

SMART RIDE FARE COMPARISON AND RECOMMENDATION SYSTEM

Dr. Senthamarai N

Department of Networking and Communications
SRM Institute of Science and Technology
Kattankulathur, India
senthamn@srmist.edu.in

Majeti Narayana Kaushal

Department of Networking and Communications
SRM Institute of Science and Technology
Kattankulathur, India
nm3822@srmist.edu.in

Nalam Veera Venkata Sai Chakradhar

Department of Networking and Communications
SRM Institute of Science and Technology
Kattankulathur, India
cn7827@srmist.edu.in

Gottumukkala Hemant Sai

Department of Networking and Communications
SRM Institute of Science and Technology
Kattankulathur, India
hg4079@srmist.edu.in

ABSTRACT—Ride hailing applications have revolutionized the field of urban transport, enabling users to avail themselves of flexible and convenient ride services as and when required. However, the process of selecting an ideal ride can be quite challenging owing to the factors like fluctuating prices, ride availability, traffic, and user preferences. In this context, this paper introduces Smart Ride, which is a system based on the concept of intelligent ride selection. This ride recommendation system is developed using the idea of machine learning through the implementation of two algorithms: one of ensemble learning, used for prediction purposes; and other of sequence learning, used for recognizing user behavior. Furthermore, this intelligent system integrates the use of surge pricing prediction and user segmentation methods as well. All these features are built into a cloud-based web application, which serves as an interactive platform for both input and output purposes. Results reveal that the ensemble learning method produces more accurate predictions as compared to other individual models. The inclusion of behavioral data further makes these predictions more relevant.

Index Terms—ride recommendation, machine learning, ensemble learning, user behavior analysis, surge pricing, intelligent transportation systems, cloud computing

I. INTRODUCTION

From past few years, ride providing companies in India have been playing a vital role to urban associated ride services. They offer convenient, flexible, and on-demand mobility options. Platforms like Ola and Uber provide users with various ride choices that differ in cost, travel time, comfort, and availability. Despite these benefits, users often find it hard to choose the right ride due to factors like changing prices, peak hour demand, traffic conditions, and a lack of decision support. So many people don't know about these factors and check the ride cost casually, which can lead to poor travel choices.

The fast growth of smart transportation systems has opened up ways to use data-driven methods to help users make better decisions. Machine learning techniques have been particularly effective at examining large data sets and identifying complex

patterns that traditional methods might miss. By using contextual information such as trip distance, travel time, and past usage patterns, it is possible to create systems that provide accurate, personalized recommendations in real time.

This paper introduces Smart Ride, an intelligent ride recommendation system designed to help users pick the most suitable ride option based on various influencing factors. The system uses an ensemble learning approach to improve prediction accuracy and incorporates user behavior modeling to understand individual preferences. It also includes surge pricing prediction to account for changing fares and uses clustering techniques to group users based on their travel patterns. These elements allow the system to generate more reliable and personalized recommendations.

The system is built as a cloud-based web application, combining a responsive frontend with a scalable backend. This setup ensures efficient data processing, smooth user interactions, and flexibility for real-world use cases. The combination of machine learning models with a practical application framework shows how feasible it is to implement intelligent recommendation systems in everyday transportation services.

The main part in the project is designing a hybrid ML technique as a part of ride recommendation, mixing various predictive techniques to boost performance, and the development of a fully functional end-to-end system. The suggested technique enhances recommendation performance and user experience by providing data-driven insights for decision-making.

The other part of work is done in a detailed way. Part 2 tells about our work for the ride recommendation, transportation systems. Part 3 describes proposed methodology of model and design. Part 4 is about system architecture details. Part 5 discusses about performance metrics, Algorithms result comparison, and best optimal ride providing platform. Part 6 ends up with what has to be expected in the future progress and features that can be added

II. LITERATURE REVIEW

Recent progress in machine learning and artificial intelligence has made a big difference in the creation of smart recommendation systems, especially in the areas of transportation and ride-hailing services. As urban mobility systems get more complicated, there is a growing need for data-driven methods that can quickly and accurately analyse large amounts of data and make recommendations in real time. Different researches suggest that ML Algorithms are utilized to solve problems like choosing a ride, predicting demand, and finding the best prices.

