IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

AN END-TO-END MACHINE LEARNING PIPELINE FOR PREDICTIVE ANALYTICS IN ONLINE FOOD DELIVERY SYSTEMS

[1]U.S.S.S.Sampath, [2]V.Jayakumar, [3]Bharati Bidikar, [4]V.Hamsa Valli, [5]V.SriCharan

[1]Student, [2]Student, [3]Adjunct Professor, [4]Student, [5]Student

[1]CSSE, [1]Andhra University, Visakhapatnam, India

[2]CSSE, [2]Andhra University, Visakhapatnam, India

[3]CSSE, [3]Andhra University, Visakhapatnam, India

[4]CSSE, [4]Andhra University, Visakhapatnam, India

[5]CSSE, [5]Andhra University, Visakhapatnam, India

Abstract: The rapid growth of online food delivery platforms has created a need for intelligent systems capable of optimising demand prediction and delivery efficiency. This paper presents *CloudPredict*, an end-to-end predictive analytics system integrating machine learning, backend APIs, and business intelligence dashboards. The system utilises real-world delivery data processed using R-based machine learning models**,** such as Random Forest and C5.0, along with K-Means clustering for customer segmentation. A Node.js backend enables API-based communication, while Power BI provides visualisation of insights. The results demonstrate high predictive accuracy and improved decision-making capabilities for logistics and marketing optimisation.

Index Terms **-** Machine Learning, Food Delivery Systems, Random Forest, Predictive Analytics, Business Intelligence, Node.js, Power BI.

I. INTRODUCTION

The expansion of online food delivery has transformed urban consumption by enabling customers to place orders through digital platforms and receive meals through distributed restaurant and courier networks. In real life, these platforms have to make quick, correct decisions when they don't know what's going to happen [1]. This includes figuring out how long it will take to deliver, predicting spikes in demand, assigning riders, handling delays, and balancing cost with service quality [1]. These decisions are harder to make in big cities because traffic jams, changing weather, events that cause surges, and geographic dispersion all make operations behave in ways that aren't linear.

The recent studies reveal a distinct research and Operational deficiency: static models and rudimentary heuristics are inadequate for the intricacies of contemporary food delivery ecosystems. One study that looked at Indian cities found that including real-time contextual variables like traffic density, weather, local events, and geospatial information made predictions more accurate. A recent study on spatio-temporal demand forecasting contended that graph-based learning more effectively captures interactions within urban zones and directional order flows for proactive resource allocation. These results encourage the creation of a unified predictive analytics system instead of separate models. Cloud Predict is suggested as a system like this: a cloud-based analytics platform that uses machine learning to make predictions in real time and in batches, can be deployed on a large scale, and helps people make decisions throughout the online food delivery process.

II. LITERATURE SURVEY

The most recent studies demonstrate the importance of predictive analytics for food delivery. Machine learning models like Random Forest, SVM, and Gradient Boosting can accurately predict delivery delays and logistics performance **[2, 3].** The combination of real-time traffic, weather, and geospatial data significantly improves prediction accuracy [4].

Deep learning methods, such as ConvLSTM and graph neural networks, are used to predict demand in spatiotemporal settings [4]. Predictive systems increase customer satisfaction and help businesses run more smoothly. Recent studies highlight that integrating advanced AI models with large-scale real-time data streams enables more precise and dynamic decision-making in food delivery systems. Additionally, the use of ensemble learning techniques enhances robustness by combining multiple model predictions to minimise errors. Cloud-based infrastructures further support scalability [7], [8].

III. PROPOSED SYSTEM

The Cloud Predict is a layered predictive analytics system with five major modules:
Data ingestion layer: The ingestion layer collects structured and semi-structured data from platform databases, partner systems, IoT-style courier location feeds, and external APIs. Evenstreaming is required for order status changes and courier movement, while scheduled batch pipelines are suitable for historical KPI aggregation and model retraining datasets. The architecture should validate schema consistency, missingness, timestamp alignment, and identifier integrity before downstream storage.

