Smart Driver Recommendation and Booking System Using Ml and Ds

Surendra Kumar Choudhary[1], Pola Sailikitha[2], Niddana Leela Saikumar[3], Patnana Thanu Sri[4], Palla Sai Sarvan[5]
[1,2,3,4,5]*Department of Computer Science and Engineering, Avanthi Institute of Engineering and Technology*

***Abstract*—The ride-hailing industry faces a significant challenge in fairly and accurately pricing insurance for drivers, as traditional models rely on static premiums that do not account for real-time driving behavior. This paper presents a Smart Driver Recommendation and Booking System using Machine Learning (ML) and Data Structures (DS) that assesses driver risk and recommends suitable drivers dynamically. The proposed system classifies drivers as Good, Average, or Poor based on trip-level behavioral metrics such as average speed, maximum speed, harsh braking events, rapid acceleration, and traffic violations. These classifications directly determine daily insurance premiums and driver recommendations, enabling personalized and fair service. The system is implemented as a Flask-based web application offering three distinct modules: a user (passenger) interface with AI-powered driver recommendations, a driver portal for registration and trip management, and an administrative dashboard for monitoring insurance policies, claims, and financial reports. A dataset of 40 drivers with over 600 synthetic trips was generated to simulate real-world conditions. Results demonstrate accurate behavioral classification, dynamic premium adjustments, and streamlined insurance claim processing. The system serves as a scalable, data-driven framework for modernizing driver recommendation and insurance in the gig economy.**

***Index Terms*—Behavior-Based Insurance, Booking System, Data Structures, Driver Recommendation, Dynamic Premium, Flask, Machine Learning, Ride-Hailing, Risk Classification, Usage-Based Insurance.**

I. INTRODUCTION

The rise of app-based ride-hailing platforms such as Uber, Ola, and Rapido has fundamentally transformed urban transportation. Millions of drivers operate daily under conditions of significant risk, yet insurance models in this sector largely remain static applying the same premium to all drivers regardless of individual driving habits or risk profiles.

Traditional insurance models fail to capture real-time behavioral data, leading to inequitable pricing: safe drivers subsidize reckless ones. Usage-Based Insurance (UBI) and Behavior-Based Insurance (BBI) have emerged as alternative paradigms, where telematics and smartphone sensors monitor driver behavior continuously. However, adoption in developing economies remains limited due to infrastructure constraints.

This research presents a comprehensive Smart Driver Recommendation and Booking System that leverages machine learning and data structures to assess driver risk and compute insurance premiums dynamically. The contributions of this work are:

- A rule-based ML classifier that labels drivers as Good, Average, or Poor using five behavioral features.
- A dynamic premium engine that adjusts insurance costs daily based on trip-level behavioral data.
- An AI-powered driver recommender that matches passengers with suitable drivers considering behavior, distance, and insurance coverage.
- A Flask web application integrating driver registration, trip logging, claim management, and administrative reporting.

II. LITERATURE SURVEY

A. Usage-Based Insurance (UBI)

Bolder Dijk et al. (2011) demonstrated that usage-based insurance significantly reduces accident rates by incentivizing safer driving behavior. Telematics-based systems monitor speed, braking, and acceleration to compute personalized premiums.

B. Machine Learning in Risk Classification

Weidner et al. (2017) applied machine learning algorithms including Random Forest and Support Vector Machines to driver telematic data, achieving accurate risk segmentation. Supervised classification models outperformed traditional actuarial scoring in dynamic environments.

C. Ride-Hailing Insurance Challenges

Ranjbari et al. (2020) identified coverage gaps in the gig economy, where drivers often lack adequate insurance during app-on periods. The study recommended real-time, behavior-linked insurance products tailored to the ride-hailing context.

D. Recommender Systems in Transportation

Collaborative filtering and content-based recommendation techniques have been applied to driver-passenger matching (Wang et al., 2019). Incorporating behavioral scores into recommender pipelines improves safety outcomes and user satisfaction.

E. Research Gaps

Despite advances in UBI and ML-based risk modeling, an integrated system combining behavioral classification, dynamic premium computation, driver recommendation, and insurance claim management in a single platform for ride-hailing remains underexplored. This work addresses that gap.

III. PROPOSED SYSTEM

A. System Overview

The proposed system is a multi-role web application comprising three modules: (1) User/Passenger Module, (2) Driver Module, and (3) Admin Module.

User → [Recommender Engine] → Driver List (with behavior scores and insurance) → Trip Booking

Driver → [Registration + Trip Logging] → Behavioral Scoring → Dynamic Premium

Admin → [Dashboard] → Policies + Claims + Daily/Monthly Reports

B. Key Modules

- User Module: Allows passengers to log in, view AI-recommended drivers, and initiate trips.
- Driver Module: Supports driver registration (with or without insurance), trip recording, and personal insurance policy viewing.
- Admin Module: Provides full visibility into insurance policies, approved claims, and financial performance reports.

C. Novelty

Unlike static insurance models, the proposed system calculates premiums daily based on actual behavioral data. Newly registered drivers are subjected to a 30-day restriction limiting them to short-distance trips (up to 5 km), reducing risk exposure during the probationary period.

