Intelligent Car Recommendation System: A Machine Learning Approach

Uzair Ahmad(22P-9021), Arsalan Mateen(22P-9024), Hamza Tahir(22P-9065),
Muhammad Sohaib Jehangir(22P-9035), Saad Kabeer(22P-9032), Azhan shoaib(22P-9054),
Talha Hanif(22P-9051), Ahmad Naeem(22P-9030), Muhammed Usman(22P-9071), Saad Ahmed(22P-9066),
Khubaib Qamar(22P-9033), Hina Rashid(22P-9198), Lariab Fatima(22P-0503),
Mazhar Saeed(22P-9034), Muhammad Ahmad(22P-9069)
National University of Computer & Emerging Sciences - FAST
Peshawar, Pakistan

***Abstract*—With the rapid expansion of the automobile industry, consumers face an overwhelming number of choices when purchasing a car. Machine learning (ML) offers an innovative solution by analyzing user preferences, past interactions, and car attributes to provide personalized vehicle recommendations. This paper explores the application of ML in building intelligent car recommendation systems, addressing challenges such as data quality, the cold-start problem, model transparency, and privacy concerns. By refining data processing and balancing interpretability, these systems can enhance the car-buying experience.**

***Index Terms*—Machine Learning, Recommendation Systems, Automobile Industry, Personalization, Data Quality**

I. Introduction

With the rapid expansion of the automobile industry, consumers now face an overwhelming number of choices when purchasing a car. From fuel type and engine capacity to budget constraints and personal preferences, making an informed decision can be complex and time-consuming. While online platforms offer basic filtering options, they often fail to provide personalized recommendations.

Machine learning (ML) presents an innovative solution to this challenge. Unlike traditional rule-based filtering, ML models can analyze user preferences, past interactions, and car attributes to suggest the most relevant vehicles [2]. These systems adapt to trends, identify patterns in data, and improve over time, making them a valuable tool for car buyers.

A. The Role of Machine Learning

ML-based recommendation systems are widely used in e-commerce, entertainment, and social media [3]. Applying similar techniques to car selection enhances user experience by providing tailored suggestions. These models can predict price ranges, classify user preferences, and refine recommendations based on real-world data. However, the accuracy of such systems depends on data quality, preprocessing, and feature engineering [1].

B. Challenges and Considerations

Building a reliable recommendation system involves overcoming several challenges:

- **Data Quality**: Incomplete, inconsistent, or noisy data affects model performance, requiring preprocessing techniques like normalization and outlier removal [1].
- **Cold-Start Problem**: New users or newly listed cars may lack historical data, making recommendations less accurate [3].
- **Model Transparency**: Some advanced ML models work as “black boxes,” making it difficult to explain why a specific car is recommended.
- **User Preferences**: Factors like brand loyalty, aesthetics, and driving habits are subjective and difficult to quantify.
- **Privacy Concerns**: Collecting and utilizing user data raises ethical considerations regarding data security and fairness.

II. Problem Statement

The automobile industry has grown rapidly in recent years due to advancements in technology, increasing consumer demand, and globalization. Car manufacturers now provide a vast range of models with different specifications, catering to diverse preferences and budgets. While this expansion benefits consumers by offering more choices, it also makes selecting the right vehicle a complex and time-consuming task.

The challenge is not just the large number of available options but also the various factors buyers must consider before making a purchase. These factors typically include price, fuel economy, engine power, mileage, brand reputation, maintenance costs, resale value, and even location-specific conditions such as fuel availability or road conditions. For example, a buyer in a city looking for a fuel-efficient car with low maintenance will have different needs than someone searching for an off-road SUV for rough terrains. Sorting through these aspects manually or through conventional browsing methods can be overwhelming and inefficient.

To simplify the car-buying process, online platforms like PakWheels, OLX, and Cars.com have emerged, offering filtering options based on brand, price, fuel type, and condition. However, these filters are mostly static and lack the ability to learn from user preferences, adjust recommendations, or adapt based on evolving choices. Research suggests that traditional recommendation methods lack personalization and often fail to align well with user preferences [4].