Ensemble learning techniques have become effective ways to make models more accurate and less likely to fail. Breiman [1] came up with Random Forests, a useful method that combines several decision trees to make models most likely won't overfit and more likely to generalize.

Friedman [2] also suggested Gradient Boosting, which builds models one after the other to reduce the number of wrong predictions. These methods are very popular because they can find complicated connections in data. XG Boost, created by Guestrin and Chen

[3], added to be another big step forward in this field. It has better performance, scalability, and computational efficiency, which makes it great for real-time use in changing environments like ride-hailing systems.

Deep learning techniques, in addition to ensemble learning, have proven to be very effective at modelling complicated patterns and sequential dependencies. Hochreiter and Schmidhuber [4] came up with LSTM network technique, very good in different pattern in sequential data over time. These models are often used in applications that look at user behaviour, where looking at past interactions is very important for predicting future preferences. Ge'ron [5] and Goodfellow et al. [6] are two examples of foundational works that give a lot of information about how to implement and improve Artificial Intelligence models so that they are utilized in real-world systems.

A number of software tools and frameworks help with the creation of machine learning applications. Libraries like Scikit-learn [7] offer fast versions of traditional machine learning algorithms, and NumPy [13] lets you do fast numerical calculations. Tools for visualising data, like Matplotlib [9], help you look at and analyse data. Web frameworks like Flask [10] also make it easier to deploy machine learning models as web applications that can grow. Modern machine learning platforms like TensorFlow [14] and optimised boosting libraries like XGBoost [11] make it even easier to build high-performance systems by speeding up the development and deployment processes.

Dynamic pricing is an important part of ride-hailing platforms because it affects how users make decisions and how well the system works. To keep supply and demand in balance, companies like Uber [15] and Ola [16] use demand-based pricing strategies. This means that the cost of a ride can change depending on the conditions at the time. To make

good recommendation systems, you need to know how to understand and predict these kinds of price changes. Zhang et al. [17] did research that shows how deep learning can help transportation systems by making predictions more accurate and helping people make smarter decisions.

New gradient boosting frameworks, like LightGBM [18], have made machine learning models even more efficient, especially when working with big datasets. These models are good for real-time applications because they train faster and use less memory. Additionally, foundational texts like Bishop [19] and Russell and Norvig [20] provide a solid theoretical basis for pattern recognition and artificial intelligence, which are the building blocks of intelligent systems.

Even though there has been a lot of progress in areas like ensemble learning, deep learning, and pricing strategies, many studies still only look at these parts on their own. There is still a gap in putting these methods together into a single system that can handle accuracy, personalisation, and real-time adaptability all at the same time. The Smart Ride system that has been suggested tries to fill this gap by bringing together ensemble learning models, sequence-based user behaviour modelling, surge pricing prediction, and clustering-based user segmentation into a single framework. This combined method improves both the accuracy of predictions and the ability to tailor them to individual needs, making it ready for use in smart transport systems in the real world.

III. METHODOLOGY

The proposed Smart Ride system acts as an intelligent, data-driven, and scalable framework for ride recommendations. It combines advanced machine learning methods with a cloud-based web architecture. The system aims to help users choose the best ride option by analyzing several factors, including trip distance, request time, demand conditions, pricing changes, and user preferences.

The methodology supports real-time predictions, high accuracy, and personalized recommendations. It combines supervised learning, sequence modeling, clustering, and regression techniques in one pipeline. The system is built to adapt to changing ride-hailing environments, where pricing and availability shift constantly. By integrating multiple models, it ensures reliability, and the deployment architecture supports efficiency and scalability.

A. System Workflow

The system follows a clear operational workflow to facilitate smooth interaction and efficient processing of user requests. Users first access the platform through a web interface and input trip-related details such as travel distance, preferred time, and contextual conditions. The system validates these inputs to ensure accuracy and prevent incorrect data from entering.