IV. METHODOLOGY

A. Dataset Generation

A synthetic dataset was generated using a custom Python script (dataset_generator.py) simulating 40 drivers (D_01 to D_40) with 12 to 18 trips each, totalling over 600 trip records. Each trip record captures:

- avg_speed: Average speed during the trip (km/h)
- max_speed: Maximum speed recorded (km/h)
- harsh_braking: Number of sudden braking events
- rapid_acceleration: Number of rapid acceleration events
- violations: Number of traffic violations
- night driving: Binary indicator (0/1) for night trips
- lat, lon: GPS coordinates simulating trip destination
- distance and time: Trip distance (km) and duration (minutes)

Trip destinations were categorized as short (within 1 km of user), medium (within 20 km), or long (within 500 km) to simulate realistic ride-hailing demand patterns across the Hyderabad region.

B. Behavioral Classification Model

The classification logic (model.py) uses a scoring function that accumulates risk points based on five behavioral features as shown in Table I.

Table I. Behavioral Risk Scoring Rules

Condition	Risk Points	Feature
avg_speed > 75 km/h	+1	avg_speed
max_speed > 100 km/h	+1	max_speed
harsh_braking > 5	+1	harsh_braking
rapid_accel > 4	+1	rapid_accel
violations > 0	+2	violations

The total risk score determines the driver's behavior class and rating as shown in Table II.

Table II. Behavior Class and Rating

Risk Score	Behavior Class	Rating
0 - 1	Good	4.5 / 5.0
2 - 3	Average	3.5 / 5.0
4 - 6	Poor	2.5 / 5.0

C. Dynamic Premium Calculation

Insurance premiums are calculated dynamically based on the coverage tier selected by the driver as shown in Table III.

Table III. Insurance Premium Tiers

Coverage (INR)	Annual Premium	Daily (INR)
1,00,000	5,000	14
2,00,000	10,000	27
3,00,000	15,000	41

The system collects trip-based earnings and computes profit as: Profit = Total Collected - Daily Premium Paid. This data is stored in daily and monthly insurance report JSON files accessible to administrators.

D. Recommender System

The recommender module (recommender.py) operates as follows:

1. Load historical trip data from drivers_trips.csv.
2. Load newly registered driver records from registered drivers. json.
3. For each driver, sample a representative trip and classify behavior using the ML model.
4. Apply 30-day restriction: new drivers are limited to trips of 0.5 km to 5 km.
5. Assign pickup distance, ETA, vehicle type, and insurance availability.
6. Shuffle and return the complete recommendation list to the user interface.

New drivers with no trip history are assigned a default behavior class of 'Good' with a rating of 5.0, reflecting a benefit-of-the-doubt policy for new entrants subject to the distance restriction.

V. SYSTEM ARCHITECTURE

A. Application Stack

The system is built using the Flask micro-framework (Python) for the backend, with Jinja2 templates and HTML/CSS/JavaScript for the frontend. Data persistence is achieved through JSON files and a CSV dataset, eliminating the need for an external database and simplifying deployment.

- drivers_trips.csv Historical trip dataset for 40 drivers.
- registered drivers. json new driver registrations.
- insurance policies. json Active insurance policies.
- insurance claims. json Approved and pending claims.
- daily_insurance_report. json Day-level financial data.
- monthly_insurance_report. json Month-level aggregated financial data.

VI. IMPLEMENTATION

A. Driver Registration and Onboarding

New drivers register via the driver portal by providing name, phone number, email, and vehicle type. Three vehicle options are supported: Bike (no insurance), Car (no insurance), and Car with Insurance. Upon registration, a unique driver ID is assigned (e.g., D_41, D_42), and an insurance policy is generated automatically if the driver selects the insured vehicle option. The registration timestamp is stored to enforce the 30- day trial restriction.

B. Behavioral Scoring and Premium Assignment

For each trip completed by a driver, the behavioral classification model evaluates five metrics and assigns a risk score. The score maps to a behavior class (Good/Average/Poor) which determines the driver's insurance tier and daily premium. The premium is deducted from the driver's daily earnings and logged in the daily report. The model is intentionally rule-based and interpretable, enabling administrators to audit classification decisions.

C. Data Flow

The data flow of the system follows the pipeline below:

Passenger Login → Recommender Engine → Driver Scoring → Driver List Display → Trip Selection
Driver Registration → Vehicle/Insurance Selection → Policy Generation → Trip Logging → Premium Deduction
Admin Login → Dashboard → Policy View / Claims Approval / Report Generation

D. Claims Management

The admin module provides an interface for reviewing and approving insurance claims. Each claim record includes a claim ID, driver ID, policy number, accident date, claim amount, status, settlement date, and reason. As of the data snapshot, six claims totaling INR 4,65,000 have been processed, including incidents of road collisions, night-trip accidents, and fractures sustained during long-distance trips.

E. File Structure

- app.py Main Flask application with routes for all three modules.
- model.py Behavioral classification function.
- recommender.py Driver recommendation engine.
- dataset_generator.py Synthetic trip data generation script.

