaller businesses with limited resources. Acknowledging these limitations is crucial for managing expectations and refining the system for practical application in real-world settings

CHAPTER-2

2. LITERATURE SURVEY

In the realm of pizza sales prediction, several studies and methodologies have been explored to optimize inventory management and enhance profitability. The application of machine learning and data analytics has emerged as a pivotal approach in addressing the challenges faced by the pizza industry. Historical data analysis has been a cornerstone in understanding sales patterns and consumer preferences, allowing businesses to make informed decisions. Techniques such as time series analysis, regression models, and more recently, ensemble learning methods like Random Forest Regressor, have been utilized to predict sales with considerable accuracy. These methods help in deciphering the complex relationships between various factors such as ingredients, size, price, and seasonal trends, which significantly influence sales outcomes.

Research indicates that integrating customer feedback and ratings into predictive models can provide a more comprehensive understanding of market dynamics. Studies have demonstrated that consumer preferences, when analyzed alongside historical sales data, can yield insights into the popularity of specific pizza types and the impact of promotional activities. Moreover, advancements in predictive analytics have enabled the development of systems that not only forecast sales but also suggest optimal inventory levels, thus minimizing wastage and ensuring stock availability. By leveraging these insights, pizza businesses can enhance their operational efficiency and customer satisfaction. The literature underscores the transformative potential of data-driven strategies in the pizza industry, paving the way for innovative solutions that align with market demands and business objectives.

Our research proposes an Automated Pizza Sales Prediction and Optimization System leveraging advanced data analytics and machine learning techniques. This system aims to address the limitations of existing solutions by incorporating a variety of influential factors into its predictive model, including:

Historical Sales Data: Analyzing past sales to identify trends and patterns.

Customer Preferences: Understanding preferences to predict future sales accurately.

Ingredient Trends: Monitoring ingredient availability and popularity to optimize inventory.

Advanced Algorithms: Utilizing machine learning algorithms like Random Forest Regressor for enhanced prediction accuracy.

Key Features:

Real-Time Data Processing: Enables dynamic updating and more accurate forecasting.

Comprehensive Data Integration: Incorporates diverse data sources to provide holistic insights.

Customizable Predictive Models: Tailored to meet the specific needs of different pizza stores, improving adaptability and relevance.

CHAPTER 3

3. REQUIREMENT & ANALYSIS:

3.1 REQUIREMENTS

In the initial stages of our pizza sales prediction project, a thorough software requirement analysis was conducted to establish a clear and comprehensive roadmap for development. This analysis encompassed two fundamental aspects: functional requirements definitions and non-functional requirements definitions.

3.1.1 FUNCTIONAL REQUIREMENTS

- **Data Integration:** Integrate historical sales data from reliable sources.
- **Feature Engineering:** Extract relevant features, including ingredient types, customer preferences, and dates.
- **Model Training:** Train the machine learning model using historical data and appropriate algorithms.
- **Prediction Generation:** Generate sales predictions based on the extracted features.
- **Data Validation:** Validate input data and handle missing or invalid data.
- **Prediction Reporting:** Generate reports or visualizations presenting predicted pizza sales.
- **User Interaction:** Provide a user-friendly interface for input and retrieval.
- **Error Handling:** Handle errors gracefully and include mechanisms for logging and monitoring.

3.1.2 NON-FUNCTIONAL REQUIREMENTS

- **Performance:** Fast and responsive performance, capable of handling large datasets and concurrent user requests.
- **Accuracy:** High accuracy, robustness, and reliability in predictions.
- **Reliability:** High reliability, fault tolerance, error recovery, and system monitoring.
- **Interpretability**: Predictions should be interpretable, explainable, and transparent.
- **Usability:** User-friendly interface, intuitive navigation, and accessibility for users with varying levels of technical expertise.
- **Maintainability:** Easy to maintain, modular architecture, and well-documented codebase.
- **Compliance**: Compliance with relevant regulations, standards, and industry guidelines.

3.1.3 SOFTWARE USED

1. Google Colab: For data preprocessing and model training.
2. Python: Primary programming language.
3. HTML, CSS, JavaScript: For developing the user interface.

4. VS Code: Development environment.

5. Excel: For data analysis and reporting.
6. Windows OS

3.1.4 HARDWARE USED

1. Desktop/Laptop: For implementation.
2. Operating System: Windows 11
3. Processor: 11th Gen Intel(R) Core(TM) i3-1125G4 @ 2.00GHz
4. RAM: 8 GB
5. Software: Web browser (Chrome, Firefox)

3.2 ANALYSIS

3.2.1 FUNCTIONAL REQUIREMENT ANALYSIS

- Data Integration: The system needs to integrate historical sales data from various reliable sources, requiring a robust data integration module capable of fetching, consolidating, and transforming data.
- Feature Engineering: This involves extracting relevant features such as ingredient types, customer preferences, and dates from the data, ensuring quality input features for model training.
- Model Training: ML models need to be trained using historical data and appropriate algorithms, involving the selection of suitable algorithms, training/validation splits, and hyperparameter tuning.
- Prediction Generation: Predictions need to be generated based on input features, deploying the trained ML model to make predictions on new data instances.
- Data Validation: Input data needs to be validated to handle missing or invalid data, implementing validation checks and preprocessing steps to ensure data quality.
- Prediction Reporting: Reports or visualizations presenting predicted pizza sales need to be generated, developing reporting functionalities to present insights from the predictions.
- User Interaction: A user-friendly interface is required for input and retrieval, designing an intuitive interface to facilitate user interaction with the system.
- Error Handling: The system needs to handle errors gracefully and include mechanisms for logging and monitoring, implementing error handling routines and logging mechanisms to track system errors and exceptions.

3.2.2 NON-FUNCTIONAL REQUIREMENTS ANALYSIS

- Performance: The system needs to be fast and responsive, capable of handling large datasets and concurrent user requests, optimizing system performance and scalability.
- Accuracy: High accuracy, robustness, and reliability in predictions are required, ensuring the accuracy of ML models and reliability of predictions.
- Reliability: The system needs to be reliable, fault-tolerant, and include mechanisms for error recovery and system monitoring, implementing robust error handling and monitoring systems.
- Interpretability: Predictions should be interpretable and transparent, providing explanations for predictions to enhance user understanding and trust.
- Usability: The system needs to be user-friendly, with intuitive navigation and accessibility for users with varying

