BLENDED ENSEMBLE LEARNING FOR DEMAND PREDICTION: AN AUTOML DRIVEN APPROACH

PHASE II REPORT

Submitted by

MOHAMED HUSSAIN S 210701161

NATHANIEL ABISHEK A 210701173

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

ANNA UNIVERSITY, CHENNAI 600 025

APRIL 2025

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Report titled **"BLENDED ENSEMBLE LEARNING FOR DEMAND PREDICTION: AN AUTOML DRIVEN APPROACH"** is the bonafide work of **MOHAMED HUSSAIN S (210701161), NATHANIEL ABISHEK A (210701173),** who carried out the work under my supervision. Certified further that to the best of myknowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE
Dr. Kumar P, M.E., Ph.D.
HEAD OF THE DEPARTMENT
Professor and Head,
Computer Science and Engineering,
Rajalakshmi Engineering College,
Rajalakshmi Nagar
Thandalam
Chennai-602 105

SIGNATURE
Dr. Senthil Pandi S, M.Tech., Ph.D.
SUPERVISOR
Associate Professor,
Computer Science and Engineering,
Rajalakshmi Engineering College,
Rajalakshmi Nagar
Thandalam
Chennai-602 105

Submitted to Project-II Viva-Voice Examination held on _____

INTERNAL EXAMINER EXTERNAL EXAMINER

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavour to put forth this report. Our sincere thanks to our Chairman **Mr. S. MEGANATHAN, B.E, F.I.E.**, our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN**, **Ph.D.,** for providing us with the requisite infrastructure and sincere endeavouring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. P. KUMAR, M.E., Ph.D.**, Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guide, **Dr. S. Senthil Pandi, M.Tech., Ph.D.**, Department of Computer Science and Engineering. Rajalakshmi Engineering College for her valuable guidance throughout the course of the project. We are glad to thank our Project Coordinator, **Dr. T. Kumaragurubaran, M.E., Ph.D**, Department of Computer Science and Engineering for his useful tips during our review to build our project.

MOHAMED HUSSAIN S (210701161)

NATHANIEL ABISHEK A (210701173)

ABSTRACT

Leveraging AutoML with ensemble models plays a crucial role in demand prediction by automating model selection, hyperparameter tuning, and evaluation. In our previous work, we employed an AutoML-based approach to identify the best-performing model which then was selected for demand forecasting. In this extended study, we improve upon the existing methodology by identifying the top five models, out of which the three best models are ensembled to enhance prediction accuracy. Furthermore, Natural Language Processing (NLP) is integrated to enable users to query the dataset dynamically for demand insights. The integration of Streamlit for the frontend and Flask for the backend creates a user friendly web interface. Our results demonstrate that the ensemble model significantly improves predictive accuracy and outperforms traditional single-model approaches. Customizing AutoML systems to address specific industry challenges, like incorporating sector-specific variables and data patterns, will also enhance their effectiveness. Combination of the sophisticated machine learning model with an intuitive web interface, paves our project that contributes to the evolution of data-driven demand forecasting by leveraging ensemble models based on the dataset, offering a scalable and intelligent solution for real-world applications. The merging of ensemble learning with AutoML significantly plays a positive role by providing accurate, scalable and efficient demand forecasting solutions across various industries.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ACKNOWLEDGEMENT	iii
	ABSTRACT	iv
	LIST OF FIGURES	vii
	LIST OF ABREVATIONS	viii
1.	INTRODUCTION	1
	1.1 GENERAL	1
	1.2 OBJECTIVE	2
	1.3 EXISTING SYSTEM	3
	1.4 PROPOSED SYSTEM	4
2.	LITERATURE SURVEY	8
3.	SYSTEM DESIGN	10
	3.1 GENERAL	10
	3.1.1 SYSTEM FLOW DIAGRAM	10
	3.1.2 SEQUENCE DIAGRAM	11
	3.1.3 CLASS DIAGRAM	12
	3.1.4 USECASE DIAGRAM	13
	3.1.5 ARCHITECTURE DIAGRAM	14
	3.1.6 ACTIVITY DIAGRAM	15
	3.1.7 COMPONENT DIAGRAM	16
	3.1.8 COLLOBORATION DIAGRAM	17

4.	PROJECT DESCRIPTION	18
	4.1 METHODOLOGIES	18
	4.1.1 RESULT DISCUSSIONS	20
5.	CONCLUSIONS AND WORK SCHEDULE	23
	5.1 REFERENCES	24
	5.2 SCREENSHOTS	27
	5.3 APPENDIX	32

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
1	SYSTEM FLOW DIAGRAM	10
2	SEQUENCE DIAGRAM	11
3	USECASE DIAGRAM	12
4	CLASS DIAGRAM	13
5	ARCHETECTURE DIAGRAM	14
6	ACTIVITY DIAGRAM	15
7	COMPONENT DIAGRAM	16
8	COLLABORATION DIAGRAM	17
9	DATASET UPLOAD	27
10	DATA PREPROCESSING	28
11	DATA PROFILING	29
12	METRICS COMPARISON	30
13	ENSEMBLING MODELS	31

LIST OF ABBREVATIONS

AutoML	Automated Machine Learning
EDA	Exploratory Data Analysis
NLP	Natural Language Processing
MAE	Mean Absolute Error
RSME	Root Mean Square Error
RF-SVR	Random Forest-Support Vector Regression
sMAPE	Symmetric Mean Absolute Percentage Error
SPL	Scaled Pinball Loss
LSTM	Long Short-Term Memory
SVR	Support Vector Regression
BMA`	Bayesian Model Averaging
SME	Small- and Medium-sized Enterprises
R^2	Coefficient of Determination
SVM	Support Vector Machine
MANET	Mobile Ad Hoc Network
Federated Learning	FL

CHAPTER 1

1.INTRODUCTION

1.1 GENERAL

Recent improvements in Machine Learning and Automation have significantly advanced demand prediction, with Automated Machine Learning and ensemble methods, playing a pivotal role in these innovations. AutoML simplifies complex processes like selecting the model, tuning, and preprocessing the data, which ultimately results in minimizing the requirement of manual effort and specialized supervision . By automating these tasks, businesses can develop and deploy machine learning models with better efficiency, leading to faster and more precise forecasting solutions.

Ensemble techniques, which strategically combine the predictive power of multiple individual models, offer a robust and effective approach to enhancing predictive accuracy. This approach addresses the inherent limitations of relying on a single, potentially flawed model. Instead of searching for the single "best" model, ensemble learning leverages the diverse insights gleaned from a collection of models. This aggregation of predictions not only improves overall accuracy but also enhances the stability and reliability of the forecasting process. In our prior research endeavours, our focus was primarily on identifying and utilizing a single top-performing model for demand forecasting. While this approach provided valuable insights, it did not fully capture the inherent complexities and fluctuations often present in real-world sales data.

Ensemble techniques, which combine multiple models, provide a robust approach to improve predictive accuracy by overcome the limitations of individual models. Rather than relying on a single best-performing model, ensemble learning enhances stability and reliability by leveraging insights from multiple models. In our previous research, we focused on selecting a single top-performing model for demand forecasting. While this approach yielded useful insights, it did not fully account for

the complexities and fluctuations in sales data. To address this, we refine our methodology by identifying the top five models generated by AutoML, and create an ensemble from the three most effective ones.

To overcome this limitation and achieve greater forecasting accuracy, we have refined our methodology. We now identify the top five performing models generated by AutoML and subsequently create an ensemble model from the three most effective ones. This refined approach allows us to harness a broader range of predictive signals and thus produce more robust and accurate forecasts. Integrating Natural Language Processing (NLP) enhances user interaction by allowing intuitive, natural language queries without technical expertise. Combined with AutoML and ensemble learning, this creates a scalable, accurate demand forecasting system.

1.2 OBJECTIVE

This project aims to build a smart, automated demand forecasting system using AutoML and ensemble techniques to boost prediction accuracy and deliver practical insights for business use.The proposed system focuses on:

1. Automating Model Selection and Optimization:

Utilizing AutoML frameworks to automate model selection, training, and hyperparameter tuning, significantly minimizing reliance on manual effort and specialized domain knowledge. This ensures the identification of the most suitable models for demand forecasting based on data characteristics.

2. Enhancing Prediction Accuracy through Ensemble Learning:

Rather than depending on just one model, the system employs ensemble learning techniques such as stacking, boosting, and by choosing and merging the top three AutoML models, the system harnesses the collective strengths of multiple high-performing algorithms, the system improves accuracy, reduces overfitting, and enhances overall prediction reliability.

3. Enabling Interactive and Intuitive User Querying:

To improve accessibility, Natural Language Processing (NLP) is integrated, allowing users to interact with the system using simple text-based queries. This feature enables non-technical users to extract demand insights, ask analytical questions, and obtain meaningful forecasts without requiring knowledge of complex data analytics tools.

4. Facilitating Comprehensive Data Analysis and Automated EDA:

The system incorporates automated Exploratory Data Analysis (EDA) to provide insights into correlations, seasonal trends, and anomalies within the dataset. This allows businesses to make informed decisions based on patterns detected in historical data, optimizing inventory and supply chain strategies.

5. Developing a Scalable and Efficient Web-Based Platform:

The demand prediction system is built as a user-friendly web application**,** integrating Streamlit for the frontend and Flask for the backend API. This enables users to seamlessly upload datasets, preprocess data, train models, visualize results, and download predictions through an intuitive graphical interface.

1.3 EXISTING SYSTEM

Current demand prediction systems rely on a combination of traditional statistical methods, machine learning algorithms, and enterprise tools tailored for specific industries. Statistical models, such as Linear Regression and ARIMA (Auto-Regressive Integrated Moving Average), are commonly employed for analyzing historical data and identifying trends. These models are particularly effective for straightforward, stationary datasets, making them widely used in sectors like retail, logistics, and manufacturing to forecast sales or inventory requirements. Similarly, decision trees and rule-based approaches are utilized for relatively simple forecasting tasks, where decision-making can be guided by predefined thresholds or historical averages.

Algorithms like Random Forest and SVM are often applied to cases with non-linear trends and datasets of moderate complexity.These models are often used in e-commerce, healthcare, and energy sectors to predict customer behavior, electricity demand, or resource requirements. Enterprise Resource Planning (ERP) systems and rule-based forecasting systems are also prominent in industries like supply chain and distribution, automating routine forecasting tasks and integrating demand predictions with other business processes.

While these systems serve specific purposes effectively, they face several limitations when it comes to handling the complexity of modern data. Many traditional models, including statistical ones like ARIMA, struggle to adapt to large-scale datasets or dynamic, real-time inputs. Similarly, manual efforts required for preprocessing data and tuning machine learning models make them resource-intensive and dependent on expert knowledge. Moreover, rule-based systems lack the flexibility to account for unforeseen factors like sudden market changes or demand spikes.