This gap highlights a key issue—while vast amounts of data are available on car marketplaces, there is no intelligent system capable of utilizing this data for personalized recommendations. Users often face frustration due to the sheer number of similar listings, inconsistent information, and limited customization. As a result, they rely on manual comparisons or external opinions, which can introduce bias and slow down the decision-making process.

Additionally, the lack of personalized recommendations may lead to poor choices. A buyer might overlook a car that perfectly matches their requirements simply because it did not meet the rigid filtering criteria. Conversely, they might receive suggestions that only partially match their needs, such as recommending a sports car to a family-focused buyer concerned with safety and fuel economy.

Another issue is information overload. Online platforms feature thousands of listings with minor variations in price or features, making it difficult for users to determine which offers the best value. The mental effort required to compare multiple vehicles based on different attributes can lead to decision fatigue, where users either delay their purchase or make a hasty decision without proper evaluation.

Furthermore, many buyers—especially first-time car owners—lack technical knowledge about vehicle specifications. They might not fully understand how engine size impacts performance or how different fuel types affect long-term expenses. A recommendation system that simplifies these details based on user inputs could improve accessibility and overall satisfaction.

The core challenge lies in the gap between the availability of data and its intelligent interpretation. There is a strong need for a smart system that can analyze user preferences, interpret vehicle data meaningfully, and generate accurate, personalized recommendations. This issue is particularly relevant in markets like Pakistan, where the used car market is large, diverse, and less structured compared to more developed economies.

To address this problem, this study aims to leverage machine learning to develop an intelligent recommendation system. By bridging the gap between complex car data and user needs, such a system could enhance the buying experience, reduce the time required to find a suitable car, and ultimately improve decision-making efficiency.

III. Motivation for Using Machine Learning

The automotive market has grown increasingly complex, with a vast range of vehicle offerings making traditional recommendation techniques inadequate for guiding consumer choices effectively. As customer expectations evolve, the demand for intelligent systems capable of delivering personalized and accurate car recommendations has intensified. Machine Learning (ML), a subfield of Artificial Intelligence (AI), presents a compelling solution by leveraging past data, uncovering hidden patterns, and making informed predictions. Given its ability to adapt and refine recommendations dynamically, ML is particularly suited for applications in the used car market.

Traditional rule-based recommendation systems depend on predefined logic and static filters that lack adaptability to individual preferences or market trends. While useful in simple scenarios, they struggle as the number of features grows and consumer preferences become more nuanced. For instance, a buyer may seek a vehicle that balances performance and fuel efficiency while also considering factors such as resale value, brand reliability, and maintenance costs. Rule-based systems fail to capture and process such multifaceted preferences, whereas ML-based approaches excel by identifying complex, nonlinear relationships among variables and continuously refining recommendations as new data becomes available.

A key strength of ML lies in its ability to utilize both explicit and implicit data. Explicit data includes user-specified preferences like car brand, price range, fuel type, and transmission. Implicit data, on the other hand, is derived from user interactions, such as browsing history, search behavior, and engagement with specific listings. Research highlights that recommendation systems integrating both data types achieve superior personalization and prediction accuracy [5]. By adopting this hybrid approach, ML models can develop comprehensive user profiles, capturing not just stated preferences but also inferred interests.

Another significant advantage of ML in car recommendation systems is its capability to address both classification and regression problems. Classification techniques can categorize vehicles into segments such as economy, mid-range, luxury, or sports based on user preferences and budget. Meanwhile, regression models excel at predicting continuous variables like car price or estimated mileage based on attributes such as engine capacity, year of manufacture, and condition. These capabilities enable ML-powered recommendation systems to generate highly tailored and insightful suggestions.

Supervised learning techniques such as Decision Trees, Random Forests, Logistic Regression, and Support Vector Machines (SVM) are particularly effective when trained on labeled datasets, such as the PakWheels Used Car Dataset (2022). These models can predict key outcomes, such as whether a vehicle meets a buyer's requirements or its expected market value, by analyzing historical listings. Additionally, unsupervised learning methods like K-Means clustering can group similar cars or user profiles, enabling recommendations based on shared preferences.