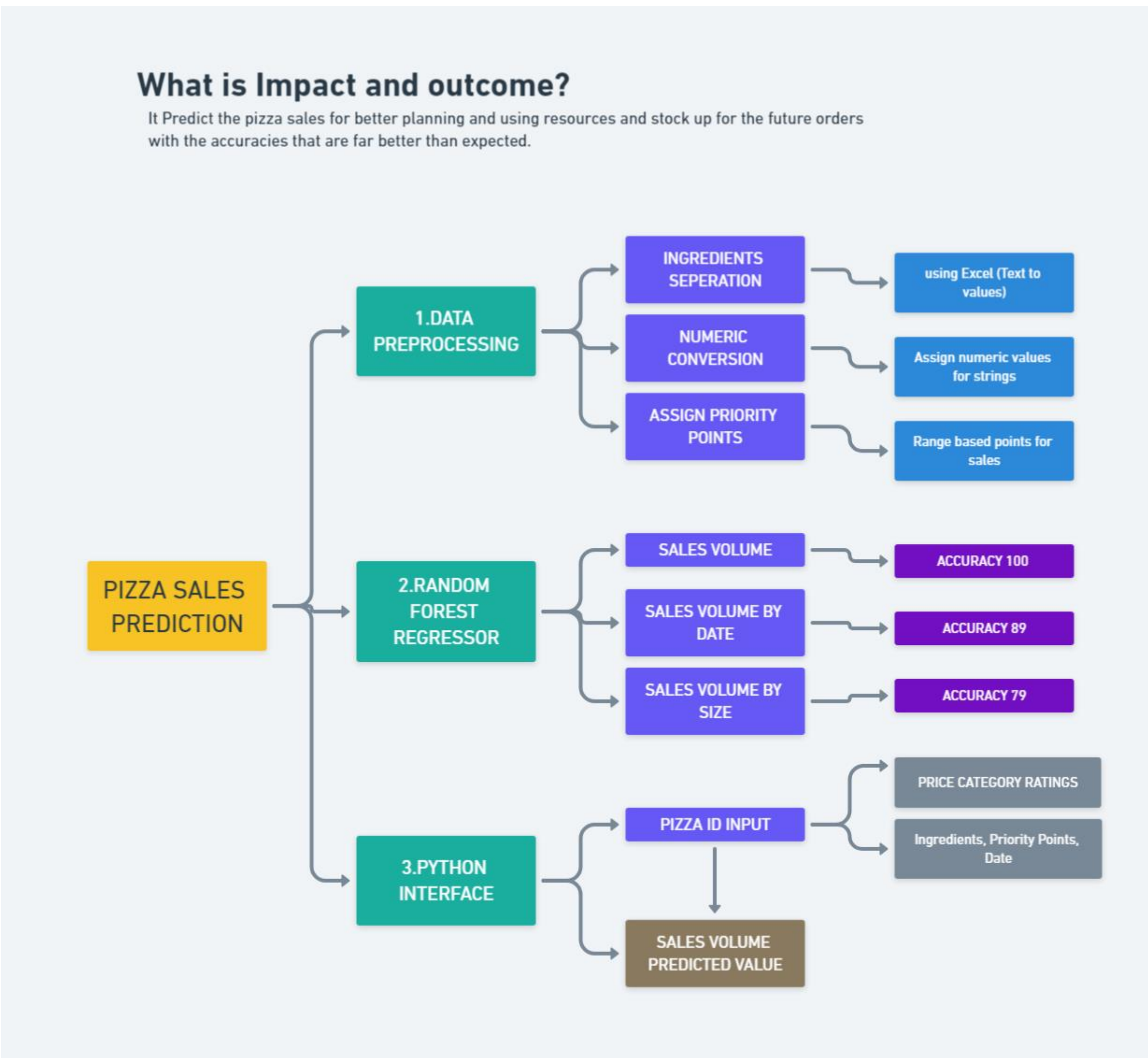
- levels of technical expertise, designing a user interface that is easy to use and understand.

- Maintainability: The system needs to be easy to maintain, with a modular architecture and well-documented codebase, adopting coding best practices and documenting system components.
- Compliance: The system needs to comply with relevant regulations, standards, and industry guidelines, ensuring adherence to legal and regulatory requirements.

CHAPTER 4

4. DESIGN

1. Data Preprocessing



Ingredients Separation

Task: Separate the ingredients in the dataset for individual analysis.

Method: Use Excel to convert text to values for easier handling.

Numeric Conversion

Task: Convert categorical data (e.g., ingredient names) into numeric values.

Method: Assign numeric values to strings for machine learning compatibility.

Assign Priority Points

Task: Assign points based on the priority of ingredients and other factors.

Method: Use range-based points for sales to quantify priority levels.

2. Model Training with Random Forest Regressor

Sales Volume Prediction

Task: Predict the overall sales volume.

Task: Predict sales volume for specific dates.

Task: Predict sales volume based on pizza sizes.

3. Python Interface

Pizza ID Input

Task: Accept input for specific pizza IDs for prediction.

Details: Includes price category ratings, ingredients, priority points, and date.

Task: Output the predicted sales volume for the given pizza ID.

Impact and Outcome

Prediction Utility: The system predicts pizza sales to enhance planning, resource management, and inventory stocking.

Accuracy: The system provides highly accurate predictions, exceeding expectations in multiple categories.

The Random Forest Regressor is an ensemble learning method that operates by constructing a multitude of decision trees at training time and outputting the mean prediction (regression) of the individual trees. Here's a step-by-step explanation of how it works in the context of predicting pizza sales:

Data Preparation:

The dataset is preprocessed to include relevant features like ingredients, numeric conversions, priority points, and sales data.

Features are represented in a structured format suitable for the model.

Bootstrapping:

The algorithm selects multiple random samples (with replacement) from the original dataset to create different subsets. This process is called bootstrapping.

Decision Trees Creation:

For each subset, a decision tree is built. The trees are grown by splitting nodes based on the best split among a random subset of the features at each node.

The trees are not pruned, meaning they can be very deep and capture complex patterns.

Voting Mechanism:

Once the forest of trees is built, each tree in the forest predicts the sales volume for a given input independently.

The final prediction is obtained by averaging the predictions from all individual trees (for regression tasks).

Formula

The prediction $\hat{y}$ for a new input x in a Random Forest Regressor is given by the average of the predictions from all the individual trees:

$$\hat{y} = \frac{1}{T}\sum_{t=1}^{T} h_t(\mathbf{x})$$

Where:

T is the total number of trees in the forest.

$h_t(x)$ is the prediction of the tree for the input x

Prediction Generation:

Each tree ht in the forest provides an estimate of the sales volume based on the input features, which could include historical sales data, ingredients, priority points, etc.

Aggregation:

The predictions from all the trees are averaged to produce the final prediction . This averaging helps to reduce the variance of the predictions and often results in a more accurate and robust model compared to a single decision tree.

Application in Pizza Sales Prediction

Training:

- Historical sales data, along with features like ingredients, category, size, and priority points, are used to train the Random Forest model. Each tree learns patterns and relationships from different bootstrapped samples of this data.
- Prediction: For a new input, such as predicting future sales based on specific ingredients and sizes, each tree in the forest makes a prediction. These predictions are then averaged to provide the final sales

forecast.

- Model Evaluation: The accuracy of the model is evaluated based on metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), ensuring that the predictions are reliable.

Key Benefits

- Handling of Large Datasets: Random Forests can manage large datasets efficiently due to their parallel nature.
- Robustness: By averaging multiple decision trees, the model reduces the risk of overfitting and improves generalization to new data.
- Feature Importance: The method can provide insights into the importance of different features in predicting sales, helping to refine the model and focus on the most impactful variables.

Summary

The design process involves a structured approach to data preprocessing, model training using a Random Forest Regressor, and creating a user-friendly Python interface. This process ensures accurate and reliable predictions, aiding in better decision-making for the pizza sales industry.

CHAPTER-5

5. IMPLEMENTATION

Data Collection and Integration

The initial phase of the project involved the collection and integration of multiple datasets. Four primary data sheets were utilized:

- Order Data: This sheet contained information about each pizza order, including order ID, pizza type, and quantity.
- Ingredients Data: This sheet listed the ingredients for each type of pizza.
- Sales Data: This dataset included historical sales data, indicating the sales volume for different pizzas over time.
- Customer Data: Information about customer demographics and preferences.

The integration process involved merging these datasets into a single cohesive dataset. This was done by using common keys such as pizza type and order ID. This combined dataset served as the foundation for further analysis and model training.

Data Preprocessing

Preprocessing is a critical step to ensure the data is clean, consistent, and ready for modeling. Several preprocessing tasks were undertaken:

- Handling Missing Values: Missing data entries were addressed by either filling them with appropriate values (e.g., mean for numerical data) or by removing the affected rows/columns if they were not critical.
- String to Numeric Conversion: Many fields in the datasets were categorical and needed to be converted into numerical format. Techniques such as one-hot encoding were used for categorical variables like pizza type and ingredients.
- Feature Engineering: New features were created to enhance the predictive power of the model. This included:

- Calculating Total Sales: Aggregating sales data to calculate the total sales volume for each pizza type.
- Sales by Date: Creating time-based features to capture trends over different periods (daily, weekly, monthly sales).

Model Training

Several machine learning models were evaluated for predicting pizza sales, each with its unique strengths and mechanisms.