Despite their limitations, these systems have provided a foundation for demand prediction across industries. However, the evolving complexity of datasets, combined with the need for scalability, real-time adaptability, and improved accuracy, highlights the demand for more advanced and automated solutions.

1.4 PROPOSED SYSTEM

The project harnesses the power of machine learning, specifically AutoML, in order to build an automated, and highly efficient system that predicts demand, further augmented with Natural Language Processing (NLP) to facilitate intuitive user interactions. The entire process is broken down into numerous interconnected phases:

Data Collection and preparing:

The initial phase of the project focuses on gathering comprehensive historical sales data from a variety of sources. These sources may include retail outlets, online platforms, supermarkets, or any other relevant sales channels. The collected dataset typically encompasses essential details such as product IDs, corresponding sales volumes, dates of transactions, and relevant customer demographics. This rich dataset forms the foundation upon which the predictive models are built. Once collected, the data is meticulously structured into a standardized format, such as CSV (Comma Separated Values) or Excel, ensuring compatibility with the system and facilitating easy data ingestion. This crucial step ensures that the data used for model training is of the highest quality and reliability, ultimately contributing to the accuracy and robustness of the predictive models.

Data Processing and Exploratory Data Analysis (EDA):

With the data collected and formatted, the next phase involves a series of crucial data processing steps. These steps include missing value imputation to address any gaps in the data, outlier detection to identify and handle extreme or unusual data points, and data normalization to bring all variables to a similar scale. Categorical encoding is also performed to convert categorical variables into number notations, usable by machine learning algorithms. Simultaneously, Exploratory Data Analysis is performed using a combinations of statistical methods and visualization tools. This in-depth analysis aims to uncover hidden trends, identify significant correlations between variables, and recognize any seasonal patterns or cyclical variations present in the sales data. This optimization process ensures that the dataset is tailored for the specific model being used, thereby improving

prediction accuracy and reducing computational complexity. tools to uncover trends, correlations, and seasonal patterns in the data.

Model Selection and Ensemble Learning:

In this phase, the system employs AutoML to evaluate a wide variety of machine learning models, ranking them using metrics like MAE and RMSE. The top five models are selected, and the best three are combined using advanced ensemble techniques such as stacking, boosting, or bagging. This ensemble model holds onto the strengths of multiple algorithms to improve prediction accuracy, while reducing the risk of overfitting. Hyperparameter tuning is automated to optimize model performance and reduce computational costs. Cross-validation is used to validate the ensemble model, ensuring its robustness and reliability across different data distributions.

NLP Integration for Query-Based Predictions:

The system incorporates Natural Language Processing (NLP) in order to serve dynamic, query-based user interactions. Utilizing NLP techniques such as tokenization, named entity recognition and intent classification by the system understands and processes. Users can ask questions like "What is the sales forecast for Product A in the next quarter?" or "Why was the demand low last month?" The NLP module translates these queries into structured database commands, retrieves relevant data, and generates predictive insights, presenting the results in interactive visualizations for easy interpretation.

Web Interface Development:

To provide a user-friendly interface, the system integrates Streamlit for the interactive frontend and Flask for backend API management, model execution, and database handling. The web interface supports dataset uploads, enabling users to easily input their own data. It also provides real-time demand forecasting capabilities, allowing users to generate predictions on demand. Furthermore, the integrated NLP module allows for query-based insights, providing users with a dynamic and interactive way to explore the data and predictions. By combining the power of AutoML, ensemble learning, and NLP, this comprehensive and scalable solution enhances data-driven decision-making, enabling businesses to optimize inventory levels, streamline supply chains, and improve overall operations. As machine learning and automation technologies continue to evolve, this synergistic approach promises to drive even greater forecasting accuracy and contribute to increased business success in the future.

CHAPTER 2

2. LITERATURE SURVEY

H. Iftikhar, et al., [1] This study introduces a novel time-series ensemble technique for electricity demand forecasting, emphasizing the effectiveness of combining multiple machine learning models to capture complex consumption patterns. The research demonstrates that ensemble learning significantly enhances short-term forecasting accuracy. These findings directly validate your project's approach of leveraging ensemble techniques for demand prediction, ensuring robust and scalable forecasting solutions.

P. Kumar, et al., [2] The authors explore how ensemble learning can optimize credit scoring by improving loan approval decisions. By combining multiple models, they enhance decision reliability and precision, ensuring higher stability in financial risk assessment. This study aligns with your project's objective of integrating ensemble learning to enhance demand prediction accuracy across various industries. It further reinforces the idea that model stacking and optimization can yield better predictive insights.

Y. Zhang, et al., [3] This work transitions from traditional machine learning approaches to ensemble learning for demand forecasting, showcasing the advantages of aggregating multiple predictive models. The authors highlight that ensemble methods consistently outperform single models in forecasting reliability, adaptability, and robustness. Their findings provide strong support for your methodology of integrating advanced ensemble techniques to improve prediction accuracy across different datasets.

D. Hulak and G. Taylor, [4] This research investigates A combined ARIMA model setup tailored for short-term forecasting of electricity demand, presenting an innovative hybrid forecasting technique that merges traditional statistical methods with machine learning-based ensemble models. The insights gained from this study complement your project's methodology of leveraging AutoML-selected models to create a hybrid ensemble system that enhances demand prediction robustness.

P. Naik, et al., [5] The study explores automated ensemble modeling for biomass prediction using satellite imagery, demonstrating the power of stacking multiple models to handle complex datasets. The research highlights AutoML's role in selecting and optimizing models

to maximize predictive accuracy while minimizing manual intervention. This directly supports your methodology of using stacked ensemble techniques to refine demand forecasting, making it more adaptable to diverse and evolving datasets.

A. Garg and A. Chaudhary, [6] This paper emphasizes the importance of interpretability in machine learning models by applying AutoML and LIME for analyzing IPL auction datasets. The authors argue that explainable AI techniques are crucial for ensuring transparency in predictive analytics. Similarly, your project aims to provide users with interpretable demand forecasting outputs, allowing businesses to make informed decisions based on clear and explainable predictions.

S. P. Menon, et al., [7] The study applies AutoML techniques for brain tumor diagnosis, demonstrating how automated model selection improves classification accuracy while reducing human intervention. This aligns with your project's goal of automating demand prediction processes while ensuring high precision. The scalability and efficiency showcased in this research validate the feasibility of AutoML-driven demand forecasting systems.

S. Talapaneni, et al., [8] The authors introduce a voting ensemble model for heart disease prediction, demonstrating how combining multiple models enhances reliability and robustness. Their approach highlights the significance of model diversity in improving predictive performance. This methodology supports your strategy of using ensemble learning to enhance demand forecasting accuracy, particularly in cases where traditional single-model approaches struggle to generalize well.

K. Han, et al., [9] This research introduces a novel genetic algorithm-based ensemble selection method, optimizing multi-layered models for improved accuracy. The study demonstrates how genetic algorithms dynamically refine model selection, ensuring adaptability to different datasets. This aligns with your project's goal of continuously refining ensemble frameworks to optimize demand forecasting for various industries.

P. Kumar, et al., [10] The study applies ensemble learning for market basket analysis, showcasing its effectiveness in identifying consumer behavior patterns and optimizing retail sales. The research emphasizes that ensemble learning improves predictive power in retail analytics by integrating multiple model insights. This aligns with your project's vision of

providing businesses with actionable demand forecasts to support long-term strategic planning, inventory management, and operational efficiency.

Y. Jin, et al., [11] This work focuses on the use of stacking ensemble learning for online car-hailing demand forecasting, addressing the scalability and accuracy challenges in large-scale real-time prediction. By tackling dynamic demand fluctuations and ensuring precision in forecasted outcomes, the study reinforces your project's aim of handling diverse datasets effectively while providing highly accurate demand predictions.

V. E. Kovalevsky and N. A. Zhukova, [12] The authors highlight AutoML's adaptability in time-series forecasting tasks, particularly for dynamic and continuously evolving data. Their study emphasizes how automated model selection improves forecast precision while reducing computational overhead. This supports your project's focus on building a flexible demand prediction system that can accommodate changing datasets and fluctuating market conditions.

A. K. Sharma, et al., [13] This study integrates machine learning with time series models to enhance demand forecasting accuracy in the automotive aftermarket sector. By coupling these techniques, the authors improve predictive performance, which aligns with the objective of leveraging ensemble methods in your project to optimize demand predictions across varying datasets.

S. M. T. U. Raju, et al., [14] The authors suggest using ensemble learning for demand forecasting in the steel industry, aiming to enhance accuracy by merging several machine learning models. Their findings highlight the strength of ensemble methods in industrial use cases, supporting the use of blended ensemble strategies in your demand prediction system.

G. Duan and J. Dong, [15] This paper presents an A demand forecasting model for home appliances built on ensemble learning techniques, emphasizing the integration of multiple predictive algorithms to enhance forecast accuracy. The study's methodology aligns with your project's focus on developing an automated demand prediction system powered by ensemble learning.

A. Mitra, et al., [16] This work explores a hybrid machine learning approach for demand forecasting in a multi-channel retail environment. The authors compare different forecasting

models and propose a novel hybrid strategy to improve accuracy, supporting your project's emphasis on combining AutoML-selected top models into a powerful ensemble.

M. Q. Raza, et al., [17] The authors introduce a multivariate ensemble forecasting framework to predict demand on anomalous days. By utilizing neural networks optimized with global best particle swarm optimization (GPSO), the study provides insights into handling fluctuating demand patterns, which is relevant to your project's goal of adaptive and robust demand forecasting.

P. Pełka and G. Dudek, [18] This study employs pattern similarity-based ensemble forecasting to predict monthly electricity demand, demonstrating how historical consumption patterns can be leveraged for improved forecasting. The research Emphasizes the value of leveraging historical data trends in predictive modeling, aligning well with your project's approach to demand forecasting

Y. Hmamouche, et al., [19] The authors propose a scalable framework for large-scale time series prediction, focusing on handling high-dimensional datasets efficiently. Their work supports the need for scalability and computational efficiency in demand forecasting, aligning with your project's objective of automating demand prediction while maintaining performance on large datasets.