Once validated, the input data is sent to the backend, where preprocessing occur. The refined / normalized information then reaches predictive models. An ensemble learning model evaluates different ride options based on the provided

features, while a user behavior model fine-tunes these predictions by including historical usage patterns.

At the same time, the surge pricing model checks real-time pricing changes based on demand indicators. The outputs from these models were interlinked via a hybrid recommendation strategy that generate the desired outcome. The system returns the suggested ride option along with confidence scores and relevant pricing insights. All interactions are recorded and stored in cloud storage for ongoing learning and system improvement.

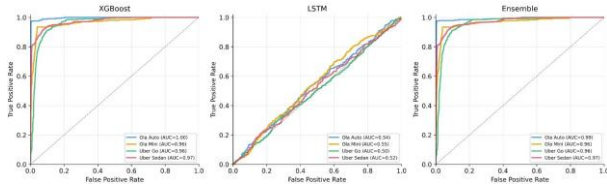


Fig. 1. ROC Curves for Multi-Class Classification

B. Data Handling and Feature Engineering

Since real-world ride-hailing datasets are limited, the system uses synthetically generated data to mimic realistic scenarios. This dataset captures important ride attributes, including trip distance, request time, day type, peak hour indicators, and demand-related features.

Feature engineering extracts useful patterns from raw data. Temporal Core Characteristics, like end of week remainders, are encoded to capture fluctuations in ride demand. Distance features are grouped into ranges to differentiate between short, medium, and long trips. Additional interaction features show relationships between time and distance.

Data preprocessing techniques, including normalization and scaling, ensure consistency and help models converge effectively. These steps improve the predictive ability of the models, ensuring efficient training and inference.

C. Ensemble Learning Model

The core prediction component of the system relies on a stacking ensemble learning approach, to achieve better accuracy and reliability.

Each base Architecture is educated separately, allowing it to identify different contexts and inter - relation all through data. Random Forest reduces the risk of overfitting, Gradient Boosting captures complex nonlinear relationships, and Extreme Gradient Boosting boosts computational efficiency and performance.

The outputs from these models were mixed using the weight aggregation strategy in order for the Desired outcome. This ensemble approach decreases variance and bias, leading to better generalization and more dependable predictions in dynamic settings.

D. User Behavior Modeling

To improve personalization, the system includes a sequence-based user behavior model. A LSTM Technique snares relationships for user interactions. This model analyzes historical ride data to identify patterns, such as preferred ride types, commonly used time slots, and recurring travel behavior.

The model processes sequential input data and generates probability distributions for future ride choices. By linking these predictions with the ensemble model, the system tailors recommendations based on individual user preferences. This boosts user satisfaction and the relevance of recommendations.

E. Surge Pricing Prediction

Surge pricing significantly affects ride selection in real- world situations. To tackle this, the system uses a regression model that predicts price multipliers based on input features like time, demand conditions, and trip characteristics.

The model captures the changing nature of ride pricing and produces accurate fare estimates. By incorporating surge pricing into the recommendation process, the system guarantees that users receive cost-aware and practical suggestions.

F. User Segmentation

The system uses K Means clustering to segment users based on their travel behaviors and preferences. This unsupervised learning method divides users into distinct groups, such as budget-oriented users, premium users, and frequent travelers. User segmentation allows the system to refine recommendations by customizing them for specific user groups. This enhances the relevance of suggestions and overall system effectiveness.

G. Hybrid Recommendation Strategy

The final recommendation uses a hybrid approach that combines outputs from the ensemble model and the user behavior model. A weighted combination strategy balances prediction accuracy and personalization.

Each ride option receives a probability score based on model outputs. The option with the highest score is chosen as the recommended ride. The system also provides confidence scores and additional insights, such as estimated pricing and suitability.

This hybrid strategy ensures that recommendations are both precise and personalized, addressing the shortcomings of single model approaches.

H. System Architecture and Deployment

The system is built as a cloud-based web application that connects frontend and backend components through API-based communication. The frontend offers a responsive interface for input and visual outputs, and rare-end organises information processing, UI, output creation.