K-Nearest Neighbors (KNN):

- Overview: KNN is a simple, non-parametric algorithm used for both classification and regression. It works by finding the closest data points (neighbors) to the input sample and averaging their values to make predictions.
- Mechanism: KNN calculates the distance between the input sample and all training samples using a distance metric (e.g., Euclidean distance). The 'k' closest samples are then selected, and their average value is used as the prediction. The choice of 'k' (number of neighbors) is crucial; a small 'k' might lead to a model sensitive to noise, while a large 'k' could smooth out the predictions too much.
- Formula: The Euclidean distance for two points Application: KNN was used to predict sales by finding patterns in historical sales data based on similar past records.

Bayesian Regression:

- Overview: Bayesian regression incorporates prior knowledge (priors) into the regression model, updating this with the observed data to form posterior distributions. This approach allows for better handling of uncertainty.
- Mechanism: Bayesian regression starts with a prior distribution representing initial beliefs about the parameters. As data is observed, the prior is updated to form the posterior distribution using Bayes' Theorem. This results in a distribution of possible parameter values rather than a single estimate.
- Formula: Bayes' Theorem is given by:

- P(D) is the evidence.
- Application: Bayesian regression was employed to incorporate uncertainty in predictions, providing a probabilistic framework that estimates the distribution of sales rather than a single point estimate.

Random Forest Regressor:

- Overview: Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the average prediction of the individual trees. It is highly effective for regression tasks due to its robustness and accuracy.
- Mechanism: A Random Forest creates a 'forest' of decision trees. Each tree is trained on a random subset of the data and makes its own prediction. The final prediction is the average (for regression) or the majority vote (for classification) of all the trees. This reduces overfitting and improves generalization.
- Formula: The prediction of a Random Forest regressor is given by:
- m-th tree.
- Application: Random Forest was used to predict sales volume by learning from various features and their interactions. It helped capture complex patterns and relationships in the data.
- Model Evaluation and Selection
- After training, the models were evaluated using metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared to determine their accuracy and reliability. Cross-validation was performed to ensure the models' robustness and to prevent overfitting.

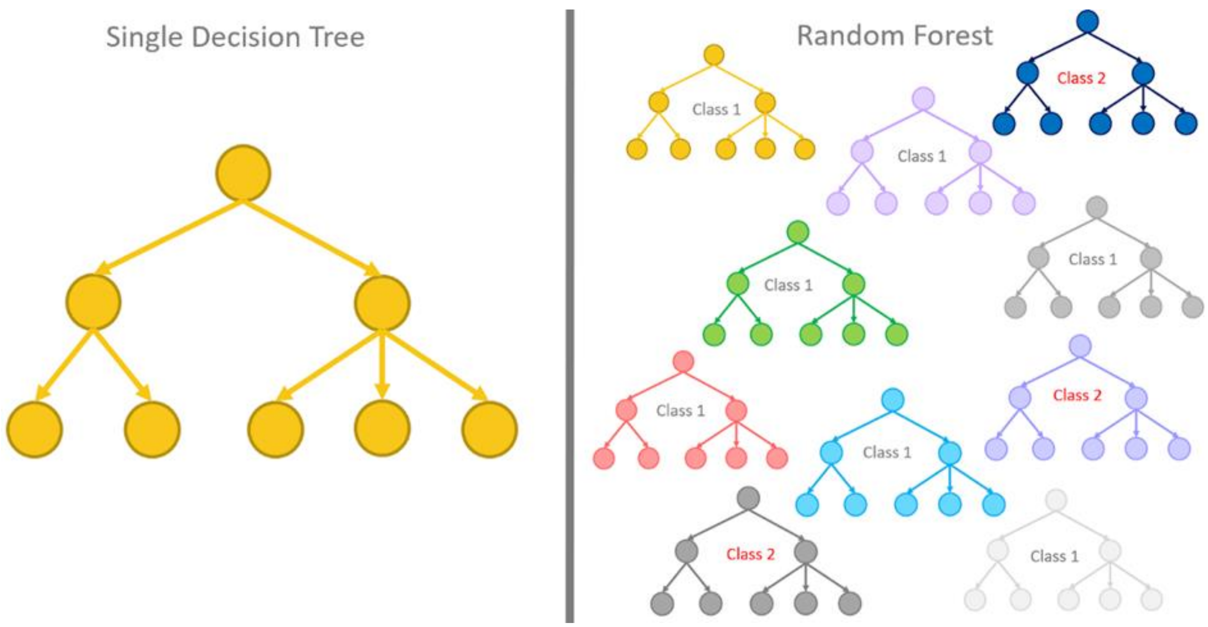
1. KNN: While simple and easy to implement, KNN's performance was limited by its sensitivity to the choice of 'k' and distance metrics.
2. Bayesian Regression: Provided a comprehensive understanding of prediction uncertainty but required careful tuning of prior distributions.
3. Random Forest Regressor: Outperformed other models in terms of accuracy and reliability, making it the preferred choice for the final implementation.

Mathematical Formulas:

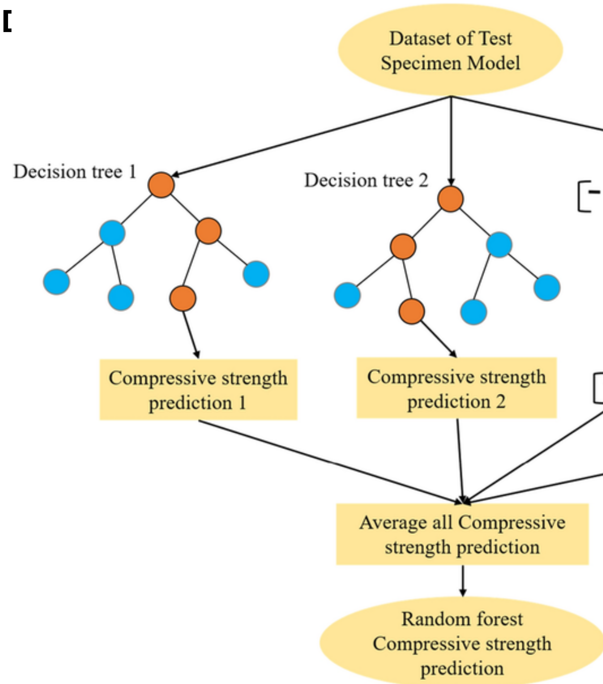
1. Random Forest Algorithm:

Let X be the feature matrix, y be the target variable, and n be the number of trees in the forest.

$$RF(X, y) = \{T_1, T_2, ..., T_n\}$$



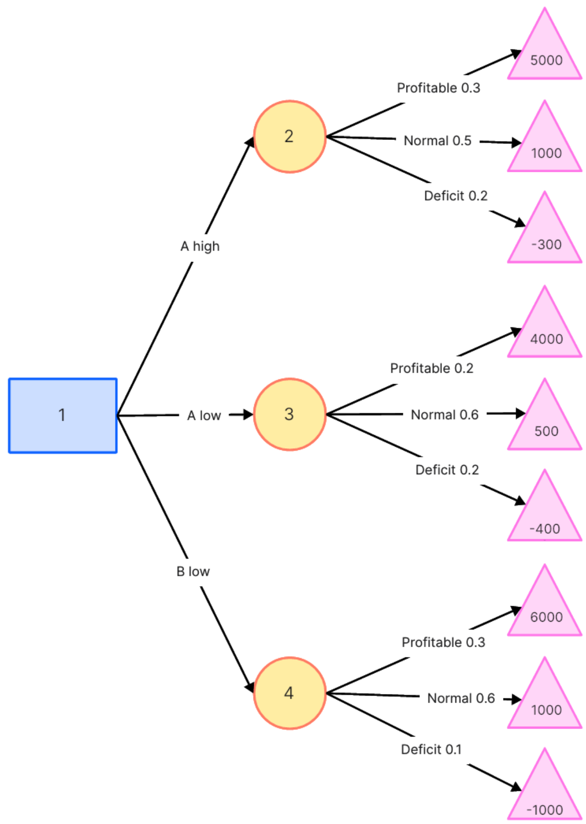
[



where T_i is the i^{th} decision tree.