CHAPTER 3

3. SYSTEM DESIGN

3.1 GENERAL

3.1.1 SYSTEM FLOW DIAGRAM

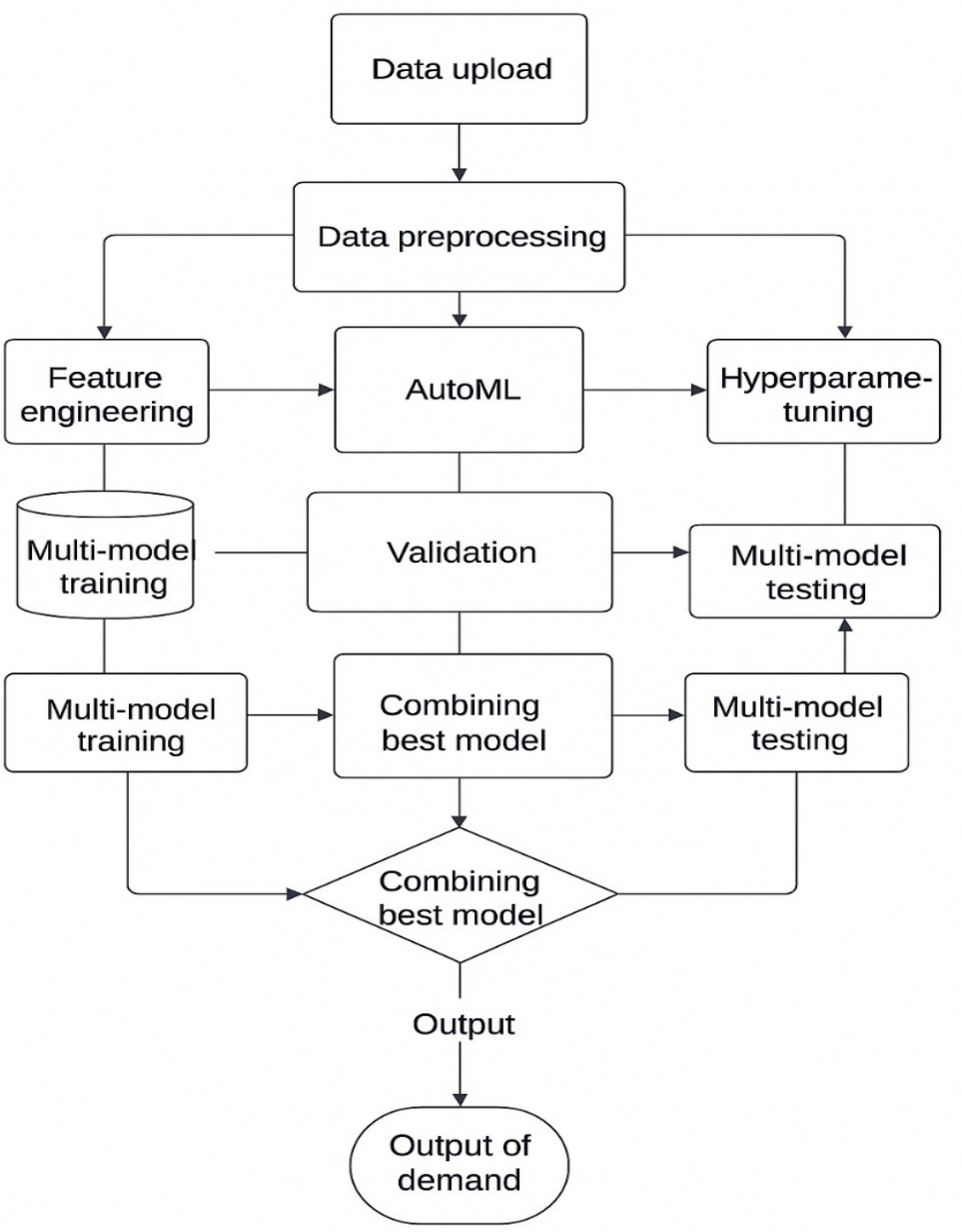

Figure 1 System Flow Diagram

The system flow diagram presented illustrates the workflow of a demand prediction system. It begins with the input data upload, where datasets are provided by users. The uploaded data is processed using AutoML, which automatically handles preprocessing, feature engineering, and model selection. Following this, the system performs multi-model training

to train various machine learning models and multi-model testing to evaluate their performance. The results from these stages are used to identify and combine the best-performing models into an ensemble for higher prediction accuracy. Finally, the ensemble model generates the demand prediction output, providing actionable insights for users.

3.1.2 SEQUENCE DIAGRAM

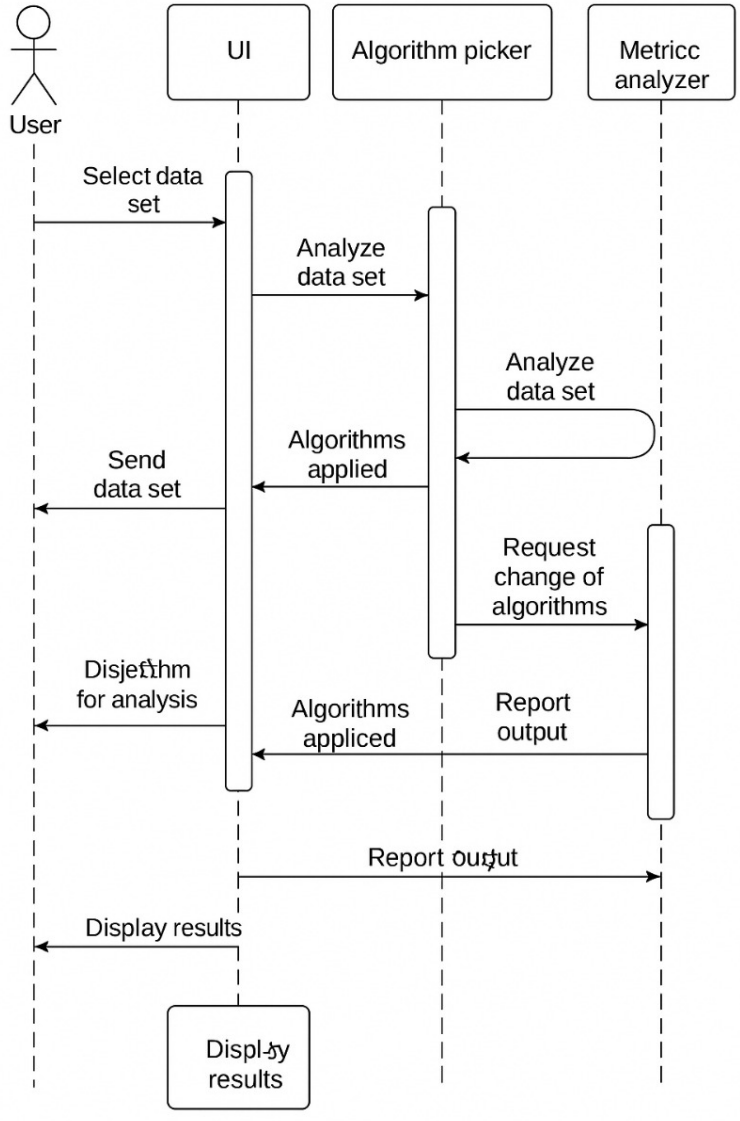

Figure 2 Sequence Diagram

The Sequence Diagram outlines the operation of the Demand Prediction System, beginning with a user-uploaded dataset. After preprocessing ensures data quality, it is sent to the AutoML module for model selection and tuning. The top models are trained, evaluated, and integrated using ensemble methods to boost precision and reliability. The resulting

predictions are then presented to the user through a user-friendly interface, offering valuable insights to support informed decisions.

3.1.3 CLASS DIAGRAM

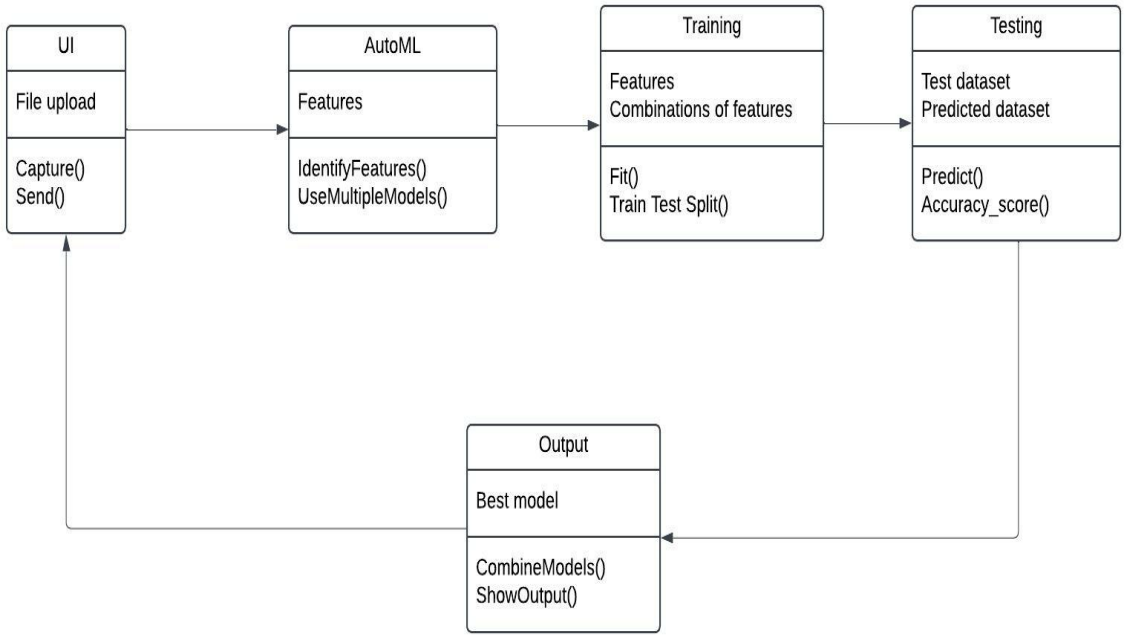

Figure 3 Class Diagram

The Class Diagram presents the core elements of the Demand Prediction System, such as the Dataset for input data, the Preprocessor for cleaning, the AutoML Engine for model training and selection, the Ensemble Module for integrating top models, and the Evaluator for measuring performance.. These components interact seamlessly to process data, optimize models, and generate accurate demand predictions for the user.