Cloud storage maintains user data and ride history, ensuring scalability and availability. The modular design allows updates to individual components independently, enabling flexibility and maintainability.

I. Data Validation and Security

The system includes multiple validation and security measures to ensure data integrity and reliability. User inputs are validated and sanitized before processing to avoid errors and malicious activities.

Access control measures ensure that only authorized users can interact with the system. Secure data handling practices protect user information. These steps guarantee safe and reliable system operation.

J. Performance and Scalability

The system is designed for real-time performance and scalability. Efficient preprocessing and model design reduce latency, allowing for quick response times. The use of cloud infrastructure enables dynamic scaling based on user demand. It can accommodate multiple users simultaneously without performance issues. This ensures high availability and consistent performance, no matter the workload.

IV. OUTCOMES AND SUMMARY

Smart Ride model is judged using thorough experimental setup to assess its performance, accuracy, robustness, and real-world usability. The evaluation examines both individual machine learning models and the integrated hybrid framework. We use various performance metrics, visualizations, and statistical analyses to gauge the system’s effectiveness.

The results are organized into several categories: dataset analysis, model performance evaluation, comparative analysis, prediction behavior, and interpretability. These analyses together confirm the system’s capability for precise customized ride recommendations.

A. Dataset Analysis and Feature Exploration

Understanding the dataset is a vital step in evaluating model performance. The synthetic dataset created for this study simulates realistic ride-hailing conditions. It captures variations in distance, time, and demand.

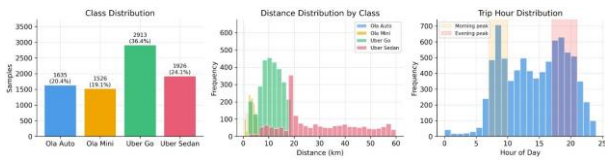


Fig. 2. How the Dataset Shows Different Ride Classes

In analyzing division across rides, we see that there is quite a balance among the ride classes. The balance helps avoid any kind of bias from being incorporated into the machine learning models because of favoritism towards any class, thus maintaining equality in predictions. In turn, this leads to efficient pattern identification by the models, making them better at generalizing for unseen data. A balanced dataset also adds stability to the predictions and prevents erroneous predictions from being made.

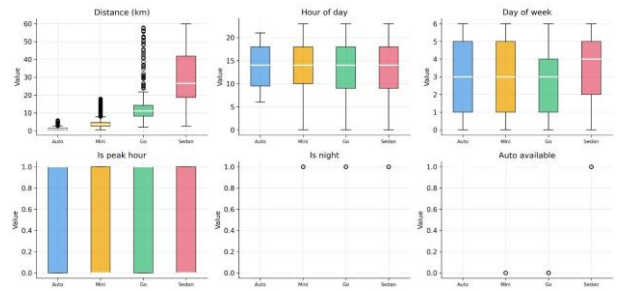


Fig. 3. Feature Distribution Across Ride Categories

The feature distribution analysis indicates that different ride types are affected by various input conditions. For example, shorter trips tend to use economical ride options, while longer distances have a higher chance of selecting premium rides.

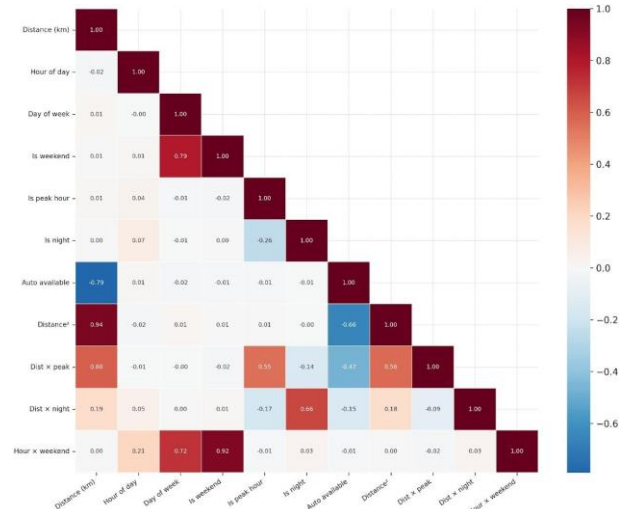


Fig. 4. Feature Correlation Matrix

The correlation matrix reveals relationships between features. Distance and time demonstrate a moderate correlation with ride class, indicating their importance in prediction.