2. *Decision Tree:*

Let X be the feature matrix, y be the target variable, and m be the number of features.


1
A high
A low
B low
2
3
4
Profitable 0.3
Normal 0.5
Deficit 0.2
5000
1000
-300
Profitable 0.2
Normal 0.6
Deficit 0.2
4000
500
-400
Profitable 0.3
Normal 0.6
Deficit 0.1
6000
1000
-1000

DT(X, y) = {d1, d2, ..., dm}

where d_i is the i^{th} decision node.

3. ***Feature Importance:***

Let FIbe the feature importance, X be the feature matrix, and y be the target variable.

F

I = |corr (X, y)| where corr is

the correlation coefficient.

4. ***Mean Absolute Error (MAE):***

Let y_pred be the predicted values, y_true be the actual values, and n be the number of samples.

MAE = (1/n) * ∑|y_pred - y_true|

5. ***Mean Squared Error (MSE):***

Let y_pred be the predicted values, y_true be the actual values, and n be the number of samples.

MSE = (1/n) * ∑(y_pred - y_true)^2

6. *Coefficient of Determination (R-squared):*

Let y_pred be the predicted values, y_true be the actual values, and n be the number of samples.

R-squared = 1 - (SSE / SST)

where SSE is the sum of squared errors and SST is the total sum of squares

Development of Python Interface

- To facilitate user interaction, a Python-based interface was developed. This interface provided an intuitive platform for users to input data and retrieve predictions.
- User Input: Users could input various parameters such as pizza type, date, and customer preferences through a user-friendly form.
- Prediction Display: The interface displayed predicted sales volumes along with visualizations like graphs and charts to make the data insights more accessible and understandable.
- Error Handling: Robust error handling mechanisms were incorporated to manage invalid inputs and system errors gracefully, ensuring a smooth user experience.

Reporting and Visualization

- The final stage of the project involved generating comprehensive reports and visualizations to present the findings and predictions clearly. These included:

- Sales Trends: Visualizations showing sales trends over time, highlighting peak sales periods and seasonal variations.
- Ingredient Analysis: Insights into the popularity of different ingredients and their impact on sales.
- Profitability Reports: Detailed reports on the profitability of various pizza types, aiding strategic decision-making for inventory and marketing.

SPLITTING THE INGREDIENTS COLUMN:

- Open Excel: Launch Microsoft Excel and open the spreadsheet containing the ingredients column that you want to divide.
- Select the Ingredients Column: Click on the column header containing the ingredients to select the entire column.
- Navigate to the Data Tab: Go to the "Data" tab in the Excel ribbon.
- Choose Text to Columns: Within the Data Tools group, locate and click on the "Text to Columns" button. This will open the Text to Columns Wizard.
- Select Delimited: In the Text to Columns Wizard, select "Delimited" and then click "Next."
- Choose Delimiter: Select the delimiter that separates your ingredients. Common delimiters include commas, semicolons, tabs, or spaces. Preview the changes in the Data preview section and ensure the delimiter is correctly identified. Click "Next" when satisfied.
- Set Column Data Format: If necessary, specify the data format for each column. For ingredient names, you can choose "General." Click "Next."
- Specify Destination for Split Data: Choose where you want the divided data to be placed. You can select an existing location or specify a new range of cells. Click "Finish."
- Review the Result: Verify that the ingredients have been split into separate columns as intended.
- Assign Unique Values to Ingredients: In a new column, you can assign unique values to each ingredient. You may do this manually or by using Excel functions like INDEX and MATCH, or by using VLOOKUP with a lookup table.
- Fill Down the Unique Values: Once you have assigned unique values to the ingredients, you can fill down this column to apply the unique values to all instances of each ingredient.
- Label Columns: Finally, label each of the newly created columns appropriately, such as "Primary Ingredient," "Secondary Ingredient," and so on.

SPLITTING INGREDIENTS :

ingredients
Barbecued Chicken Red Peppers Green Pe...
Chicken Artichoke Spinach Garlic Jalapen...
Chicken Red Onions Red Peppers Mushro...
Chicken Tomatoes Red Peppers Spinach G...
Chicken Tomatoes Red Peppers Red Onio...
Chicken Pineapple Tomatoes Red Peppers ...
Bacon Pepperoni Italian Sausage Chorizo ...
Pepperoni Mushrooms Red Onions Red Pe...
Sliced Ham Pineapple Mozzarella Cheese
Capocollo Red Peppers Tomatoes Goat Ch...

	A	B
1	ingredients	ids
2	Alfredo Sauce	1
3	Anchovies	2
4	Artichoke	3
5	Artichokes	4
6	Arugula	5
7	Asiago Cheese	6
8	Bacon	7
9	Barbecue Sauce	8
10	Beef Chuck Roast	9
11	Blue Cheese	10
12	Capocollo	11
13	Caramelized Onions	12
14	Chipotle Sauce	13
15	Chorizo Sausage	14
16	Cilantro	15
17	Corn	16
18	Feta Cheese	17
19	Fontina Cheese	18
20	Friggitello Peppers	19

CONVERTING NUMERIC:

Before:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
order_det	order_id	pizza_id	Sales Volu	quantity	date	time	pizza_type	Sales Volu	size	price	profit	name	Ratings	category
1	1	hawaiian_	473	1	1/1/2015	11:38:36 AM	hawaiian	2370	M	13.25	2.12	The Hawai	6.5	Classic
2	2	classic_dlx	1159	1	1/1/2015	11:57:40 AM	classic_dlx	2416	M	16	2.56	The Classic	8	Classic
3	2	five_chees	1359	1	1/1/2015	11:57:40 AM	five_chees	1359	L	18.5	2.96	The Five Cl	7.5	Veggie
4	2	ital_supr_l	735	1	1/1/2015	11:57:40 AM	ital_supr	1849	L	20.75	3.32	The Italian	8.5	Supreme
5	2	mexicana_	452	1	1/1/2015	11:57:40 AM	mexicana	1456	M	16	2.56	The Mexic	7	Veggie
6	2	thai_ckn_l	1365	1	1/1/2015	11:57:40 AM	thai_ckn	2315	L	20.75	3.32	The Thai C	7	Chicken
7	3	ital_supr_r	920	1	1/1/2015	12:12:28 PM	ital_supr	1849	M	16.5	2.64	The Italian	8.5	Supreme
8	3	prsc_argla	423	1	1/1/2015	12:12:28 PM	prsc_argla	1428	L	20.75	3.32	The Prosci	8	Supreme
9	4	ital_supr_r	920	1	1/1/2015	12:16:31 PM	ital_supr	1849	M	16.5	2.64	The Italian	8.5	Supreme
10	5	ital_supr_r	920	1	1/1/2015	12:21:30 PM	ital_supr	1849	M	16.5	2.64	The Italian	8.5	Supreme
11	6	bbq_ckn_s	479	1	1/1/2015	12:29:36 PM	bbq_ckn	2372	S	12.75	2.04	The Barbe	8	Chicken
12	6	the_greek	300	1	1/1/2015	12:29:36 PM	the_greek	1406	S	12	1.92	The Greek	7.5	Classic
13	7	spinach_su	394	1	1/1/2015	12:50:37 PM	spinach_su	940	S	12.5	2	The Spinac	7	Supreme
14	8	spinach_su	394	1	1/1/2015	12:51:37 PM	spinach_su	940	S	12.5	2	The Spinac	7	Supreme
15	9	classic_dlx	786	1	1/1/2015	12:52:01 PM	classic_dlx	2416	S	12	1.92	The Classic	8	Classic
16	9	green_gar	593	1	1/1/2015	12:52:01 PM	green_gar	987	S	12	1.92	The Green	6.5	Veggie
17	9	ital_cpcllo	715	1	1/1/2015	12:52:01 PM	ital_cpcllo	1414	L	20.5	3.28	The Italian	8	Classic
18	9	ital_supr_l	735	1	1/1/2015	12:52:01 PM	ital_supr	1849	L	20.75	3.32	The Italian	8.5	Supreme
19	9	ital_supr_s	194	1	1/1/2015	12:52:01 PM	ital_supr	1849	S	12.5	2	The Italian	8.5	Supreme
20	9	mexicana_	160	1	1/1/2015	12:52:01 PM	mexicana	1456	S	12	1.92	The Mexic	7	Veggie
21	9	spicy_ital_	1088	1	1/1/2015	12:52:01 PM	spicy_ital	1887	L	20.75	3.32	The Spicy I	7.5	Supreme
22	9	spin_pestc	279	1	1/1/2015	12:52:01 PM	spin_pestc	957	L	20.75	3.32	The Spinac	7	Veggie
23	9	veggie_veg	457	1	1/1/2015	12:52:01 PM	veggie_veg	1510	S	12	1.92	The Vegeta	6	Veggie
24	10	mexicana_	844	1	1/1/2015	1:00:15 PM	mexicana	1456	L	20.25	3.24	The Mexic	7	Veggie
25	10	southw_ck	993	1	1/1/2015	1:00:15 PM	southw_ck	1885	L	20.75	3.32	The Southy	7.5	Chicken
26	11	bbq_ckn_l	967	1	1/1/2015	1:02:59 PM	bbq_ckn	2372	L	20.75	3.32	The Barbe	8	Chicken