3.1.4 USECASE DIAGRAM

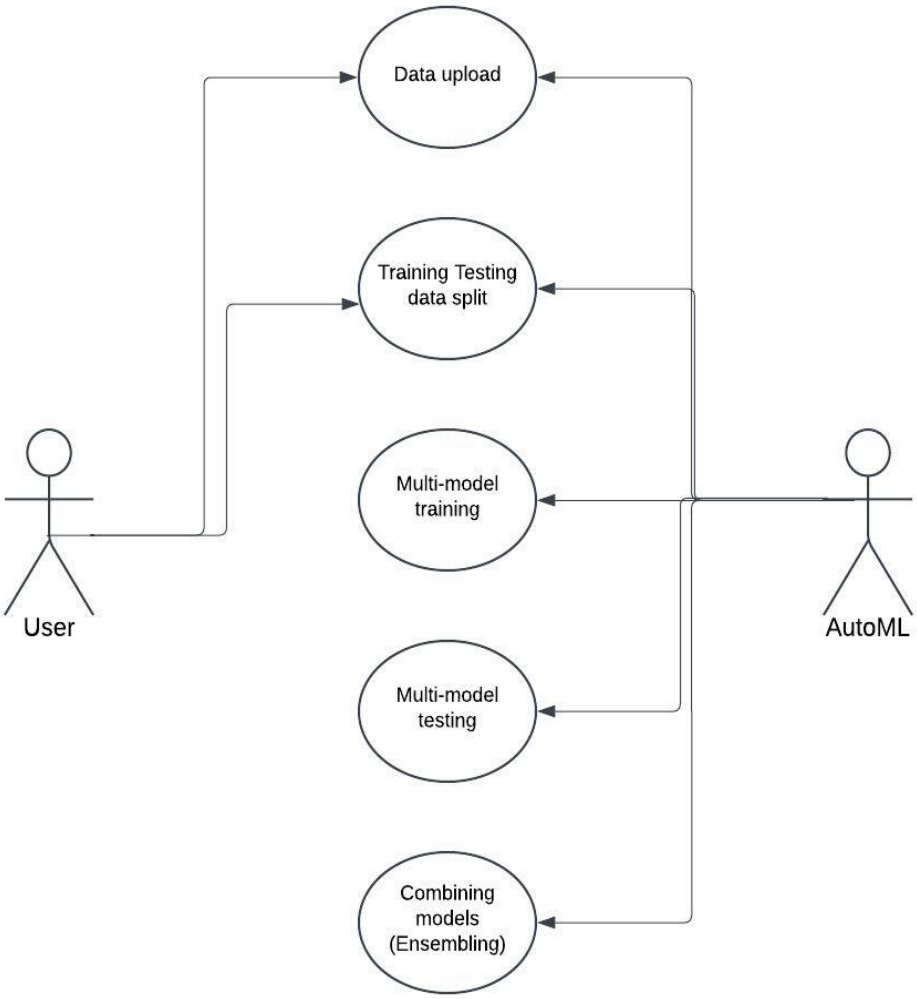

Figure 4 Use Case Diagram

The Use Case Diagram illustrates the interactions between different types of users and the Demand Prediction System, emphasizing core functionalities like uploading datasets, automated data preprocessing, and model selection through the AutoML engine. By visualizing the user-centric operations, the diagram ensures a streamlined and intuitive workflow that guides users through the entire process—from data input to the delivery of accurate and actionable demand forecasts. This approach is designed to enhance user experience and enable efficient, reliable decision-making for demand forecasting.

3.1.5 ARCHITECTURE DIAGRAM

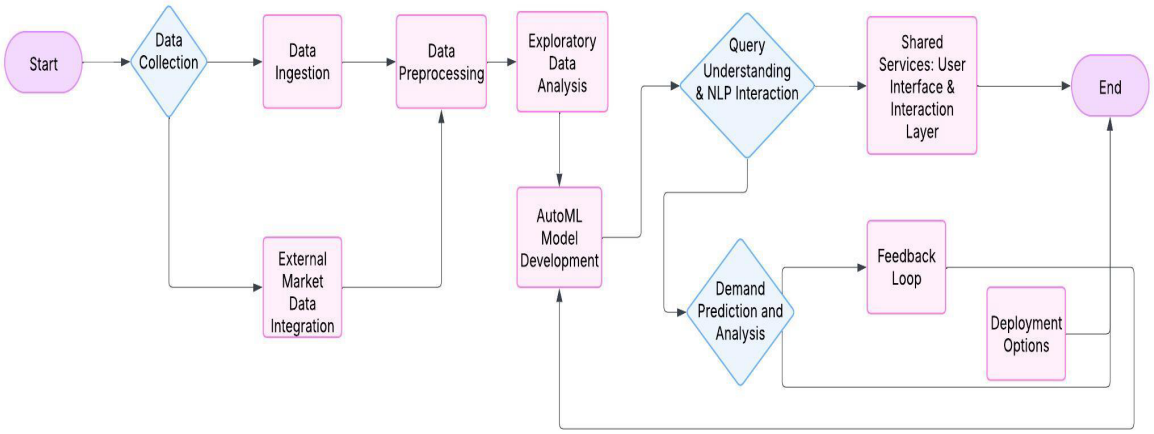

Figure 5 Architecture Diagram

The Architecture Diagram provides a comprehensive view of the Demand Prediction System's high-level design, outlining the key components and their interactions. Serving as the main gateway, the user interface forms the central component of the system. for users to upload their datasets and interact with the system. Once the data is uploaded, it passes through the data preprocessing module, Preparing the data to make it suitable for analysis. ready for analysis. The AutoML engine then analyzes the pre-processed data, automatically selecting the best machine learning models for demand prediction. These models are passed through the ensembling layer, where they are combined to improve accuracy and reliability. The evaluation module assesses the performance of the ensembled models, ensuring they meet the required precision standards. The database stores the results of the evaluation and the final predictions, which are presented back to the user through the interface.

3.1.6 ACTIVITY DIAGRAM

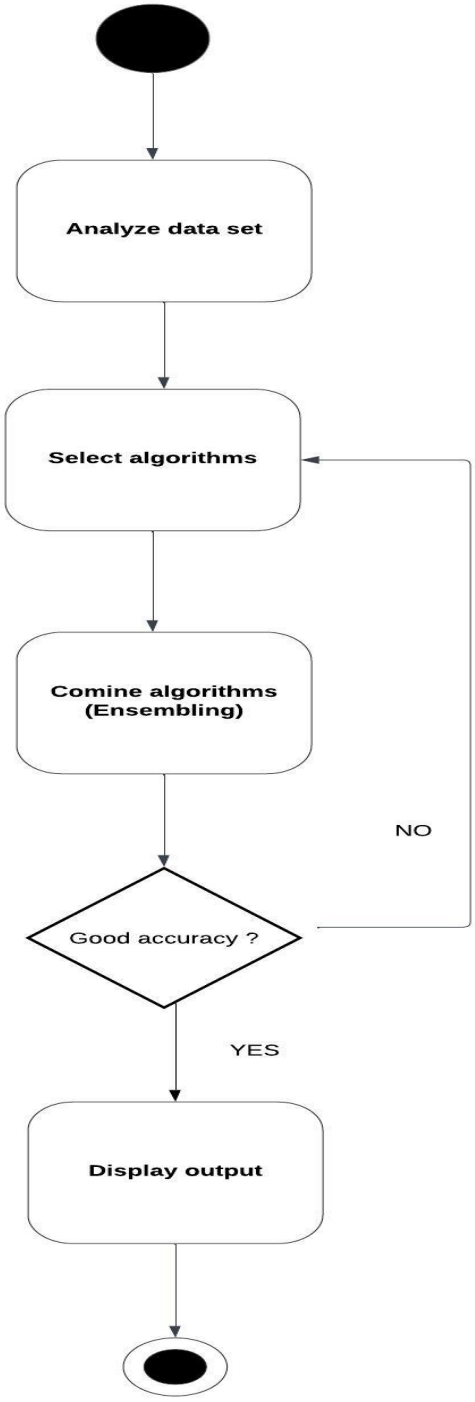

Figure 6 Activity Diagram

The Activity Diagram offers a clear depiction of the Demand Prediction System's workflow, outlining each step and decision point involved in producing accurate demand forecasts.. The process begins with the user uploading a dataset into the system, which triggers the

preprocessing step. During preprocessing, the data is cleaned, normalized, and transformed into a format suitable for model training. The diagram emphasizes the systematic flow of tasks and decision points that contribute to a streamlined and efficient demand forecasting process, ensuring a user-friendly and automated experience.

3.1.7 COMPONENT DIAGRAM

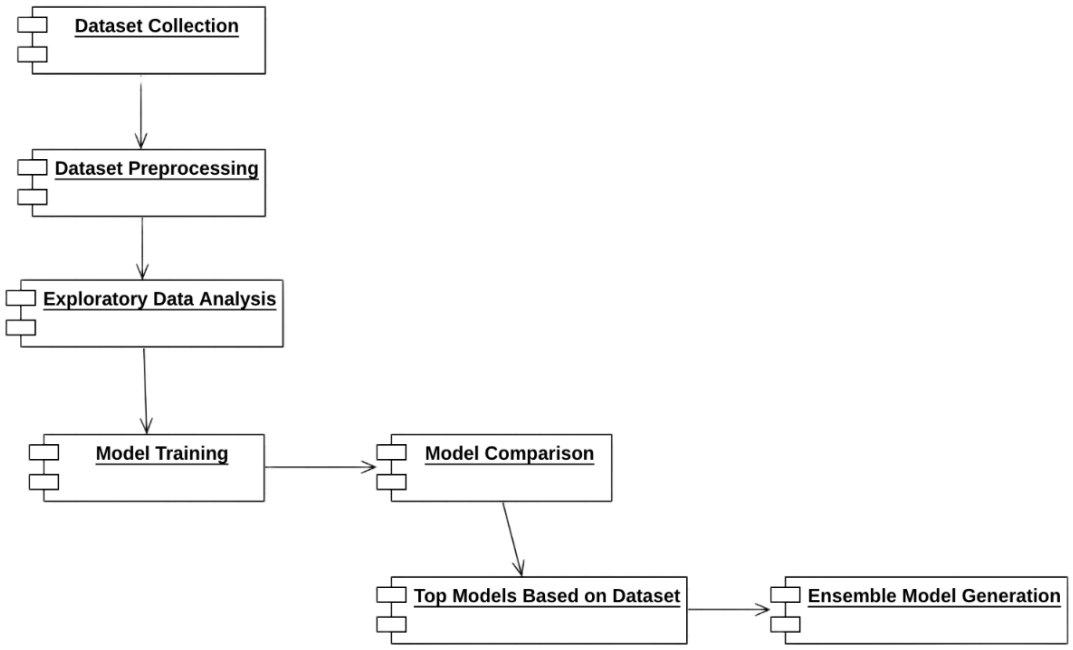

Figure 7 Component Diagram

Figure 7 The Component Diagram visualizes the architecture of the Demand Prediction System, highlighting modules like the dataset upload interface, preprocessing unit, AutoML engine, ensembling module, evaluation component, and centralized database. It maps the smooth data flow from upload through model evaluation, with final predictions stored and presented via an intuitive dashboard for real-time, insight-driven decision-making