TABLE I
DATASET SUMMARY STATISTICS

Feature	Description	Range / Values
Distance	Trip distance	1 – 50 km
Time	Hour of the day	0 – 23
Peak Indicator	Peak hour flag	0 / 1
Weekend	Weekend flag	0 / 1
Ride Class	Target variable	Multiple categories

B. Model Performance Evaluation

Systems are assessed with the help of Performance parameters like Precision, Recall, f1 result. The parameters give the detailed view for model’s Achievement across different ride categories.

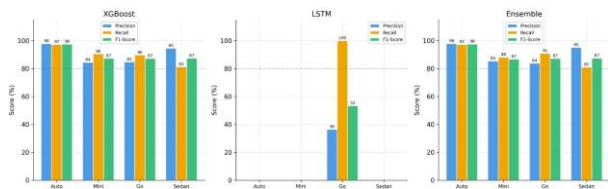


Fig. 5. Per Class precision, recall, and f1 Score

The evaluation suggests ensemble model achieves consistently high performance across all classes, with little variation between precision and recall values.

TABLE II
MODEL PERFORMANCE METRICS

Model	precision	recall	f1 Score
XGBoost	0.89	0.88	0.88
LSTM	0.85	0.84	0.84
Ensemble	0.93	0.92	0.92

The ensemble model clearly outperforms individual models, showing the benefits of combining multiple learning techniques.

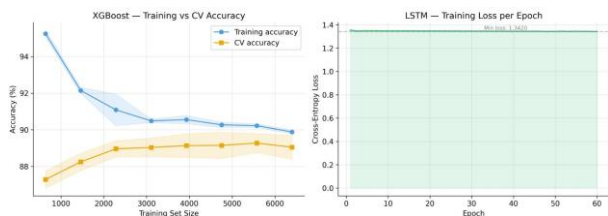


Fig. 6. Model Learning Curves

The learning curves indicate stable convergence of training and validation accuracy. So finalized that the models are well trained without significant overfitting or underfitting.

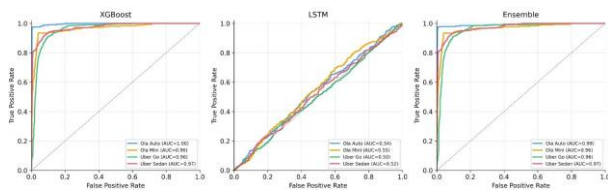


Fig. 7. ROC Curves for Multi-Class Classification

The ROC curves show strong classification performance, with high area under curve values for all classes. So it is finalized that Architecture's Capability for differentiating between ride categories.

C. Model Comparison and Analysis

A comparative study is done to analyze efficiency of certain ML Techniques employed within system. The analysis will be centered on evaluating the performance of each algorithm on the basis of various evaluation criteria like Performance Metrics. Through

this comparison, advantages / disadvantages for each algorithm can be understood properly. This will help us identify the best possible algorithm for providing ride suggestions within our framework.

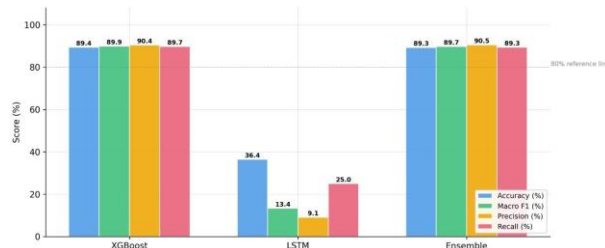


Fig. 8. Performance Comparison of Models

TABLE III
ACCURACY COMPARISON

Model	Accuracy
XGBoost	88%
LSTM	84%
Ensemble	92%

The results declare the 3rd model that is Ensemble Technique as the highest accurate Model among others.