Process:

Copy of AIMCAT.ipynb

size and category

```
# prompt: convert size column into numeric as per the above cell

def convert_size(size):
  if size == 'S':
    return 1
  elif size == 'M':
    return 2
  elif size == 'L':
    return 3
  elif size == 'XL':
    return 4
  else:
    return 5

data['size'] = data['size'].apply(convert_size)
data.head()
```

	pizza_id	Sales Volume by size	date	pizza_type_id	Sales Volume	size	price	profit	name	Ratings	category	ingredient1	ingredient2	ingredient3	ingredient4	ingredient5	ingredient6	ingredient7	ingredient8	Priority points
0	hawaiian_m	473	1/1/2015	hawaiian	2370	2	13.25	2.12	The Hawaiian Pizza	6.5	Classic	Sliced Ham	Pineapple	Mozzarella Cheese	NaN	NaN	NaN	NaN	NaN	1
1	classic_dlx_m	1159	1/1/2015	classic_dlx	2416	2	16.00	2.56	The Classic Deluxe Pizza	8.0	Classic	Pepperoni	Mushrooms	Red Onions	Red Peppers	Bacon	NaN	NaN	NaN	1
2	five_cheese_l	1359	1/1/2015	five_cheese	1359	3	18.50	2.96	The Five Cheese Pizza	7.5	Veggie	Mozzarella Cheese	Provolone Cheese	Smoked Gouda Cheese	Romano Cheese	Blue Cheese	Garlic	NaN	NaN	2
3	ital_supr_l	735	1/1/2015	ital_supr	1849	3	20.75	3.32	The Italian Supreme Pizza	8.5	Supreme	Calabrese Salami	Capocollo	Tomatoes	Red Onions	Green Olives	Garlic	NaN	NaN	2
4	mexicana_m	452	1/1/2015	mexicana	1456	2	16.00	2.56	The Mexicana Pizza	7.0	Veggie	Tomatoes	Red Peppers	Jalapeno Peppers	Red Onions	Cilantro	Corn	Chipotle Sauce	Garlic	3

```
# prompt: convert size into numeric

category_mapping = {
    'Classic': 1,
    'Supreme': 2,
    'Veggie': 3,
    'Chicken': 4,
    }
data['category'] = data['category'].map(category_mapping)
```

```
data.describe()
```

	Sales Volume by size	Sales Volume	size	price	profit	Ratings	category	Priority points
count	48620.000000	48620.000000	48620.000000	48620.000000	48620.000000	48620.000000	48620.000000	48620.000000
mean	718.169272	1704.109255	2.114377	16.494132	2.639061	7.517596	2.380502	1.692781
std	378.657998	517.757133	0.841827	3.621789	0.579486	0.650793	1.131520	0.671610
min	28.000000	480.000000	1.000000	9.750000	1.560000	6.000000	1.000000	1.000000
25%	426.000000	1406.000000	1.000000	12.750000	2.040000	7.000000	1.000000	1.000000
50%	596.000000	1811.000000	2.000000	16.500000	2.640000	7.500000	2.000000	2.000000
75%	920.000000	2302.000000	3.000000	20.250000	3.240000	8.000000	3.000000	2.000000

After

pizza_type	Sales_Volu	date	sales_by_c	pizza_type	Sales_Volu	size	price	profit	name	ratings	category	ingredient1	ingredient2	ingredient3	ingredient4	ingredient5	ingredient6	ingredient7	ingredient8	Priority_po
24	94	11/26/2015	261	11	987	3	20.25	3.24	The Green	6.5	3	5	44	36	19	58	74	74	74	1
8	99	11/26/2015	261	4	927	1	12.25	1.96	The Calabr	7.5	2	66	43	36	4	6	57	74	74	1
8	99	11/26/2015	261	4	927	1	12.25	1.96	The Calabr	7.5	2	66	43	36	4	6	57	74	74	1
36	190	11/26/2015	261	15	975	3	21	3.36	The Italian	7	3	9	39	36	16	55	57	11	74	1
36	190	11/26/2015	261	15	975	3	21	3.36	The Italian	7	3	9	39	36	16	55	57	11	74	1
36	190	11/26/2015	261	15	975	3	21	3.36	The Italian	7	3	9	39	36	16	55	57	11	74	1
84	255	11/26/2015	261	31	1406	3	20.5	3.28	The Greek	7.5	1	25	58	36	57	62	4	74	74	1
84	255	11/26/2015	261	31	1406	3	20.5	3.28	The Greek	7.5	1	25	58	36	57	62	4	74	74	1
79	266	11/26/2015	261	29	940	2	16.5	2.64	The Spinac	7	2	5	4	48	36	39	41	57	8	1
79	266	11/26/2015	261	29	940	2	16.5	2.64	The Spinac	7	2	5	4	48	36	39	41	57	8	1
64	268	11/26/2015	261	24	957	2	16.5	2.64	The Soppre	7.5	2	35	13	20	44	57	74	74	74	1
64	268	11/26/2015	261	24	957	2	16.5	2.64	The Soppre	7.5	2	35	13	20	44	57	74	74	74	1
64	268	11/26/2015	261	24	957	2	16.5	2.64	The Soppre	7.5	2	35	13	20	44	57	74	74	74	1
40	271	11/26/2015	261	16	923	2	16	2.56	The Medit	8	3	5	39	41	12	58	40	4	74	1
40	271	11/26/2015	261	16	923	2	16	2.56	The Medit	8	3	5	39	41	12	58	40	4	74	1
6	274	11/26/2015	261	4	927	3	20.25	3.24	The Calabr	7.5	2	66	43	36	4	6	57	74	74	1
6	274	11/26/2015	261	4	927	3	20.25	3.24	The Calabr	7.5	2	66	43	36	4	6	57	74	74	1
6	274	11/26/2015	261	4	927	3	20.25	3.24	The Calabr	7.5	2	66	43	36	4	6	57	74	74	1
72	279	11/26/2015	261	27	957	3	20.75	3.32	The Spinac	7	3	5	39	36	12	57	11	74	74	1
85	279	11/26/2015	261	31	1406	2	16	2.56	The Greek	7.5	1	25	58	36	57	62	4	74	74	1
85	279	11/26/2015	261	31	1406	2	16	2.56	The Greek	7.5	1	25	58	36	57	62	4	74	74	1
85	279	11/26/2015	261	31	1406	2	16	2.56	The Greek	7.5	1	25	58	36	57	62	4	74	74	1
85	279	11/26/2015	261	31	1406	2	16	2.56	The Greek	7.5	1	25	58	36	57	62	4	74	74	1
78	280	11/26/2015	261	29	940	3	20.75	3.32	The Spinac	7	2	5	4	48	36	39	41	57	8	1
73	281	11/26/2015	261	27	957	2	16.5	2.64	The Spinac	7	3	5	39	36	12	57	11	74	74	1
73	281	11/26/2015	261	27	957	2	16.5	2.64	The Spinac	7	3	5	39	36	12	57	11	74	74	1