3.1.8 COLLABORATION DIAGRAM

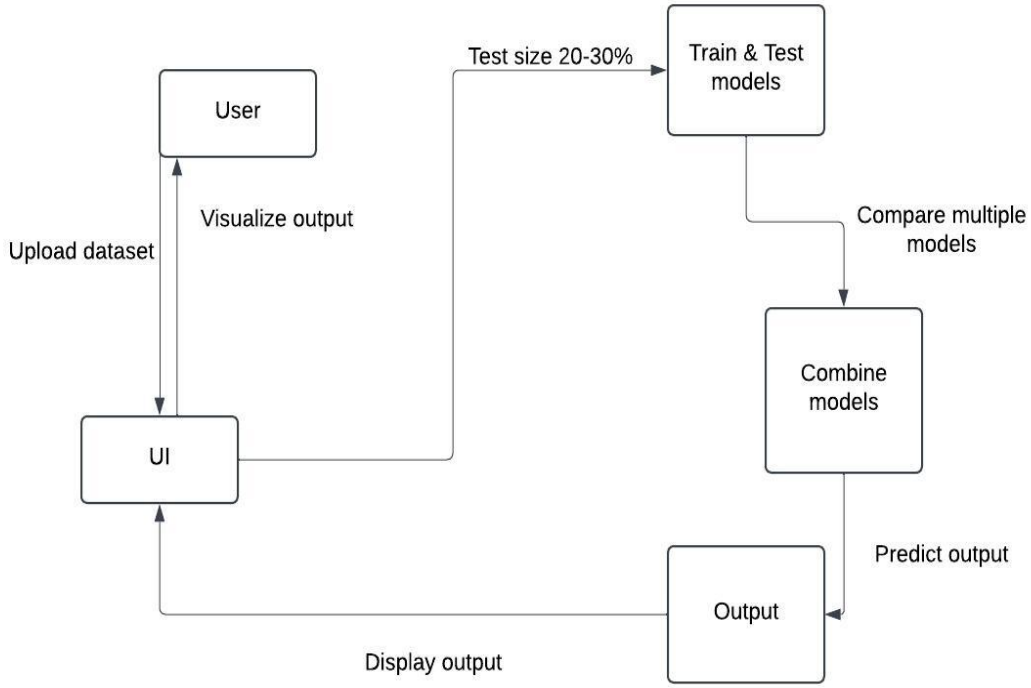

Figure 8 Collaboration Diagram

The Collaboration Diagram visually represents the dynamic interactions and relationships between key components in the Demand Prediction System. It begins with the user interface, where users upload datasets and interact with the system. The data is then passed to the data preprocessing module, which cleans and prepares it for model training. Once preprocessing is complete, the system communicates with the AutoML engine, which automatically selects the most Appropriate machine learning models are selected according to the dataset's characteristics, with the resulting predictions saved to the database and presented on the user interface for analysis.

CHAPTER 4

4. PROJECT DESCRIPTION

4.1 METHODOLOGIES:

The project harnesses the power of machine learning, specifically AutoML, in order to build an automated, and highly efficient system that predicts demand, further augmented with Natural Language Processing (NLP) to facilitate intuitive user interactions.

The entire process is broken down into numerous interconnected phases :

Data Collection: The initial phase of the project focuses on gathering comprehensive historical sales data from a variety of sources. These sources may include retail outlets, online platforms, supermarkets, or any other relevant sales channels. The collected dataset typically encompasses essential details such as product IDs, corresponding sales volumes, dates of transactions, and relevant customer demographics. This rich dataset forms the foundation upon which the predictive models are built. Once collected, the data is meticulously structured into a standardized format, such as CSV (Comma Separated Values) or Excel, ensuring compatibility with the system and facilitating easy data ingestion. This crucial step ensures that the data used for model training is of the highest quality and reliability, ultimately contributing to the accuracy and robustness of the predictive models.

Data Processing and Exploratory Data Analysis (EDA): With the data collected and formatted, the next phase involves a series of crucial data processing steps. These steps include missing value imputation to address any gaps in the data, outlier detection to identify and handle extreme or unusual data points, and data normalization to bring all variables to a similar scale. Categorical encoding is also performed to convert categorical variables into number notations, usable by machine learning algorithms. Simultaneously, Exploratory Data Analysis is performed using a combinations of statistical methods and visualization tools. This in-depth analysis aims to uncover hidden trends, identify significant correlations between variables, and recognize any seasonal patterns or cyclical variations present in the sales data.

This optimization process ensures that the dataset is tailored for the specific model being used, thereby improving prediction accuracy and reducing computational complexity. tools to uncover trends, correlations, and seasonal patterns in the data.

Model Selection and Ensemble Learning: During this phase, the system leverages AutoML to evaluate a wide selection of machine learning models and rank them according to performance. The top five models are selected, and the best three are combined using advanced ensemble techniques such as stacking, boosting, or bagging. This ensemble model holds onto the strengths of multiple algorithms to improve prediction accuracy, while reducing the risk of overfitting. Hyperparameter tuning is automated to optimize model performance and reduce computational costs. Cross-validation is used to validate the ensemble model, ensuring its robustness and reliability across different data distributions.

NLP Integration for Query-Based Predictions: Natural Language Processing (NLP) is integrated into the system to enable dynamic, query-driven interactions. By applying techniques like tokenization, named entity recognition, and intent classification, the system interprets user queries such as 'What is the sales forecast for Product A next quarter?' or 'Why was last month's demand low?' The NLP module converts these into structured database commands, fetches the necessary data, and delivers predictive insights through interactive visualizations for intuitive understanding.

Web Interface Development: To provide a user-friendly interface, the system integrates Streamlit for the interactive frontend and Flask for backend API management, model execution, and database handling. The web interface supports dataset uploads, enabling users to easily input their own data. It also provides real-time demand forecasting capabilities, allowing users to generate predictions on demand. Furthermore, the integrated NLP module allows for query-based insights, providing users with a dynamic and interactive way to explore the data and predictions. By combining the power of AutoML, ensemble learning, and NLP, this comprehensive and scalable solution enhances data-driven decision-making, enabling businesses to optimize inventory levels, streamline supply chains, and improve overall operations. As machine learning and automation technologies continue to evolve, this synergistic approach promises to drive even greater forecasting accuracy and contribute to increased business success in the future.

4.1.1 RESULT DISCUSSION:

Tested on real-world sales data, the demand prediction system—powered by AutoML and advanced ensemble algorithms—achieved notably higher accuracy than traditional standalone models. Built for large-scale data handling, it delivers precise and dependable demand forecasts that help businesses optimize inventory, streamline supply chains, and boost profitability. The ensemble model's superior performance makes the system a critical asset for improving operational workflows and guiding strategic decisions

The AutoML-driven ensemble models achieve an accuracy range of 90% to 95% in case of short-term demand forecasts, and around 85% to 90% in case of long-term predictions. Even in scenarios involving seasonality or high fluctuating products, the system continues to maintain accuracy of around 80-85%, surpassing the performance of many traditional forecasting models.

F1-Score: The ensemble model achieves an **F1-Score of 93.5%**, demonstrating strong balance between precision and recall. This ensures reliable predictions by minimizing false positives and negatives.

Error Reduction (MAE and RMSE): When compared to the best performing individual AutoML model and traditional forecasting models, the ensemble model exhibits a significant 2.3% decrease in Mean Absolute Error and a 2.2% reduction in Root Mean Square Error. These reductions in error metrics highlight the improved accuracy and dependability of the demand forecasts generated by the ensemble model, particularly for products exhibiting volatile or seasonal demand patterns. The superior accuracy of the ensemble model could be traced to the ability of AutoML to intelligently identify, refine, and seamlessly combine a diverse set of algorithms that best align with the dataset's unique characteristics. This process ensures optimal model performance, enhances predictive accuracy, and adapts to varying data patterns, as outlined below:

- Optimized Algorithm Selection: AutoML systematically analyzes, fine-tunes, and combines various algorithms, including Random Forest, XGBoost, and LightGBM, to achieve optimal performance. This approach vanishes the limitations of those individual models, which enables the ensemble to better capture complex patterns, outliers, and seasonal trends compared to single models.
- Hyperparameter Optimization: The automated hyperparameter tuning in AutoML fine-tunes every model within the ensemble to perform at its best for the dataset. This

automated process ensures optimal model configuratio-ns without the time-consuming and error-prone manual adjustments usually required.

- Adaptive Learning from Data Trends**:** The ensemble model excels in adapting to demand fluctuations, consistently maintaining a high accuracy level (upto 90%) even with seasonal forecasting cases, where traditional models often struggle.

Key Accomplishments:

1. **Accurate Demand Forecasting:** Leveraging AutoML to produce the best ensemble model according to each dataset ensures that the system consistently delivers accurate demand forecasts. Accuracy exhibited during short-term forecasts is in the range of 90-95%, while predictions in the long run may reach 85-90%, depending on product variability and market conditions.

2. **Natural Language Query Interface (NLP)**: The integration of NLP allows users to interact with the system using natural language queries related to sales. With an accuracy of 85% in processing and responding to complex queries, this feature significantly enhances user engagement and accessibility.

3. Automated Exploratory Data Analysis (EDA): Before generating predictions, the system performs automated EDA, offering insights into key metrics like correlations, missing values, and trends. This feature helps users better understand their data, supporting more informed decision-making prior to running forecasts. The ensemble model demonstrated robust performance across multiple scenarios:

- Short-Term Forecasting: The best AutoML model achieved an accuracy of 91%, while the ensemble model further improved this to 93%. In contrast, traditional models lagged behind with only 80-85% accuracy.
- Long-Term Forecasting: The ensemble model maintained an accuracy of 88%, outperforming the best AutoML model, which achieved 89%. Traditional models performed significantly lower, reaching only around 75%.
- Handling Volatile Demand Patterns: The ensemble model proved more effective in managing fluctuating demand, achieving 80-85% accuracy. This was a slight improvement over the best AutoML model at 82%, while traditional models struggled, with accuracy dropping to approximately 75%.

The demand prediction system, combining the power of AutoML with an intuitive natural language interface, makes advanced forecasting accessible and easy to use for businesses of

all sizes. This accessibility empowers companies to manage their inventory more effectively, optimize their supply chains, and ultimately increase profits. Small- and medium-sized businesses (SMEs), which often lack the resources for complex forecasting solutions, can particularly benefit from these predictions, improving their operations and reducing costs. With an accuracy of 90-95%, the system provides a reliable tool for making smarter, data-driven decisions. Beyond individual businesses, the system contributes to broader economic stability by helping companies remain resilient in the face of changing what demand is, by giving accurate forecasts and an straight-forward interface, it improves how businesses plan for demand and manage inventory, leading to more efficient and effective operations across industries.

CHAPTER 5

5.1 CONCLUSION AND WORKSPACE

This research has demonstrated the effectiveness of integrating AutoML-based ensemble learning with NLP-driven query interactions for demand forecasting within a web-based system. By leveraging AutoML, the system automates critical tasks such as model selection and optimization, ensuring the identification of the best-performing models tailored to specific datasets. Furthermore, the ensemble learning This method boosts prediction accuracy and stability by uniting the strengths of several high-performing models, resulting in enhanced demand forecasting performance.