Storage and feature layer: The storage layer should combine a raw data lake, a curated analytical warehouse, and an online/offline feature store. This is important because ETA serving needs low-latency online features, whereas training jobs require historical, reproducible offline features built from the same logic. Features such as geospatial distance, average prep delay, localised demand intensity, traffic severity index, and weather-risk flags should be computed and stored for reuse across models.

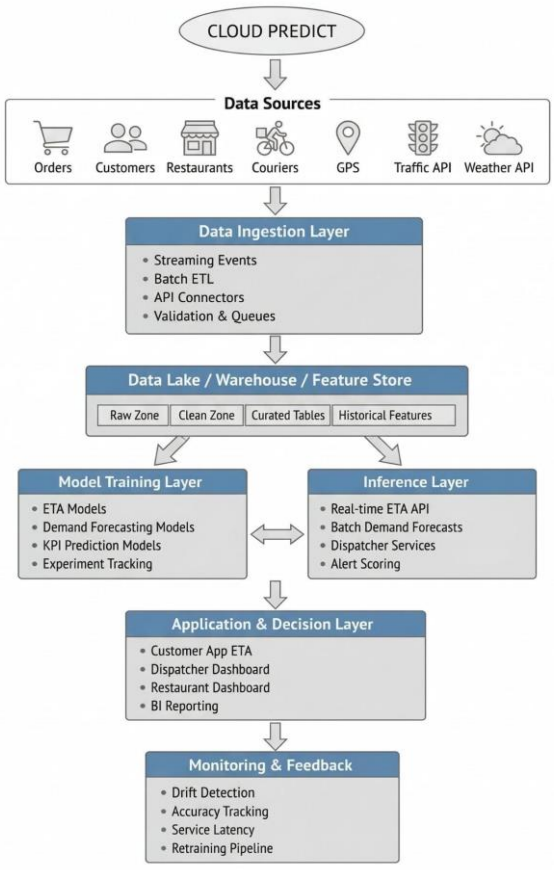

Figure 1: Model Architecture

Model lifecycle layer: The model lifecycle layer handles training, validation, experiment tracking, hyperparameter tuning, model registry, and controlled deployment. The literature shows that food delivery prediction benefits from systematic model comparison across linear, tree-based, ensemble, and advanced architectures. Therefore, Cloud Predict should support parallel experimentation and version control to ensure reproducibility and reliable rollout.

Application layer: The application layer exposes predictions through low-latency APIs and batch jobs. Real-time APIs are necessary for ETAs shown to customers and dispatch suggestions, while batch prediction is appropriate for zone demand plans, rider allocation planning, and management KPI reports. A hybrid serving pattern is therefore essential.

Monitoring layer: The monitoring layer should track prediction error, service latency, data drift, concept drift, and business impact metrics. Because traffic behaviour, weather patterns, restaurant availability, and city conditions change over time, model performance can degrade even when infrastructure remains stable. Monitoring must therefore trigger alerts and retraining pipelines when thresholds are crossed.

The architecture separates data engineering from model serving while preserving a shared feature foundation, which is consistent with current production ML platform design patterns. It also supports both offline analytical use and emphasised operational prediction, a distinction emphasised in industrial ML infrastructure descriptions.

IV. METHADOLOGY

Dataset design: The system should integrate order-level, courier-level, restaurant-level, and environment-level data. At minimum, the literature suggests including distance, weather conditions, traffic density, delivery-agent characteristics, vehicle type, multiple deliveries, time-related variables, and city characteristics for ETA prediction [3], [4]. For demand forecasting, the dataset should be aggregated into time windows by zone, capturing volume counts, inflow-outflow relationships, and neighbouring-zone interactions.