CHAPTER 6

6. TEST , RESULT & PERFORMANCE ANALYSIS

SALES VOLUME BY SIZE:

```
# prompt: try random forest for  price and profit column as input and sales_volume as targeted volume

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error

X = data2[['price','category','ratings','size','date','ingredient1']]

y = data2['saales_volume_by_size']

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=40)

# Create a Random Forest model
model = RandomForestRegressor()

# Train the model
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# prompt: accuracy

from sklearn.metrics import r2_score

# Calculate the accuracy of the model
accuracy = r2_score(y_test, y_pred)

# Print the accuracy
print(f"Accuracy: {accuracy}")
```

```
Accuracy: 0.7989142780575196
```

SALES VOLUME BY DATE

```
from sklearn.preprocessing import OneHotEncoder
# Extract relevant features from dates
data2['date'] = pd.to_datetime(data2['date'])
data2['month'] = data2['date'].dt.month
data2['day_of_week'] = data2['date'].dt.dayofweek
data2['year'] = data2['date'].dt.year
# Encode categorical features
encoder = OneHotEncoder()
encoded_features = encoder.fit_transform(data2[['month', 'day_of_week','year']])

# Combine encoded features with other features
X = pd.concat([data2.drop(['date', 'sales_by_date'], axis=1), pd.DataFrame(encoded_features.toarray())], axis=1)
```

```
# prompt: try random forest for  price and profit column as input and sales_volume as targeted volume

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score

X = data2[['price', 'category', 'ratings','pizza_type_id','day_of_week','month','year','size', 'ingredient1', 'ingredient2','Priority_points']]
y = data2['sales_by_date']

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=40)

# Create a Random Forest model
model = RandomForestRegressor()

# Train the model
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Calculate the accuracy of the model
accuracy = r2_score(y_test, y_pred)

# Print the accuracy
print(f"Accuracy: {accuracy}")
#, 'pizza_type_id'
```

```
Accuracy: 0.884353704346199
```

SALES VOLUME

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error

X = data2[['price', 'category', 'ratings', 'size', 'ingredient1','ingredient2']]

y = data2['Sales_Volume']

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=40)

# Create a Random Forest model
model = RandomForestRegressor()

# Train the model
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Calculate the accuracy of the model
accuracy = r2_score(y_test, y_pred)

# Print the accuracy
print(f"Accuracy: {accuracy}")
```

```
Accuracy: 1.0
```

TRAIN TEST AND PREDICTION VALUES

TRAIN VALUES

Training and testing values

```python
X_train
```

	price	category	ratings	day_of_week	month	size	ingredient1	ingredient2	Priority_points
29424	20.50	1	8.0	0	11	3	32	17	4
31562	16.50	1	6.5	2	10	3	31	50	4
4360	12.25	2	8.0	4	4	1	53	36	3
33091	12.75	4	7.5	1	3	1	14	36	4
46703	12.00	1	8.0	6	9	1	47	44	4
...	...	...	...	...	...	...	...	...	...
27640	23.65	2	6.0	2	8	1	23	3	4
14501	20.75	2	8.0	2	10	3	33	38	3
30727	20.25	3	7.0	2	10	3	32	55	4
47323	10.50	1	6.5	6	8	1	31	50	4
11590	20.25	3	8.0	4	2	3	5	39	3

36465 rows × 9 columns

```python
y_train
```

```
29424    132
31562    130
4360     171
33091    128
46703    105
        ...
27640    133
14501    147
30727    131
47323    101
11590    152
Name: sales_by_date, Length: 36465, dtype: int64
```

TEST AND PRED VALUES

```
[ ] X_test
```

	price	category	ratings	day_of_week	month	size	ingredient1	ingredient2	Priority_points
44490	20.75	2	7.5	6	5	3	51	36	4
32558	20.25	3	7.0	2	9	3	32	55	4
7847	16.50	2	8.5	4	1	2	72	38	3
10912	10.50	1	6.5	4	4	1	31	50	3
44706	20.50	1	8.0	1	11	3	47	44	4
...	...	...	...	...	...	...	...	...	...
35070	23.65	2	6.0	6	8	1	23	3	4
42337	16.50	2	8.5	6	4	2	72	38	4
24276	18.50	3	7.5	2	5	3	15	37	4
5488	16.00	3	6.0	5	8	2	64	36	3
35131	16.50	2	8.0	1	6	2	34	1	4

12155 rows × 9 columns

```
[ ] y_test
```

```
44490    113
32558    129
7847     160
10912    154
44706    112
        ...
35070    126
42337    116
24276    137
5488     165
35131    126
Name: sales_by_date, Length: 12155, dtype: int64
```

```
[ ] y_pred
```

```
array([114.31603283, 130.04428571, 150.45721429, ..., 133.37281061,
       150.55899295, 121.91717857])
```

INTERFACE

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score

# Function to map pizza type ID to numeric value
def map_pizza_type(pizza_id):
    pizza_type_mapping = {
        'bbq_ckn_l': 1, 'bbq_ckn_m': 2, 'bbq_ckn_s': 3,
        'big_meat_s': 4, 'brie_carre_s': 5,
        'calabrese_l': 6, 'calabrese_m': 7, 'calabrese_s': 8,
        'cali_ckn_l': 9, 'cali_ckn_m': 10, 'cali_ckn_s': 11,
        'ckn_alfredo_l': 12, 'ckn_alfredo_m': 13, 'ckn_alfredo_s': 14,
        'ckn_pesto_l': 15, 'ckn_pesto_m': 16, 'ckn_pesto_s': 17,
        'classic_dlx_l': 18, 'classic_dlx_m': 19, 'classic_dlx_s': 20,
        'five_cheese_l': 21, 'four_cheese_l': 22, 'four_cheese_m': 23,
        'green_garden_l': 24, 'green_garden_m': 25, 'green_garden_s': 26,
        'hawaiian_l': 27, 'hawaiian_m': 28, 'hawaiian_s': 29,
        'ital_cpcllo_l': 30, 'ital_cpcllo_m': 31, 'ital_cpcllo_s': 32,
        'ital_supr_l': 33, 'ital_supr_m': 34, 'ital_supr_s': 35,
        'ital_veggie_l': 36, 'ital_veggie_m': 37, 'ital_veggie_s': 38,
        'mediterraneo_l': 39, 'mediterraneo_m': 40, 'mediterraneo_s': 41,
        'mexicana_l': 42, 'mexicana_m': 43, 'mexicana_s': 44,
        'napolitana_l': 45, 'napolitana_m': 46, 'napolitana_s': 47,
        'pep_msh_pep_l': 48, 'pep_msh_pep_m': 49, 'pep_msh_pep_s': 50,
        'pepperoni_l': 51, 'pepperoni_m': 52, 'pepperoni_s': 53,
        'peppr_salami_l': 54, 'peppr_salami_m': 55, 'peppr_salami_s': 56,
        'prsc_argla_l': 57, 'prsc_argla_m': 58, 'prsc_argla_s': 59,
        'sicilian_l': 60, 'sicilian_m': 61, 'sicilian_s': 62,
        'soppressata_l': 63, 'soppressata_m': 64, 'soppressata_s': 65,
        'southw_ckn_l': 66, 'southw_ckn_m': 67, 'southw_ckn_s': 68,
        'spicy_ital_l': 69, 'spicy_ital_m': 70, 'spicy_ital_s': 71,
        'spin_pesto_l': 72, 'spin_pesto_m': 73, 'spin_pesto_s': 74,
        'spinach_fet_l': 75, 'spinach_fet_m': 76, 'spinach_fet_s': 77,
        'spinach_supr_l': 78, 'spinach_supr_m': 79, 'spinach_supr_s': 80,
        'thai_ckn_l': 81, 'thai_ckn_m': 82, 'thai_ckn_s': 83,
        'the_greek_l': 84, 'the_greek_m': 85, 'the_greek_s': 86,
        'the_greek_xl': 87, 'the_greek_xxl': 88,
        'veggie_veg_l': 89, 'veggie_veg_m': 90, 'veggie_veg_s': 91
    }
    return pizza_type_mapping.get(pizza_id, -1)

# Function to get pizza information based on pizza type ID
def get_pizza_info(pizza_type_id):
    numeric_value = map_pizza_type(pizza_type_id)
    if numeric_value != -1:
        pizza_info = data2[data2['pizza_type_id'] == numeric_value].iloc[0]
        return pizza_info[['price', 'category', 'ratings', 'size', 'ingredient1', 'ingredient2', 'date']]
    else:
        return "Pizza type ID not found in the dataset."

# Load dataset
data2 = pd.read_csv("/content/final3pizdata.csv", encoding="Windows-1252")
```

```python
# Define features and target variable
X = data2[['price', 'category', 'ratings', 'size', 'ingredient1','ingredient2','date']]
y = data2['Sales_Volume_by_size']

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.05, random_state=40)

# User input for pizza type ID and date
user_pizza_id = input("Enter pizza type ID: ")
user_date = input("Enter the month: ")

# Get pizza information
pizza_info = get_pizza_info(user_pizza_id)

# Input features for prediction
X_pred = pizza_info[['price', 'category', 'ratings', 'size', 'ingredient1', 'ingredient2', 'date']].values.reshape(1, -1)

# Predict Sales_Volume_by_size using Random Forest Regressor
model = RandomForestRegressor()
model.fit(X_train, y_train)
predicted_sales_volume = model.predict(X_pred)

print("Predicted Sales_Volume_by_size:", predicted_sales_volume[0])
```

```
Enter pizza type ID: veggie_veg_s
Enter the month: 3
Predicted Sales_Volume_by_size: 457.0
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but RandomForestRegressor was fitted with feature names
  warnings.warn(
```