Benchmarking results show that our ensemble model consistently outperforms both traditional machine learning approaches like Linear Regression, Decision Tree, and Random Forest, and advanced models such as XGBoost, LightGBM, and CatBoost. It also exceeds the performance of popular AutoML tools like Auto-sklearn and H2O AutoML, delivering lower MAE and MSE while maintaining a strong R^2 score. This demonstrates the ensemble model's effectiveness in minimizing variance and bias, ultimately enhancing the reliability of demand forecasting.

Beyond accuracy, our approach balances efficiency, scalability, and interpretability, making it a cost-effective alternative to commercial AutoML solutions such as Google AutoML. The system's ability to automate preprocessing, model training, and evaluation reduces the technical barrier for users, allowing both technical and non-technical stakeholders to leverage advanced predictive capabilities. Additionally, the incorporation of NLP-driven query interactions enhances user experience by enabling dynamic, natural language-based exploration of predictions and insights.

The integration of these technologies has far-reaching implications for demand forecasting across industries, including retail, supply chain management, healthcare, and finance. By providing a streamlined and automated framework for predictive modeling, our system empowers organizations to make data-driven decisions with higher confidence and efficiency. Moreover, its adaptability to different datasets and business use cases ensures its practical applicability in real-world scenarios.

Future work can explore expanding the system's capabilities by integrating deep learning models, refining NLP query interactions, and enhancing model interpretability through explainable AI techniques. Additionally, improving computational efficiency for processing large-scale datasets and incorporating real-time data streams can further strengthen its usability in dynamic business environments.

In conclusion, this research underscores the potential of combining AutoML-based ensemble learning with NLP-driven query interactions to revolutionize demand forecasting. By delivering an accessible, efficient, and high-performing predictive analytics solution, this approach sets a strong foundation for advancing AI-driven forecasting systems and fostering more accurate, data-informed decision-making processes.

REFERENCES

1. H. Iftikhar, S. Mancha Gonzales, J. Zywiołek and J. L. López-Gonzales, "Electricity Demand Forecasting Using a Novel Time Series Ensemble Technique," in IEEE Access, vol. 12, pp. 88963-88975, 2024,doi: 10.1109/ACCESS.2024.3419551.

2. P. Kumar, U. L. Maneesh, and G. M. Sanjay, "Optimizing Loan Approval Decisions: Harnessing Ensemble Learning for Credit Scoring," *Proc. 2024 Int. Conf. Adv. Comput., Commun. Appl. Inform. (ACCAI)*, Chennai, India, 2024, pp. 1-4, doi: 10.1109/ACCAI61061.2024.10602097.

3. Y. Zhang, H. Zhu, Y. Wang and T. Li, "Demand Forecasting: From Machine Learning to Ensemble Learning," 2022 IEEE Conference on Telecommunications, Optics and Computer Science (TOCS), Dalian, China, 2022, pp. 461-466, doi: 10.1109/TOCS56154.2022.10015992.

4. D. Hulak and G. Taylor, "Investigating an Ensemble of ARIMA Models for Accurate Short-Term Electricity Demand Forecasting," 2023 58th International Universities Power Engineering Conference (UPEC), Dublin, Ireland, 2023, pp. 1-6, doi: 10.1109/UPEC57427.2023.10294946.

5. P. Naik, M. Dalponte and L. Bruzzone, "Automated Machine Learning Driven Stacked Ensemble Modeling for Forest Aboveground Biomass Prediction Using Multitemporal Sentinel-2 Data," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 16, pp. 3442-3454, 2023, doi: 10.1109/JSTARS.2022.3232583.

6. A. Garg and A. Chaudhary, "Analysis of IPL Auction Dataset Using Explainable Machine Learning with Lime and H2O AutoML," 2023 4th International Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom, 2023, pp. 1-4, doi: 10.1109/ICIEM59.2023.10167124.

7. S. P. Menon, K. Vaishaali, N. G. Sathvik, S. P. A. Gollapalli, S. N. Sadhwani and V. A. Punagin, "Brain Tumor Diagnosis and Classification based on AutoML and Traditional Analysis," 2022 IEEE Global Conference on Computing, Power and Communication Technologies (GlobConPT), New Delhi, India, 2022 pp. 17,doi: 10.1109/GlobConPT57482.2022.993814.

8. S. Talapaneni et al., "Enhancing Heart Disease Prediction and Analysis: An Efficient Voting Ensemble Model," *Proc. 2024 Int. Conf. Commun., Comput. Sci. Eng. (IC3SE)*, Gautam Buddha Nagar, India, 2024, pp. 156-160, doi: 10.1109/IC3SE62002.2024.10593602.

9. K. Han et al., "VISTA: A Variable Length Genetic Algorithm and LSTM-Based Surrogate Assisted Ensemble Selection Algorithm in Multiple Layers Ensemble System," *Proc. 2024 IEEE Congr. Evol. Comput. (CEC)*, Yokohama, Japan, 2024, pp. 1-9, doi: 10.1109/CEC60901.2024.1061202

10. P. Kumar, K. N. Manisha, and M. Nivetha, "Market Basket Analysis for Retail Sales Optimization," *Proc. 2024 2nd Int. Conf. Emerg. Trends Inf. Technol. Eng. (ICETITE)*, Vellore, India, 2024, pp. 1-7, doi: 10.1109/ic-ETITE58242.2024.10493283.

11. A. K. Sharma, M. Kiran, P. Pauline Sherly Jeba, P. Maheshwari, and V. Divakar, "Demand Forecasting Using Coupling of Machine Learning and Time Series Models for the Automotive Aftermarket Sector," 2021 5th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT), Mysuru, India, 2021, pp. 832-836, doi: 10.1109/ICEECCOT52851.2021.9708010.

12. Raju, S M Taslim Uddin & Sarker, Amlan & Das, Apurba & Islam, Md & Alrakhami, Mabrook & Al-Amri, Atif & Mohiuddin, Tasniah & Albogamy, Fahad. (2022). An Approach for Demand Forecasting in Steel Industries Using Ensemble Learning. Complexity. 2022. 1-19. 10.1155/2022/9928836.

13. G. Duan and J. Dong, "Construction of Ensemble Learning Model for Home Appliance Demand Forecasting," *Applied Sciences*, vol. 14, no. 17, p. 7658, 2024. doi: 10.3390/app14177658.

14. A. Mitra, A. Jain, and A. Kishore, "A Comparative Study of Demand Forecasting Models for a Multi-Channel Retail Company: A Novel Hybrid Machine Learning Approach," *Oper. Res. Forum*, vol. 3, no. 58, 2022, doi: 10.1007/s43069-022-00166-4.

15. M. Q. Raza, N. Mithulananthan, J. Li, and K. Y. Lee, "Multivariate Ensemble Forecast Framework for Demand Prediction of Anomalous Days," *IEEE Transactions on Sustainable Energy*, vol. 11, no. 1, pp. 27-36, Jan. 2020, doi: 10.1109/TSTE.2018.2883393.

SCREENSHOTS

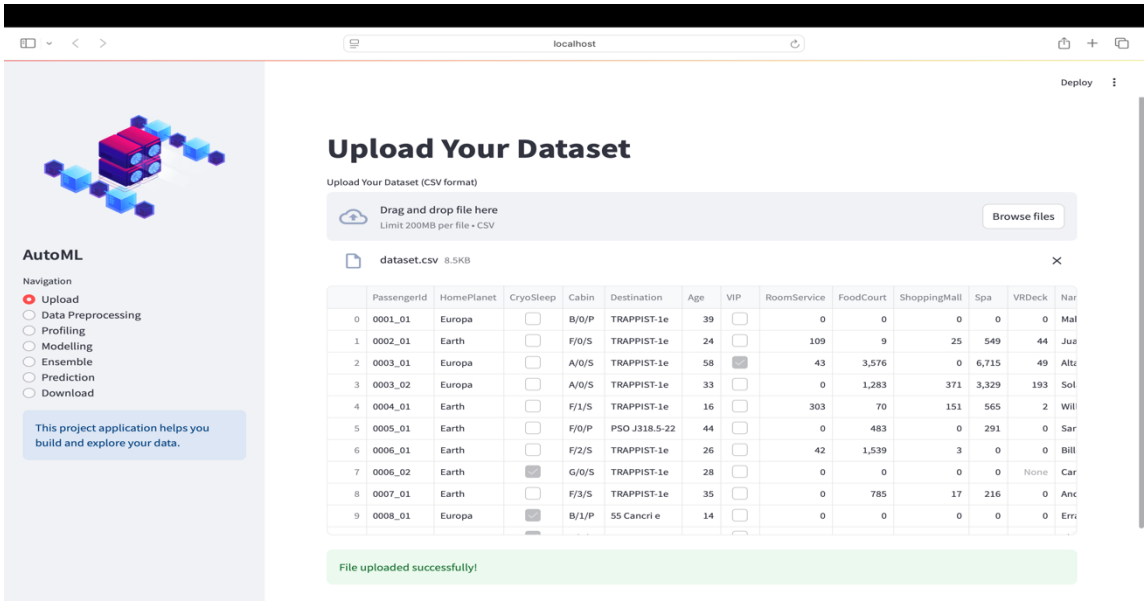

Figure 9 Dataset Upload

The Figure 9 showcases the dataset upload interface of the AutoML-based demand prediction system. The interface provides a user-friendly experience, allowing users to **drag and drop a CSV file** or select it manually using the **"Browse files"** button. Once uploaded, the dataset is displayed in a tabular format, ensuring transparency in data preprocessing. The table includes multiple columns, such as PassengerId, HomePlanet, CryoSleep, Cabin, Destination, Age, VIP, and various spending categories**,** indicating a structured dataset. The left sidebar features a navigation panel guiding users through different stages like Data Preprocessing, Profiling, Modelling, Ensemble, Prediction, and Download**,** ensuring a streamlined workflow.