Scalability is another crucial advantage of ML-driven recommendation systems. As datasets grow in volume and complexity, ML models can continue to adapt and maintain high performance. This adaptability is essential in the used car market, where listings, pricing trends, and user behaviors evolve rapidly. Unlike static rule-based systems that require frequent manual updates, ML models can be retrained with fresh data to ensure continued accuracy [6].

Transparency and interpretability are additional benefits of ML models. For example, a Random Forest model can assign feature importance scores to variables like engine capacity, vehicle age, and fuel type, highlighting their influence on recommendations or price predictions. This enhances accountability and helps users understand why a particular suggestion was made.

In conclusion, machine learning provides a robust, adaptable, and intelligent framework for building car recommendation systems. Its ability to process diverse data sources, identify hidden patterns, learn from historical trends, and adapt to new information makes it significantly more effective than traditional methods. As the automotive industry continues to become more data-driven and consumer expectations evolve, ML-powered recommendation systems will play a crucial role in enhancing user experiences, supporting informed decision-making, and driving innovation in the sector.

IV. Requirements

Developing a machine learning-based car recommendation system involves a mix of software, hardware, and data resources. These components work together to support data collection, model training, evaluation, and deployment. A well-structured setup ensures efficiency, scalability, and reproducibility throughout the process.

A. Software Requirements

Python is the primary language for this system due to its simplicity and strong ecosystem for data science and machine learning. Key libraries include:

- Pandas for data manipulation using DataFrames
- NumPy for handling arrays and scientific computation
- Scikit-learn for a wide range of ML algorithms and tools
- Matplotlib and Seaborn for data visualization
- Jupyter Notebook for interactive development and documentation

For more advanced models, libraries like XGBoost, TensorFlow, or Keras can be used. Tools such as GridSearchCV, Cross_val_score, and SHAP support model tuning and explainability.

B. Hardware Requirements

The system uses the PakWheels Used Car Dataset (October 2022), and a standard modern setup is sufficient:

- CPU: Multi-core (e.g., Intel i5/i7 or AMD Ryzen 5+)
- RAM: Minimum 8GB; 16GB+ recommended for heavier workloads
- Storage: 256GB+ SSD for faster access
- GPU: Optional, useful for deep learning but not required for this project

Cloud platforms like Google Colab, AWS, or Azure can be used for heavier tasks or collaborative development.

C. Data Requirements

The dataset includes car listings with features like make, model, year, price, mileage, engine capacity, and more. A good dataset should be diverse, complete, accurate, and recent. Preprocessing steps include handling missing values, encoding categorical variables, normalizing numerical features, outlier removal (e.g., using IQR or Z-score), and feature engineering (e.g., calculating car age).

D. Tools and Infrastructure

Version control is managed via Git and GitHub for collaboration. Additional tools include:

- Virtual environments (venv or conda) to manage dependencies
- Notebook tools (nbconvert, JupyterHub) for versioning
- Model persistence with joblib or pickle for saving trained models

V. Dataset Overview

A good dataset is crucial for any machine learning project, especially for recommendation systems where accuracy is key. In this research, we used the PakWheels Used Car Dataset (October 2022), available on Kaggle [7], [8]. This dataset is based on real-world car listings from Pakistan's leading automobile marketplace, PakWheels. Since it consists of actual consumer listings, it provides a diverse and realistic collection of used car advertisements across different cities and price ranges in Pakistan.

A. Dataset Description and Features

The dataset contains thousands of car listings, with each row representing a unique used car. The key details included in the dataset are:

- Make (e.g., Toyota, Honda, Suzuki)
- Model (e.g., Corolla, Civic, Cultus)
- Year (Year of manufacture)
- Price (Listed price in PKR)

- Mileage (Total kilometers traveled)
- Engine Capacity (e.g., 1000cc, 1300cc)
- Transmission (Manual or Automatic)
- Fuel Type (Petrol, Diesel, Hybrid, CNG)
- Condition (Used, New, Slightly Used, etc.)
- Location (City where the car is listed)
- Registration City
- Other Features (Color, Body Type, ABS, Navigation, Alloy Wheels, etc.)

This dataset contains a mix of numerical (e.g., price, mileage) and categorical (e.g., make, fuel type) data, making it suitable for both classification and regression models.