Preprocessing: The preprocessing should include missing-value treatment, type normalisation, timestamp parsing, categorical encoding, outlier handling, geospatial distance computation, and temporal feature extraction. One recent study removed rows containing null values and converted heterogeneous fields into consistent numeric and categorical forms before model development. In a production system, stricter data quality rules and imputation strategies may be preferable where operational continuity matters.

Feature engineering: The feature engineering is central to system performance. The literature specifically highlights the value of haversine distance, traffic indicators, weather states, city type, festival indicators, and temporal variables for ETA prediction [3], [4]. For demand forecasting, graph edges representing inter-zone adjacency or order-flow relationships should be engineered from historical spatial movement data.

Model training: The model training should use train-validation-test splits together with cross-validation and hyperparameter tuning where appropriate. A strong ETA training benchmark set would include linear regression, decision trees, random forests, XGBoost, LightGBM, and SVM because this family has already been compared in recent delivery prediction work. KPI models should include Random Forest and other ensemble classifiers/regressors because prior work showed strong comparative performance [2].

Evaluation: The regression tasks should use MSE, RMSE, MAE, and R^2, while classification tasks should use precision, recall, F1-score, AUC, and confusion matrices. For demand forecasting, additional temporal metrics such as MAPE or weighted error by zone can be added depending on business priorities. Online evaluation should measure not only statistical accuracy but also operational outcomes such as reduced lateness, improved rider utilisation, and higher ETA reliability.

V. RESULTS AND ANALYSIS

The dataset used in this study contains approximately 2000 records representing food delivery transactions.

Features include: Age, gender, marital status, occupation, monthly income, educational qualification, family size, medium, restaurant type, mealtime preference, ease and convenience, city tier, order time, avg cost (two people), order hour, and order frequency.

A practical deployment roadmap for Cloud Predict can follow five phases.
Foundation setup: establish cloud storage, streaming ingestion, historical warehouse, and data governance controls.
ETA pilot: build the first ETA pipeline using contextual and spatial features since this task already has strong evidence in current literature.
Demand module rollout: add zonal demand forecasting with time-window aggregation and graph-based experimentation.
KPI intelligence module: introduce management-facing predictive dashboards and risk scoring using ensemble methods.
Continuous optimisation: automate monitoring, retraining, and A/B testing to connect model performance with business outcomes.

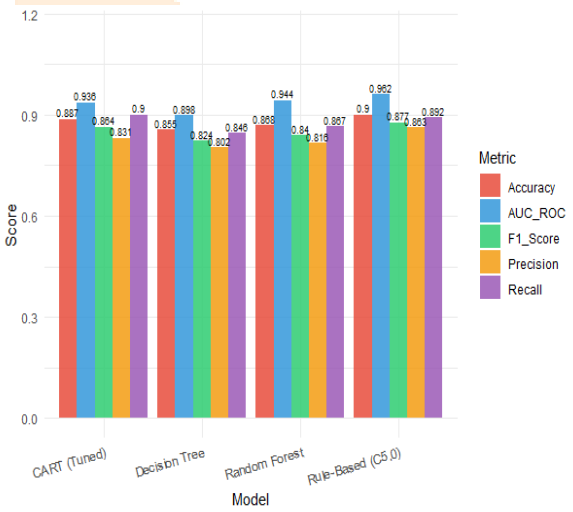

Figure 2: **Model Performance**

The Random Forest model demonstrated strong performance across multiple evaluation metrics. It achieved an accuracy of 86.8%, indicating a high proportion of correct predictions. The model also recorded an AUC-ROC score of 94.4%, reflecting excellent ability to distinguish between classes. In terms of classification balance, it obtained an F1-score of 84.0%, which represents a good trade-off between precision and recall. The precision was 81.6%, showing that most of the predicted positive cases were correct, while the recall reached 86.7%, indicating that the model successfully identified a large proportion of actual positive cases. Overall, these results highlight the effectiveness of the Random Forest algorithm for this predictive task.