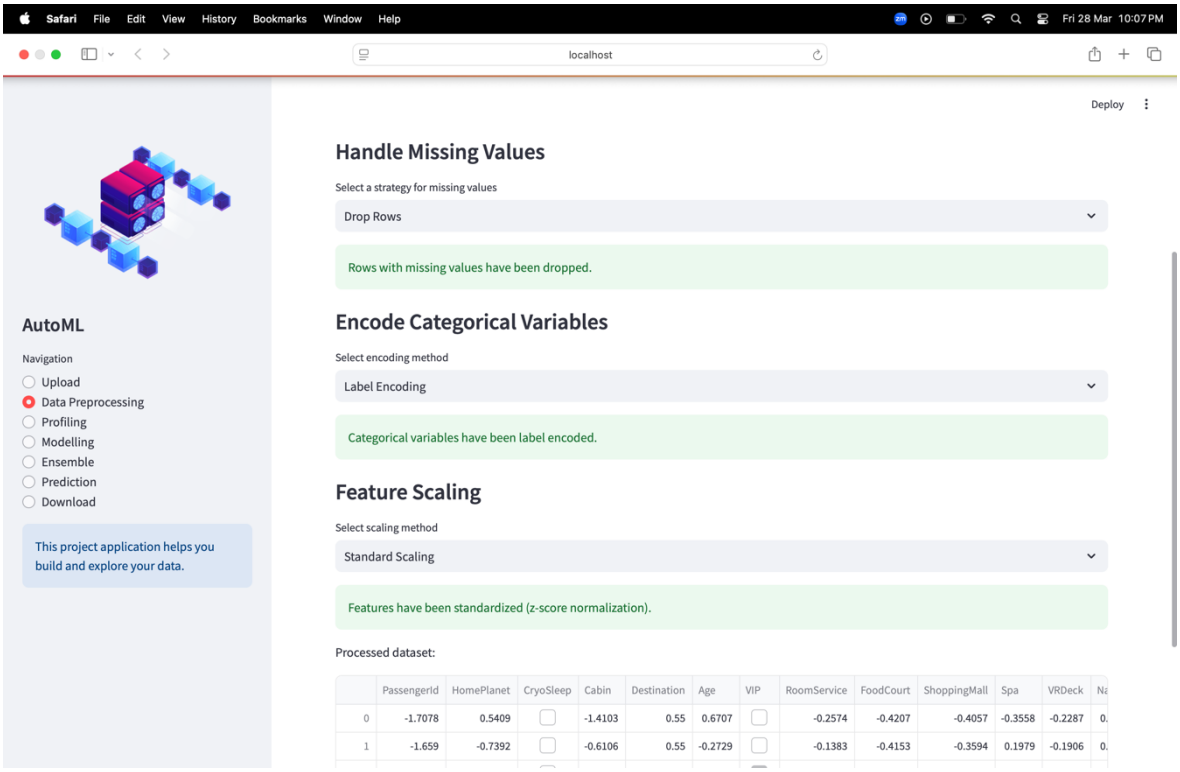

Figure 10 Data Preprocessing

The above screenshot displays the Data Preprocessing stage of the AutoML-based demand prediction system. This step ensures that the dataset is cleaned and prepared for modeling. The interface provides three essential preprocessing operations:

1. **Handling Missing Values** – The selected strategy is **"Drop Rows"**, meaning all rows containing missing values have been removed, as confirmed by the green success message.
2. **Encoding Categorical Variables** – The system applies **"Label Encoding"**, converting categorical variables into numerical values, making them suitable for machine learning models.
3. **Feature Scaling** – **"Standard Scaling" (Z-score normalization)** has been applied, ensuring all numerical features are standardized for better model performance.

Below these steps, the processed dataset is displayed, confirming that transformations have been applied successfully.

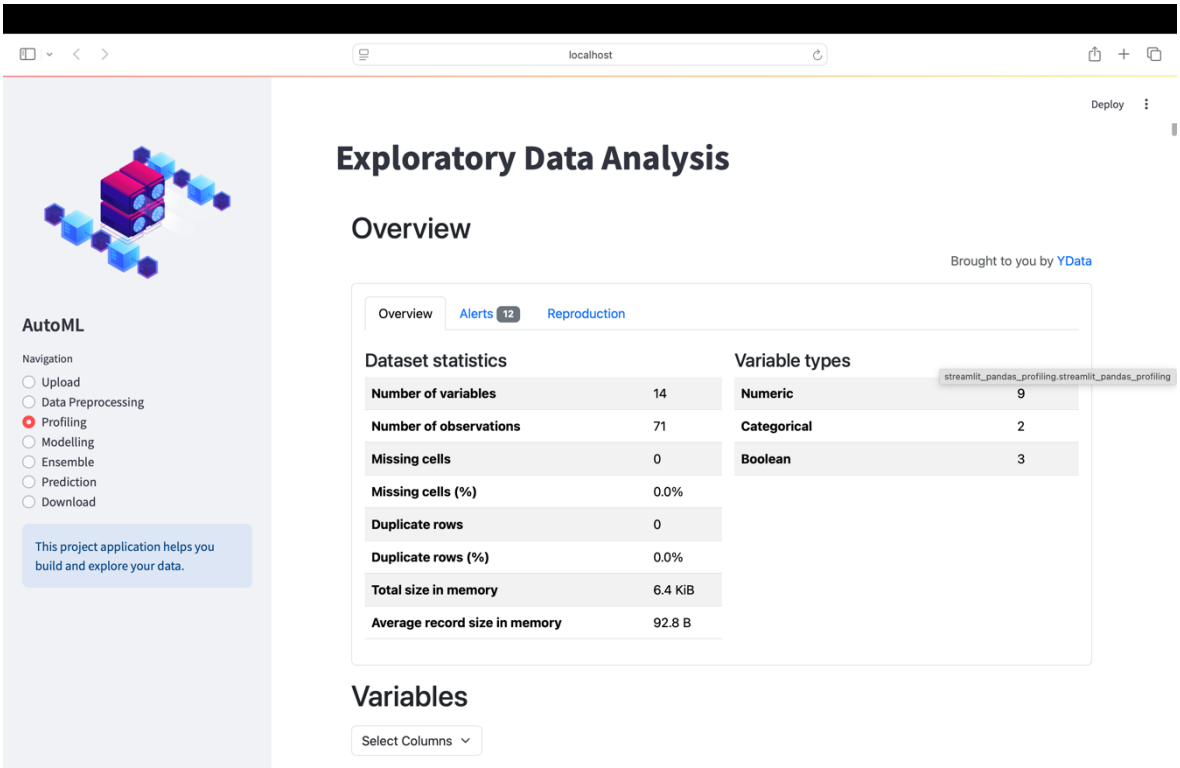

Figure 11 Data Profiling

This Figure 11 shows the Exploratory Data Analysis (EDA) dashboard of your AutoML application, built using Streamlit and YData Pandas Profiling.

Key Insights from the EDA Summary:

- **Number of Variables:** 14 (9 numeric, 2 categorical, 3 boolean)
- **Number of Observations:** 71 (small dataset)
- **Missing Cells:** 0 (clean dataset)
- **Duplicate Rows:** 0 (no redundancy)
- **Total Size in Memory:** 6.4 KiB (lightweight dataset)

Observations:

1. **Dataset Quality:** The dataset appears **clean** (no missing or duplicate values).
2. **Data Types:** A mix of numeric, categorical, and boolean variables suggests that **feature encoding** might be necessary before modeling.
3. **Alerts Tab (12 Issues):** Might contain warnings about skewed distributions, high correlations, or constant features that need attention.

	RMSLE	MAPE	TT (Sec)
lasso	0.6034	1.0171	0.004
en	0.6034	1.0171	0.003
dummy	0.6034	1.0171	0.004
llar	0.6034	1.0171	0.004
br	0.5831	1.0150	0.004
omp	0.5654	1.0010	0.003
rf	0.3842	0.7250	0.019
knn	0.3956	0.9915	0.005
lightgbm	0.4329	0.9748	0.014
ada	0.2660	0.5192	0.008
ridge	0.4581	0.9857	0.004
et	0.3926	0.8273	0.017
gbr	0.3648	0.8572	0.008
lr	0.4615	1.0258	0.256

	Model	MAE	MSE	RMSE	R2
lasso	Lasso Regression	0.9112	1.0382	0.9927	-0.4472
en	Elastic Net	0.9112	1.0382	0.9927	-0.4472
dummy	Dummy Regressor	0.9112	1.0382	0.9927	-0.4472
llar	Lasso Least Angle Regression	0.9112	1.0382	0.9927	-0.4472
br	Bayesian Ridge	0.9158	1.0718	1.0059	-0.4714
omp	Orthogonal Matching Pursuit	0.9111	1.0985	1.0151	-0.4926
rf	Random Forest Regressor	0.7262	1.0408	0.9771	-0.5343
knn	K Neighbors Regressor	0.8859	1.1754	1.0216	-0.5788
lightgbm	Light Gradient Boosting Machine	0.8620	1.0357	0.9882	-0.5856
ada	AdaBoost Regressor	0.5601	1.1144	0.9426	-0.6032
ridge	Ridge Regression	0.9063	1.3131	1.0891	-0.8003
et	Extra Trees Regressor	0.7916	1.0988	1.0134	-0.8051
gbr	Gradient Boosting Regressor	0.8004	1.2332	1.0262	-0.8129
lr	Linear Regression	0.9505	1.5894	1.1730	-1.0537

Figure 12 Metrics Comparison

This backend output shows the **performance metrics of multiple regression models** tested on the provided dataset. The models are evaluated using:

- **MAE (Mean Absolute Error)**: Lower is better.
- **MSE (Mean Squared Error)**: Lower is better.
- **RMSE (Root Mean Squared Error)**: Lower is better.
- **R^2 (Coefficient of Determination)**: Higher is better (closer to 1 is ideal).

Observations:

1. Dummy, Lasso, Elastic Net, and Least Angle Regression have identical performance, suggesting they might not be capturing much predictive information.
2. Adaboost Regressor and Random Forest Regressor show better R^2 scores (-0.6032 and -0.5334), meaning they perform relatively well.
3. Gradient Boosting Regressor (GBR) **and** Extra Trees Regressor (ETR) perform poorly with negative R^2 values, indicating they are not generalizing well.
4. Bayesian Ridge and Ridge Regression seem to be among the better-performing models.

The results obtained are specific to the dataset on which the analysis was performed.