Unlike synthetic datasets like the UCI Car Evaluation Dataset, which is based on hypothetical ratings, the PakWheels dataset is real-world data. This makes it more complex and challenging but also more valuable for building practical machine learning models.

B. *Why This Dataset?*

This dataset is a great choice for our research for several reasons:

1) **Diverse Listings**: It covers cars of different brands, prices, years, and conditions, making it suitable for various buyers (from economy cars to luxury SUVs).
2) **Real-World Pricing**: Since prices come from real user listings, they reflect actual market trends instead of fixed dealership rates.
3) **Localized Data**: The dataset is Pakistan-specific, making it useful for analyzing regional car demand and trends.
4) **Comprehensive Features**: The dataset includes important car details that affect pricing and desirability.

C. *Challenges in the Dataset*

Despite its advantages, the dataset comes with typical real-world problems:

- **Missing Data**: Some listings lack details like mileage or transmission type.
- **Inconsistent Formatting**: Variations like “Petrol” vs. “petrol” or “1300cc” vs. “1.3” need standardization.
- **Outliers**: Some prices and mileage values are unrealistically high or low.
- **Duplicate or Overlapping Entries**: Similar cars might be listed under slightly different names, requiring category merging.

To handle these issues, we applied data cleaning, normalization, outlier detection, and feature encoding before training our models. Data quality is as important as model accuracy, so preprocessing is a key step in this research.

D. *Ethical Considerations*

The dataset is publicly available on Kaggle for academic purposes, and we ensured that no personal user information (e.g., seller contacts) was used or stored. The research was conducted ethically and strictly for educational purposes.

E. *Data Splitting for Model Training*

For training and testing machine learning models, we split the dataset into 80% training and 20% testing. This ensures that models are evaluated on unseen data for better accuracy assessment. Additionally, k-fold cross-validation was used to improve model robustness and reduce the impact of random splits.

F. *Conclusion*

The PakWheels Used Car Dataset is a strong foundation for developing a>’;

System: car recommendation system. It is diverse, real-world relevant, and rich in features. While it presents some challenges (like missing data and inconsistencies), these issues also provide an opportunity to develop better preprocessing techniques, making the research more practical and applicable to real-world machine learning systems.

VI. Data Preprocessing

Data preprocessing is crucial for developing a car recommendation system using the PakWheels Used Car Dataset (October 2022), as real-world datasets often contain missing values, outliers, and inconsistencies. Without addressing these issues, machine learning models may produce unreliable results. The following subsections detail the preprocessing steps applied to ensure a clean and robust dataset.

A. *Data Cleaning*

The PakWheels dataset contained incomplete car listings, such as missing mileage, engine capacity, or fuel type. To address this, we employed two strategies:

- **Deletion**: Listings missing critical attributes, such as price or mileage, were removed from the dataset.
- **Imputation**: For minor gaps, such as missing color or city, we imputed values using the median for numerical features (e.g., mileage) and the mode for categorical features (e.g., fuel type).

Additionally, duplicate listings, identified by matching attributes like make, model, year, and mileage, were removed to ensure each car appeared only once.

B. *Data Transformation and Encoding*

Machine learning models require numerical inputs, so categorical features were transformed as follows:

- **Label Encoding**: Ordered categories, such as car condition (e.g., New > Slightly Used > Used), were encoded as integers (e.g., 1, 2, 3).
- **One-Hot Encoding**: Unordered categories, such as make or fuel type, were converted into binary columns. For example, fuel type was split into four columns (Petrol, Diesel, Hybrid, CNG), with

a 1 indicating the presence of a category and 0s elsewhere.

This transformation converted textual data into a numerical format suitable for model training.

C. Feature Engineering

To enhance the dataset's utility, we created two new features:

- **Car Age**: Calculated by subtracting the car's manufacturing year from the current year (2025). For example, a 2015 car is assigned an age of 10 years.
- **Cost per Kilometer**: Computed by dividing the car's price by its mileage, providing a metric for value per kilometer driven.

These features provide additional context for model predictions and recommendation relevance.

D. Detection and Removal of Outliers

The dataset contained unrealistic values, such as cars priced at 10 PKR or with a million kilometers of mileage. Outliers were addressed using:

- **IQR Method**: Values exceeding 1.5 times the interquartile range for price or mileage were removed.
- **Z-score Check**: Data points more than three standard deviations from the mean were flagged and excluded.