Figure 3: Order Density Heatmap

Figure 3 (Order Density Heatmap) shows that order activity is relatively low during early morning hours but gradually increases throughout the day. Peak demand occurs between 12 PM and 8 PM, especially in Tier 1 cities, which consistently exhibit the highest order density. Tier 2 cities show moderate activity, while Tier 3 cities have comparatively lower demand overall.

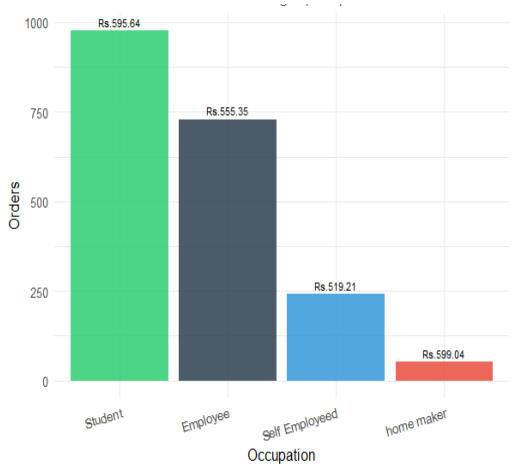

Figure 4: Orders by Customer Occupation

Figure 4 (Orders by Customer Occupation) indicates that students place the highest number of orders and also have the highest average spend per order. This is followed by employees, who contribute a significant portion of orders with moderate spending [5]. Self-employed individuals show lower order counts and spending, while homemakers contribute the least in both order volume and average order value.

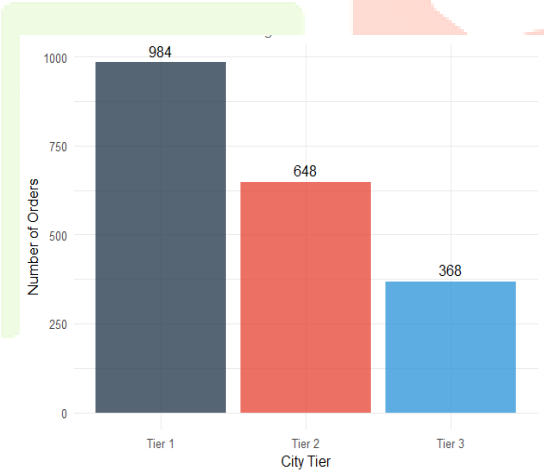

Figure 5: Order Distribution by city tier

The bar chart illustrates the distribution of orders across different city tiers. Tier 1 cities dominate with the highest number of orders (984), indicating significantly stronger demand in highly urbanised areas. Tier 2 cities follow with 648 orders, showing moderate activity, while Tier 3 cities contribute the least with 368 orders. Overall, the data suggests that order volume decreases as we move from Tier 1 to Tier 3 cities, highlighting the impact of urbanisation on customer demand in food delivery platforms.

VI. CONCLUSION

The CloudPredict system demonstrates the effectiveness of integrating machine learning with full-stack development and business intelligence tools. The system improves demand prediction and provides actionable insights for operations. The experimental results validate that ensemble learning models significantly improve prediction accuracy compared to traditional approaches. The integration of

machine learning with business intelligence tools enables data-driven decision-making in food delivery systems.

REFERENCES

[1] G. Dantzig and J. Ramser, “The Truck Dispatching Problem", *Management Science*, 1959.
[2] L. Breiman, “Random Forests", *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
[3] X. Wang et al., “Delivery Time Prediction using Gradient Boosting,” *IEEE Access*, 2019.
[4] T. Li et al., “Predicting Urban Demand using Machine Learning,” *IEEE Transactions*, 2017.
[5] R. Gupta and A. Singh, “Customer Churn Prediction in E-Commerce,” *International Journal of Data Science*, 2021.
[6] R Core Team, “R: A Language and Environment for Statistical Computing", 2023.
[7] Node.js Foundation, "Node.js", 2024.
[8] Microsoft, “Power BI Documentation", 2024.