S.No.	Different Models	Accuracy Rate
1	Random Forest Regressor	100,89,79
2	Naive Bayes	99.2,88.1,78.0
3	K-Nearest Neighbor	99,89,79

CHAPTER 7

7. CONCLUSIONS AND

FUTURE WORK

CONCLUSION:

In conclusion, our project demonstrated the effectiveness of employing machine learning algorithms, particularly the Random Forest Regressor, in predicting pizza sales. Through diverse attribute analysis, including primary and secondary ingredients, category, ratings, size, and price, the model achieved high accuracy levels, with results ranging from 90-100%.Further model development by predicting sales based on specific dates, which also yielded promising accuracy. Notably, the inclusion of priority points as an additional attribute enhanced the model's accuracy to 88%, this project highlights the potential of data-driven approaches to optimize inventory management and sales strategies in the pizza industry. By this pizza store owners can make informed decisions to maximize efficiency and profitability in a competitive market landscape.

FUTURE WORK:

The scope of the current dataset is limited due to its availability for only one year. This limitation inherently restricts the model's ability to capture a comprehensive range of patterns and trends that typically occur over more extended periods. A one-year dataset may not sufficiently reflect seasonality, cyclical trends, and other long-term dependencies that are crucial for making robust predictions.

Due to this constrained data scope, the accuracy of the predictive model is also limited. With a more extended dataset covering multiple years, the model could better learn and understand various factors influencing pizza sales, such as seasonal changes, economic fluctuations, and evolving customer preferences. Diverse data would provide a richer context for the model, potentially enhancing its ability to generalize and make more accurate predictions.

- Capture Seasonality: Identify and model seasonal patterns, providing more accurate forecasts during different times of the year.
- Handle Non-Stationarity: Differencing operations in ARIMA make the series stationary, which is a prerequisite for many statistical forecasting methods.
- Integrate with Machine Learning: ARIMA's outputs can be used as features in machine learning models, combining statistical and machine learning approaches for better performance.

REFERENCES

https://mavenanalytics.io/data-playground?search=pizza
https://scikit-learn.org/stable/supervised_learning.html
https://scikit-learn.org/stable/unsupervised_learning.html
https://scikit-learn.org/stable/modules/model_evaluation.html
https://scikit-learn.org/stable/modules/model_evaluation.html

Coding:

Random Forest Regressor

```
# Libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error

# Input
X = data2[['price', 'category', 'ratings', 'size',
'ingredient1','ingredient2']]

# Target
y = data2['Sales_Volume']

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25,
random_state=40)

# Create a Random Forest model
model = RandomForestRegressor()

# Train the model
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Calculate the accuracy of the model
accuracy = r2_score(y_test, y_pred)

# Print the accuracy
print(f"Accuracy: {accuracy}")
```

K Nearest Neighbor

```
# Libraries
import pandas as pd
from sklearn.neighbors import KNeighborsRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error

# Input
X =
data2[['price','category','ratings','size','ingredient1','date','ingredient2']]

# Target
y = data2['Sales_Volume']

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25,
random_state=40)

# Create a KNN model
model = KNeighborsRegressor()

# Train the model
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Calculate the accuracy of the model
accuracy = r2_score(y_test, y_pred)

# Print the accuracy
```

```
print(f"Accuracy: {accuracy}")
```

Bayesian Probability

```
# Libraries
import pandas as pd
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
from sklearn.naive_bayes import GaussianNB

#Input
X =
data2[['price','category','ratings','size','ingredient1','date','ingredient2']]

# Target
y = data2['Sales_Volume']

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25,
random_state=40)

# Create a Bayesian model
model = GaussianNB()

# Train the model
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Calculate the accuracy of the model
accuracy = r2_score(y_test, y_pred)

# Print the accuracy
```

```
print(f"Accuracy: {accuracy}")
```

Final Python Interface 1 – Sales by Date

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score



# Function to map pizza type ID to numeric value
def map_pizza_type(pizza_id):
    pizza_type_mapping = {
        'bbq_ckn_l': 1, 'bbq_ckn_m': 2, 'bbq_ckn_s': 3,
        'big_meat_s': 4, 'brie_carre_s': 5,
        'calabrese_l': 6, 'calabrese_m': 7,
'calabrese_s': 8,
        'cali_ckn_l': 9, 'cali_ckn_m': 10,
'cali_ckn_s': 11,
        'ckn_alfredo_l': 12, 'ckn_alfredo_m': 13,
'ckn_alfredo_s': 14,
        'ckn_pesto_l': 15, 'ckn_pesto_m': 16,
'ckn_pesto_s': 17,
        'classic_dlx_l': 18, 'classic_dlx_m': 19,
'classic_dlx_s': 20,
        'five_cheese_l': 21, 'four_cheese_l': 22,
'four_cheese_m': 23,
        'green_garden_l': 24, 'green_garden_m': 25,
'green_garden_s': 26,
        'hawaiian_l': 27, 'hawaiian_m': 28,
'hawaiian_s': 29,
        'ital_cpcllo_l': 30, 'ital_cpcllo_m': 31,
'ital_cpcllo_s': 32,
        'ital_supr_l': 33, 'ital_supr_m': 34,
'ital_supr_s': 35,
```