F. Financial Reporting

The system generates both daily and monthly insurance reports. Each record captures the insurance coverage tier, daily premium paid, total earnings collected, and net profit. The monthly report aggregates these values by driver and month, providing administrators with a comprehensive view of portfolio financial performance. Drivers generating negative monthly profit are flagged for review.

VII. RESULTS AND DISCUSSION

A. Dataset Statistics

The generated dataset contains 40 historical drivers (D_01 to D_40) and 5 newly registered drivers (D_41 to D_45). The trip dataset comprises over 600 records with a balanced distribution of short, medium, and long trips.

Driver registration data includes 2 drivers without vehicle type recorded and 3 with declared vehicle types (2 with insurance, 1 without).

B. Claims Analysis

Six insurance claims were filed between March 24 and March 25, 2026 as shown in Table IV.

Table IV. Insurance Claims Summary

Claim ID	Driver	Reason	Amt (INR)	Status
CLM-001	D_03	Road collision	25,000	Approved
CLM-002	D_35	Night accident	90,000	Approved
CLM-003	D_19	Night accident	1,50,000	Approved
CLM-004	D_37	Head injury	1,00,000	Approved
CLM-005	D_38	Leg fracture	50,000	Approved
CLM-006	D_28	Night accident	50,000	Approved

handles multi-role authentication, session management, and report generation with minimal latency.

VIII. CONCLUSION

This paper has presented a complete Smart Driver Recommendation and Booking System using ML and DS for ride-hailing drivers, integrating machine learning-driven risk classification with dynamic premium pricing, AI-powered driver recommendation, and end-to-end insurance management. The system successfully demonstrates that driving behavior data specifically speed patterns, braking events, acceleration habits, and violation history can be leveraged to compute fair, personalized insurance premiums in real time.

The multi-role Flask web application provides a practical implementation adaptable to real-world ride-hailing environments. The 30-day new-driver restriction policy introduces a graduated onboarding mechanism that limits risk exposure while allowing new drivers to build their behavioral track record. Financial reporting capabilities enable administrators to monitor portfolio profitability and identify underperforming policies proactively.

The system achieves its primary objectives: equitable insurance pricing, transparent risk classification, and streamlined claim management establishing a scalable foundation for modernizing insurance in the gig economy.

A. Financial Performance

The monthly insurance report for February 2026 shows 14 active drivers. Aggregate monthly profit across all February policies was INR 1,280 (excluding one loss-making policy for D_26 at -INR 16). The highest single-day profit recorded was INR 239 (D_03, February 24, 2026). March 2026 data include 7 driver policies with a combined monthly profit of INR 703, indicating healthy portfolio growth.

B. System Performance

The behavioral classification model operates in O (1) time per trip evaluation, making it suitable for real-time use. The recommender system processes all drivers (up to 45 in the test environment) in under one second on commodity hardware. The Flask application

IX. FUTURE SCOPE

- Integration with real-time GPS telematics APIs for live trip monitoring and continuous behavioral scoring.
- Replacement of the rule-based classifier with deep learning models (LSTM or Transformer) trained on large-scale real trip data for improved accuracy.
- Expansion of the insurance model to include vehicle damage and third-party liability coverage in addition to personal accident cover.
- Development of a native mobile application (Android/iOS) for drivers, enabling in-app premium payment, claim filing, and real-time behavioral feedback.
- Integration with payment gateways (Razorpay, UPI) for automated premium collection and claim disbursement.
- Multi-language support to serve diverse driver demographics across India.

REFERENCES

[1] J. W. Bolderdijk, J. Knockaert, E. M. Steg, and E. T. Verhoef, "Effects of pay-as-you-drive vehicle insurance on young drivers' speed choice: Results of a Dutch field experiment," Accident Analysis & Prevention, vol. 43, no. 3, pp. 1181–1186, 2011.

[2] W. Weidner, F. W. G. Transchel, and R. Weidner, "Classification of scale-sensitive telematic observables for risk individual pricing," European Actuarial Journal, vol. 7, no. 1, pp. 3–25, 2017.

[3] Ranjbari, G. Tafidis, L. Pilla, and J. Andreasson, "Ridesourcing in the global context: A systematic literature review," Transportation Research Part D, vol. 75, pp. 181–200, 2019.

[4] H. Wang, F. Tang, Y.-C. Hu, X. Wang, and S. Yokoo, "Improving pedestrian attribute recognition with multi-scale spatial tokenization," in Proc. ACM SIGKDD, 2019.

[5] P. Baecke and L. Bocca, "The value of vehicle telematics data in insurance risk selection processes," Decision Support Systems, vol. 98, pp. 69–79, 2017.

[6] M. Ayuso, M. Guillen, and A. M. Perez-Marin, "Using GPS data to analyse the distance traveled to the first accident at fault in pay-as-you-drive insurance," Transportation Research Part C, vol. 68, pp. 160–167, 2016.

[7] Chollet, Deep Learning with Python. Shelter Island, NY, USA: Manning Publications, 2017.

[8] Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2019.