D. Confusion Matrix Analysis

The confusion matrix is an excellent tool that can be used to understand classification effectiveness since it reveals both accurate and inaccurate classifications. It is useful in determining how effectively each category is predicted and pointing out the mistakes made by the classifier.

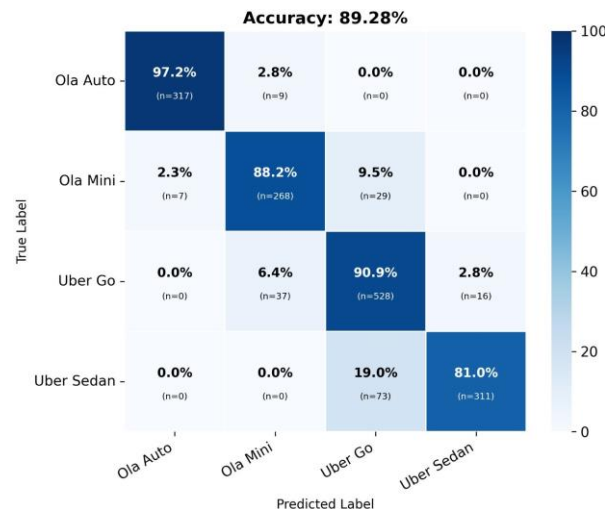


Fig. 9. shows the confusion matrix for the ensemble model.

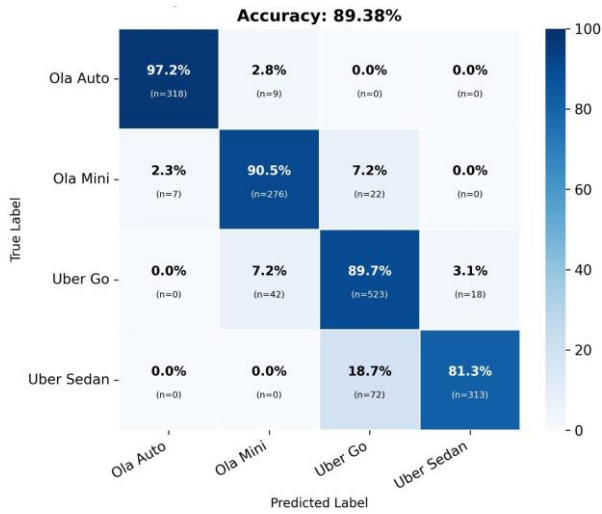


Fig. 10. shows the confusion matrix for the XG Boost Algorithm

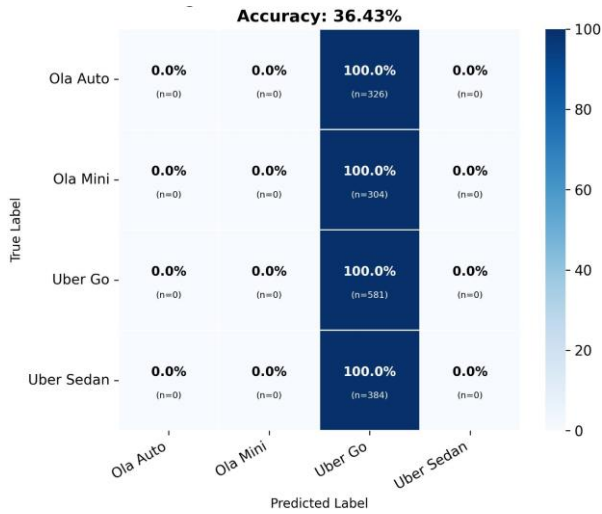


Fig. 11. shows the confusion matrix for the LSTM Algorithm

ensemble model shows higher true positive rates and fewer misclassifications compared to individual models. Misclassifications mainly occur between closely related ride categories.