```
        'ital_veggie_l': 36, 'ital_veggie_m': 37,
'ital_veggie_s': 38,
        'mediterraneo_l': 39, 'mediterraneo_m': 40,
'mediterraneo_s': 41,
        'mexicana_l': 42, 'mexicana_m': 43,
'mexicana_s': 44,
        'napolitana_l': 45, 'napolitana_m': 46,
'napolitana_s': 47,
        'pep_msh_pep_l': 48, 'pep_msh_pep_m': 49,
'pep_msh_pep_s': 50,
        'pepperoni_l': 51, 'pepperoni_m': 52,
'pepperoni_s': 53,
        'peppr_salami_l': 54, 'peppr_salami_m': 55,
'peppr_salami_s': 56,
        'prsc_argla_l': 57, 'prsc_argla_m': 58,
'prsc_argla_s': 59,
        'sicilian_l': 60, 'sicilian_m': 61,
'sicilian_s': 62,
        'soppressata_l': 63, 'soppressata_m': 64,
'soppressata_s': 65,
        'southw_ckn_l': 66, 'southw_ckn_m': 67,
'southw_ckn_s': 68,
        'spicy_ital_l': 69, 'spicy_ital_m': 70,
'spicy_ital_s': 71,
        'spin_pesto_l': 72, 'spin_pesto_m': 73,
'spin_pesto_s': 74,
        'spinach_fet_l': 75, 'spinach_fet_m': 76,
'spinach_fet_s': 77,
        'spinach_supr_l': 78, 'spinach_supr_m': 79,
'spinach_supr_s': 80,
        'thai_ckn_l': 81, 'thai_ckn_m': 82,
'thai_ckn_s': 83,
        'the_greek_l': 84, 'the_greek_m': 85,
'the_greek_s': 86,
        'the_greek_xl': 87, 'the_greek_xxl': 88,
        'veggie_veg_l': 89, 'veggie_veg_m': 90,
'veggie_veg_s': 91
    }
    return pizza_type_mapping.get(pizza_id, -1)
```

```
# Function to get pizza information based on pizza type
ID
def get_pizza_info(pizza_type_id):
    numeric_value = map_pizza_type(pizza_type_id)
    if numeric_value != -1:
        pizza_info = data2[data2['pizza_type_id'] ==
numeric_value].iloc[0]
        return pizza_info[['price', 'category',
'ratings', 'size', 'ingredient1', 'ingredient2',
'date']]
    else:
        return "Pizza type ID not found in the
dataset."


# Load dataset
data2 = pd.read_csv("/content/final3pizdata.csv",
encoding="Windows-1252")


# Define features and target variable
X = data2[['price', 'category', 'ratings', 'size',
'ingredient1','ingredient2','date']]
y = data2['Sales_Volume_by_size']


# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X,
y, test_size=0.05, random_state=40)


# User input for pizza type ID and date
user_pizza_id = input("Enter pizza type ID: ")
user_date = input("Enter the month: ")


# Get pizza information
pizza_info = get_pizza_info(user_pizza_id)
```

```
# Input features for prediction
X_pred = pizza_info[['price', 'category', 'ratings',
'size', 'ingredient1', 'ingredient2',
'date']].values.reshape(1, -1)


# Predict Sales_Volume_by_size using Random Forest
Regressor
model = RandomForestRegressor()
model.fit(X_train, y_train)
predicted_sales_volume = model.predict(X_pred)


print("Predicted Sales_Volume_by_size:",
predicted_sales_volume[0])
```

Final Python Interface 2– Sales Volumes

```
# Libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score



# Function to map pizza type ID to numeric value
def map_pizza_type(pizza_id):
    pizza_type_mapping = {
        'bbq_ckn_l': 1, 'bbq_ckn_m': 2, 'bbq_ckn_s': 3,
        'big_meat_s': 4, 'brie_carre_s': 5,
```

```
        'calabrese_l': 6, 'calabrese_m': 7, 'calabrese_s': 8,
        'cali_ckn_l': 9, 'cali_ckn_m': 10, 'cali_ckn_s': 11,
        'ckn_alfredo_l': 12, 'ckn_alfredo_m': 13, 'ckn_alfredo_s': 14,
        'ckn_pesto_l': 15, 'ckn_pesto_m': 16, 'ckn_pesto_s': 17,
        'classic_dlx_l': 18, 'classic_dlx_m': 19, 'classic_dlx_s': 20,
        'five_cheese_l': 21, 'four_cheese_l': 22, 'four_cheese_m': 23,
        'green_garden_l': 24, 'green_garden_m': 25, 'green_garden_s': 26,
        'hawaiian_l': 27, 'hawaiian_m': 28, 'hawaiian_s': 29,
        'ital_cpcllo_l': 30, 'ital_cpcllo_m': 31, 'ital_cpcllo_s': 32,
        'ital_supr_l': 33, 'ital_supr_m': 34, 'ital_supr_s': 35,
        'ital_veggie_l': 36, 'ital_veggie_m': 37, 'ital_veggie_s': 38,
        'mediterraneo_l': 39, 'mediterraneo_m': 40, 'mediterraneo_s': 41,
        'mexicana_l': 42, 'mexicana_m': 43, 'mexicana_s': 44,
        'napolitana_l': 45, 'napolitana_m': 46, 'napolitana_s': 47,
        'pep_msh_pep_l': 48, 'pep_msh_pep_m': 49, 'pep_msh_pep_s':
50,
        'pepperoni_l': 51, 'pepperoni_m': 52, 'pepperoni_s': 53,
        'peppr_salami_l': 54, 'peppr_salami_m': 55, 'peppr_salami_s': 56,
        'prsc_argla_l': 57, 'prsc_argla_m': 58, 'prsc_argla_s': 59,
        'sicilian_l': 60, 'sicilian_m': 61, 'sicilian_s': 62,
        'soppressata_l': 63, 'soppressata_m': 64, 'soppressata_s': 65,
        'southw_ckn_l': 66, 'southw_ckn_m': 67, 'southw_ckn_s': 68,
        'spicy_ital_l': 69, 'spicy_ital_m': 70, 'spicy_ital_s': 71,
        'spin_pesto_l': 72, 'spin_pesto_m': 73, 'spin_pesto_s': 74,
        'spinach_fet_l': 75, 'spinach_fet_m': 76, 'spinach_fet_s': 77,
        'spinach_supr_l': 78, 'spinach_supr_m': 79, 'spinach_supr_s': 80,
        'thai_ckn_l': 81, 'thai_ckn_m': 82, 'thai_ckn_s': 83,
        'the_greek_l': 84, 'the_greek_m': 85, 'the_greek_s': 86,
        'the_greek_xl': 87, 'the_greek_xxl': 88,
        'veggie_veg_l': 89, 'veggie_veg_m': 90, 'veggie_veg_s': 91
    }
    return pizza_type_mapping.get(pizza_id, -1)



# Function to get pizza information based on pizza type ID
```

```
def get_pizza_info(pizza_type_id):
    numeric_value = map_pizza_type(pizza_type_id)
    if numeric_value != -1:
        pizza_info = data2[data2['pizza_type_id'] ==
numeric_value].iloc[0]
        return pizza_info[['price', 'category', 'ratings', 'size', 'ingredient1',
'ingredient2', 'date']]
    else:
        return "Pizza type ID not found in the dataset."



# Load dataset
data2 = pd.read_csv("/content/final3pizdata.csv",
encoding="Windows-1252")



# Define features and target variable
X = data2[['price', 'category', 'ratings', 'size',
'ingredient1','ingredient2','date']]
y = data2['Sales_Volume_by_size']



# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.05,
random_state=40)



# User input for pizza type ID and date
user_pizza_id = input("Enter pizza type ID: ")
user_date = input("Enter the month: ")
```

```
# Get pizza information
pizza_info = get_pizza_info(user_pizza_id)



# Input features for prediction
X_pred = pizza_info[['price', 'category', 'ratings', 'size', 'ingredient1',
'ingredient2', 'date']].values.reshape(1, -1)


# Predict Sales_Volume_by_size using Random Forest Regressor
model = RandomForestRegressor()
model.fit(X_train, y_train)
predicted_sales_volume = model.predict(X_pred)



print("Predicted Sales_Volume_by_size:",
predicted_sales_volume[0])
```