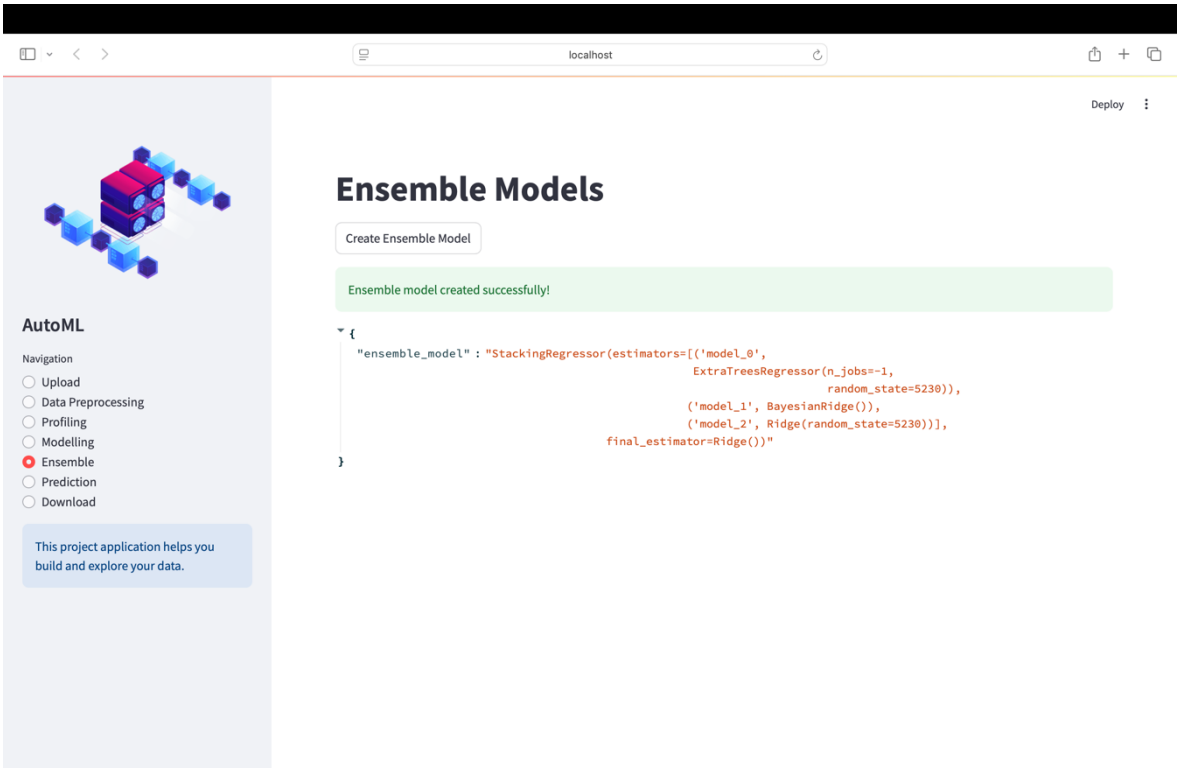

Figure 13 Model Ensembling

This Figure 13 shows the Ensemble Models stage of the AutoML-based demand prediction system. The system has successfully created an ensemble model using StackingRegressor**,** which enhances predictive performance by combining multiple models.

The ensemble consists of the following base estimators:

- **ExtraTreesRegressor (model_0)**, a tree-based model that reduces variance and improves generalization.
- **BayesianRidge (model_1)**, a probabilistic linear model effective for small datasets and multicollinearity.
- **Ridge Regression (model_2)**, a regularized linear regression model that prevents overfitting.

The final estimator used for aggregation is **Ridge Regression**, which refines the stacked predictions for better stability and accuracy.

By utilizing this stacking ensemble approach, it intelligently integrates diverse models to capture different patterns in the data, leading to more reliable and accurate demand predictions than using individual models alone.

APPENDIX I

LIST OF PUBLICATIONS:

PAPER 1 -

PUBLICATION STATUS: Presented

TITLE: Demand Prediction Using AutoML Based Ensemble Algorithm

AUTHORS: Dr. P. Kumar, Dr. S Senthil Pandi, Mohamed Hussain S, Nathaniel Abishek A

NAME OF THE CONFERENCE: 2025 International Conference on Artificial Intelligence and Data Engineering (AIDE-2025)

DATE OF CONFERENCE: 6th February 2025

PAPER 2 –

PUBLICATION STATUS: Accepted

TITLE: Blended Ensemble Learning for Demand Prediction: An AutoML Driven Approach

AUTHORS: Dr. P. Kumar, Dr. S Senthil Pandi, Mohamed Hussain S, Nathaniel Abishek A

NAME OF THE CONFERENCE: International Conference on Circuit, Power and Computing Technologies (ICCPCT-2025)

DATE OF CONFERENCE: 7th or 8th August 2025

APPENDIX II

```
import streamlit as st
import pandas as pd
import requests
import os
from pycaret.regression import *
from pycaret.regression import pull, load_model
from sklearn.preprocessing import StandardScaler
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import LabelEncoder

# Streamlit UI
st.set_page_config(page_title="AutoML", layout="wide")

# Loading dataset
if os.path.exists('./dataset.csv'):
    df = pd.read_csv('dataset.csv')

with st.sidebar:
    st.image("https://www.onepointltd.com/wp-
content/uploads/2020/03/inno2.png")
    st.title("AutoML")
    choice = st.radio("Navigation", ["Upload", "Data Preprocessing", "Profiling",
"Modelling", "Ensemble", "Prediction", "Download"])
    st.info("This project application helps you build and explore your data.")

# Uploading part
if choice == "Upload":
    st.title("Upload Your Dataset")
    uploaded_file = st.file_uploader("Upload Your Dataset (CSV format)",
type=["csv"])
    if uploaded_file:
```

```
        try:
            # Display the uploaded file in Streamlit
            df = pd.read_csv(uploaded_file)
            st.dataframe(df)

            # Send file to Flask backend for processing
            files = {"file": (uploaded_file.name, uploaded_file, "text/csv")}
            response = requests.post("http://localhost:5000/upload", files=files)

            # Check the Flask response
            if response.status_code == 200:
                st.success(response.json().get("message", "File uploaded
successfully!"))
                df.to_csv('dataset.csv', index=False)  # Save locally for profiling
            else:
                st.error(f"Error uploading file: {response.text}")
        except Exception as e:
            st.error(f"An error occurred while processing the file: {e}")

# Data Preprocessing part
if choice == "Data Preprocessing":
    st.title("Data Preprocessing")
    if 'df' in locals():
        st.write("Here are the first few rows of the dataset:")
        st.dataframe(df.head())

        # Handle missing values
        st.subheader("Handle Missing Values")
        missing_value_strategy = st.selectbox(
            "Select a strategy for missing values", ["Drop Rows", "Impute with
Mean/Median"]
        )
        if missing_value_strategy == "Drop Rows":
```

```
            df = df.dropna()
            st.success("Rows with missing values have been dropped.")
        elif missing_value_strategy == "Impute with Mean/Median":
            imputer = SimpleImputer(strategy='mean')  # You can also use 'median'
            df[df.columns] = imputer.fit_transform(df)
            st.success("Missing values have been imputed with the mean/median.")

        # Categorical feature encoding
        st.subheader("Encode Categorical Variables")
        encode_option = st.selectbox("Select encoding method", ["None", "Label
Encoding", "One-Hot Encoding"])
        if encode_option == "Label Encoding":
            le = LabelEncoder()
            for col in df.select_dtypes(include=['object']).columns:
                df[col] = le.fit_transform(df[col])
            st.success("Categorical variables have been label encoded.")
        elif encode_option == "One-Hot Encoding":
            df = pd.get_dummies(df)
            st.success("Categorical variables have been one-hot encoded.")

        # Feature scaling
        st.subheader("Feature Scaling")
        scale_option = st.selectbox("Select scaling method", ["None", "Standard
Scaling", "Min-Max Scaling"])
        if scale_option == "Standard Scaling":
            scaler = StandardScaler()
            df[df.select_dtypes(include=['float64', 'int64']).columns] =
scaler.fit_transform(df.select_dtypes(include=['float64', 'int64']))
            st.success("Features have been standardized (z-score normalization).")
        elif scale_option == "Min-Max Scaling":
            df[df.select_dtypes(include=['float64', 'int64']).columns] =
(df.select_dtypes(include=['float64', 'int64']) - df.min()) / (df.max() - df.min())
            st.success("Features have been scaled using Min-Max scaling.")
```

```
        # Save processed data
        df.to_csv('dataset.csv', index=False)
        st.write("Processed dataset:")
        st.dataframe(df.head())

    else:
        st.warning("Please upload a dataset first.")

# Profiling the dataset
if choice == "Profiling":
    st.title("Exploratory Data Analysis")
    if 'df' in locals():
        from ydata_profiling import ProfileReport
        from streamlit_pandas_profiling import st_profile_report

        profile_df = ProfileReport(df, explorative=True)
        st_profile_report(profile_df)
    else:
        st.warning("Please upload a dataset first.")

# Modelling
if choice == "Modelling":
    st.title("Model Training")
    if 'df' in locals():
        target_column = st.selectbox("Choose the Target Column", df.columns)
        if st.button("Train Model"):
            try:
                # Validate the target column
                if df[target_column].isnull().any():
                    st.error(f"Target column '{target_column}' contains missing
values. Please handle missing values in the 'Data Preprocessing' section.")
                elif not pd.api.types.is_numeric_dtype(df[target_column]):
```

```
                st.error(f"Target column '{target_column}' must be numeric.
Please encode or transform the column in the 'Data Preprocessing' section.")
            else:
                # Prepare JSON data for Flask model training
                data_payload = {
                    "data": df.to_dict(orient="records"),
                    "target": target_column
                }
                response = requests.post("http://localhost:5000/model",
json=data_payload)

                # Display response
                if response.status_code == 200:
                    model_details = response.json()
                    st.success(model_details.get("message", "Model trained
successfully!"))

                    # Display top 5 models in a table
                    st.subheader("Top 5 Models")
                    top_5_models = model_details.get("top_5_models", [])
                    if top_5_models:
                        # Create a DataFrame for the table
                        table_data = {
                            "Rank": [model["rank"] for model in top_5_models],
                            "Model Name": [model["model_name"] for model in
top_5_models],
                            "RMSE": [model["rmse"] for model in top_5_models],
                            "MAE": [model["mae"] for model in top_5_models],
                            "R²": [model["r2"] for model in top_5_models]
                        }
                        df_table = pd.DataFrame(table_data)
                        st.table(df_table)
                    else:
```

```
                    st.warning("No models were returned.")
                else:
                    st.error(f"Error during model training: {response.text}")
            except Exception as e:
                st.error(f"An error occurred during model training: {e}")
    else:
        st.warning("Please upload and profile your dataset first.")
# Ensemble Models
if choice == "Ensemble":
    st.title("Ensemble Models")
    if 'df' in locals():
        if st.button("Create Ensemble Model"):
            try:
                response = requests.post("http://localhost:5000/ensemble")
                if response.status_code == 200:
                    ensemble_details = response.json()
                    st.success(ensemble_details.get("message", "Ensemble model
created successfully!"))
                    st.json(ensemble_details.get("ensemble_details", {}))
                else:
                    st.error(f"Error during ensemble creation: {response.text}")
            except Exception as e:
                st.error(f"An error occurred during ensemble creation: {e}")
    else:
        st.warning("Please upload and profile your dataset first.")

# Prediction
if choice == "Prediction":
    st.title("Make Predictions")
    if 'df' in locals():
        st.write("Upload a CSV file for prediction:")
        prediction_file = st.file_uploader("Upload Prediction Dataset (CSV
format)", type=["csv"])
```

```
        if prediction_file:
            try:
            with open(model_path, "rb") as file:
                st.download_button("Download Trained Model", file,
file_name="best_model.pkl")
        else:
            st.error(response.json().get("error", "Model file not found."))
    except Exception as e:
        st.error(f"An error occurred: {e}")
```