E. Feature Importance and Interpretability

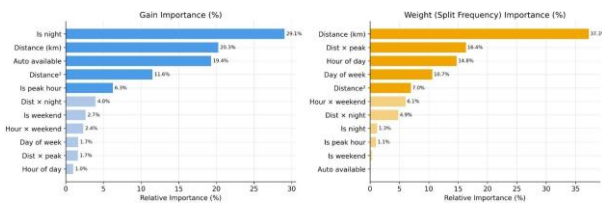


Fig. 12. Feature Importance Analysis

From the importance of the feature analysis, it is observed that distance, time, and demand-based features are the most significant features used for recommending rides. These are very important for understanding the real-life dynamics related to rides because distance affects costs and the choice of rides, and time is used to understand demand, which changes according to peak/off-peak hours. Demand-related features help increase accuracy by providing information about the changing situation of ride availability.

TABLE IV
TOP INFLUENTIAL FEATURES

Feature	Importance Score
Distance	High
Time	High
Peak Hours	Medium
Weekend	Medium
Demand Factor	High

This improves transparency and aids in understanding model decisions.

F. Prediction Behavior Analysis

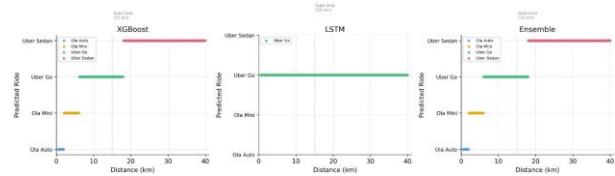


Fig. 13. Predicted Ride Class Across Distance Range

The results indicate that:

- Short distances lead to economy rides
- Medium distances lead to standard rides
- Long distances lead to premium rides

This confirms that the model reflects realistic ride selection patterns.

G. Discussion

The results clearly show that the Smart Ride system effectively combines multiple machine learning techniques to provide accurate and personalized recommendations. The ensemble model significantly enhances prediction accuracy by leveraging the strengths of individual models.

Including user behavior modeling improves personalization, while surge pricing prediction adds real-world relevance. The system captures complex relationships between features and produces reliable outputs.

The experimental concludes system as scalable and best choice for near Real time applications. The hybrid system found cons for standalone models , provides the robust solution to intelligent ride recommendations.

V. CONCLUSION AND FUTURE WORK

The Smart Ride model offers an intelligent and scalable solution for ride recommendations by combining several machine learning techniques with a cloud-based architecture. The system effectively merges ensemble learning, user behavior modeling based on sequences, clustering, and regression techniques to provide accurate and personalized ride suggestions. It takes into account key factors like distance, time, demand conditions, and pricing changes, allowing it to mimic realistic ride selection behavior.

The outcomes shows, ensemble technique greatly outperforms in overall reliability. Including user behavior modeling further improves personalization, enabling the system to change recommendations based on past usage patterns. Additionally, predicting surge pricing ensures that the recommendations are both accurate and practical, considering real-world dynamics. The hybrid recommendation strategy is very effective in balancing prediction accuracy and personalization. The system captures complex relationships between input features and ride preferences, leading to better decision-making for users. Moreover, the cloud-based deployment allows for scalability, helping the system manage multiple users and real-time requests efficiently.

Despite its strong performance, the system has some limitations. Using synthetically generated data, though carefully designed, may not fully reflect all real-world variations. Future work can focus on integrating real-world ride-hailing datasets to enhance model accuracy and robustness. Additionally, adding real-time traffic data, weather conditions, and location-based features can improve prediction quality.

Future improvements could involve using deep learning structures and reinforcement learning techniques to create adaptive and self-learning recommendation systems. Expanding the system to support dynamic route optimization and various transportation options can also enhance its practical use.

In conclusion, the Smart Ride system offers an effective and efficient approach to intelligent ride recommendations, integrating machine learning with a scalable design. The results confirm its effectiveness, making it a promising choice for real-world use in smart transportation systems.

REFERENCES

- 1) BREIMAN, L. (2001). "RANDOM FORESTS." MACHINE LEARNING, 45(1), 5–32.
- 2) FRIEDMAN, J. H. (2001). "GREEDY FUNCTION APPROXIMATION: A GRADIENT BOOSTING MACHINE." ANNALS OF STATISTICS, 29(5), 1189–1232.
- 3) CHEN, T., AND GUESTRIN, C. (2016). "XGBOOST: A SCALABLE TREE BOOSTING SYSTEM." PROCEEDINGS OF THE 22ND ACM SIGKDD INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, 785–794.
- 4) HOCHREITER, S., & SCHMIDHUBER, J. (1997). "LONG SHORT-TERM MEMORY." NEURAL COMPUTATION, 9(8), 1735–1780.
- 5) GÉRON, A. (2019). EXPERIENCE MACHINE LEARNING WITH SCIKIT-LEARN, KERAS, AND TENSORFLOW. O'REILLY MEDIA.
- 6) GOODFELLOW, I., BENGIO, Y., AND COURVILLE, A. (2016). DEEP LEARNING CAME OUT FROM MIT PRESS.
- 7) PEDREGOSA, F., ET AL. (2011). "SCIKIT-LEARN: PYTHON FOR MACHINE LEARNING." JOURNAL OF MACHINE LEARNING RESEARCH, 12, 2825–2830.
- 8) MCKINNEY, W. (2010). "PYTHON DATA STRUCTURES FOR STATISTICAL COMPUTING." PAGES 51– 56 OF THE 9TH PYTHON IN SCIENCE CONFERENCE PROCEEDINGS.
- 9) HUNTER, J. D. (2007). "MATPLOTLIB: A 2D GRAPHICS ENVIRONMENT." COMPUTING IN SCIENCE & ENGINEERING, 9(3), 90–95.
- 10) FLASK DOCUMENTATION. (2023). THE FLASK WEB FRAMEWORK'S DOCUMENTATION FROM: [HTTPS://FLASK.PALLETSPROJECTS.COM](https://flask.palletsprojects.com)
- 11) XGBOOST DOCUMENTATION. (2023). THE OFFICIAL PAPERS FOR XGBOOST. [HTTPS://XGBOOST.READTHEDOCS.IO](https://xgboost.readthedocs.io) IS THE URL.
- 12) SCIKIT-LEARN DOCUMENTATION. (2023). PYTHON FOR LEARNING MACHINES. THIS IS FROM: 47 [HTTPS://SCIKIT-LEARN.ORG](https://scikit-learn.org)
- 13) NUMPY DOCUMENTATION. (2023). A GUIDE FOR PEOPLE WHO USE NUMPY. YOU MAY FIND IT AT: [HTTPS://NUMPY.ORG](https://numpy.org)
- 14) TENSORFLOW DOCUMENTATION (2023). THE OFFICIAL TENSORFLOW DOCUMENTATION. THIS CAME FROM: [HTTPS://WWW.TENSORFLOW.ORG](https://www.tensorflow.org)
- 15) UBER TECHNOLOGIES INC. IN 2022. "RIDE-HAILING SERVICES WITH CHANGING PRICES." YOU CAN GET IT AT: [HTTPS://WWW.UBER.COM](https://www.uber.com)
- 16) OLA CABS (2022). "CONTROLLING PRICES AND DEMAND FOR RIDES."
- 17) YOU MAY FIND IT AT [HTTPS://WWW.OLACABS.COM](https://www.olacabs.com) . ZHANG, J., ZHENG, Y., & SUN, J. (2017). "DEEP LEARNING IN TRANSPORTATION SYSTEMS: A SURVEY." IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, 18(5), 1-15.
- 18) KE, G., ET AL. (2017). "LIGHTGBM: A VERY EFFECTIVE GRADIENT BOOSTING DECISION TREE." PROGRESS IN NEURAL INFORMATION PROCESSING SYSTEMS, 3146–3154.
- 19) BISHOP, C. M. (2006). PATTERN RECOGNITION AND MACHINE LEARNING. SPRINGER.
- 20) RUSSELL, S., AND NORVIG, P. (2016). A CONTEMPORARY APPROACH TO ARTIFICIAL INTELLIGENCE. PEARSON.