These methods ensured the dataset reflected realistic market conditions.

E. Scaling and Normalization

To ensure compatibility with distance-based or gradient-based algorithms, numerical features were standardized:

- **Min-Max Normalization**: Features like price, mileage, and engine capacity were scaled to a [0, 1] range.
- **Standardization**: Features were transformed to have a mean of 0 and a standard deviation of 1, where appropriate.

The choice of scaling depended on the requirements of each model.

F. Final Dataset Overview

After preprocessing, the PakWheels dataset was significantly improved:

- All missing values were either imputed or removed.
- Outliers were eliminated to ensure data reliability.
- Categorical features were encoded, and numerical features were scaled.
- New features were added to enhance model performance.

The cleaned dataset was split into 80% training and 20% testing sets to facilitate model training and evaluation. This rigorous preprocessing ensures that the dataset is well-suited for building a reliable car recommendation system.

VII. Model Selection

Selecting the right model is crucial in machine learning, as it affects accuracy, usability, and efficiency. For a car recommendation system, two main approaches can be used—regression and classification—depending on the goal.

A. Regression vs. Classification

Regression is used when predicting a continuous value like car price. Factors such as engine capacity, mileage, model year, and fuel type help estimate market value, aiding users in finding fair pricing and budget-friendly options. Conversely, classification is employed when recommending car categories such as "family," "luxury," or "fuel-efficient." This approach assists users who are unsure about specific models but have general preferences. Both methods were tested to create a comprehensive recommendation system.

B. Algorithms Used for Model Training

Various machine learning models were evaluated to identify the best fit for the system:

- **Decision Trees and Random Forests**: Decision trees are simple but prone to overfitting, while Random Forests mitigate this by combining multiple trees. They performed well for both regression and classification tasks.
- **K-Nearest Neighbors (KNN)**: A simple model that predicts based on nearby data points. While effective for small datasets, it was computationally expensive and less practical for large-scale use.
- **Logistic Regression and Support Vector Machines (SVM)**: Logistic regression is fast and interpretable but assumes linear relationships. SVM handles complex patterns better but requires careful tuning and is computationally intensive.
- **XGBoost**: A powerful boosting algorithm that builds decision trees sequentially. It was one of the best-performing models, balancing accuracy, efficiency, and flexibility.

C. Model Comparison and Evaluation

To assess performance:

- Regression models were evaluated using Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R^2 Score.
- Classification models were measured by Accuracy, Precision, Recall, and F1-Score.

Models were trained on 80% of the dataset and tested on 20%, with k-fold cross-validation to ensure unbiased results. Random Forest and XGBoost performed best across both tasks, offering high accuracy and reliability.

D. Conclusion

Choosing the right model requires balancing accuracy, speed, and interpretability. In this study, Random Forest and XGBoost proved to be the most effective for both regression and classification, making them ideal choices for an intelligent car recommendation system.

VIII. Model Training Process

After selecting the machine learning models and cleaning the dataset, the next step was training the models. This is key to building a reliable car recommendation system since it enables the algorithms to recognize patterns from historical data and make accurate predictions for new cases. This section outlines the data splitting strategy, model training process, and techniques used for cross-validation and hyperparameter tuning.

A. Data Splitting Strategy

To evaluate model performance, the dataset was divided into training and testing sets using an 80:20 split, where 80% of the data was used for training and 20% for testing. This standard practice in supervised learning assesses the model's generalization to unseen data and helps prevent overfitting. Additionally, K-Fold Cross-Validation was employed, dividing the dataset into k equal folds. The model was trained on $k - 1$ folds and tested on the remaining fold, repeating this process k times to use each fold as a test set once. The results were averaged to provide a robust performance estimate. In this study, 5-fold cross-validation was used to balance performance and computational efficiency.

B. Training the Models

The preprocessed dataset was used to train multiple machine learning models, including Random Forest, XGBoost, Linear Regression, Logistic Regression, and Support Vector Machine (SVM). Each model was trained on the training set and evaluated using cross-validation and the test set. The training process varied depending on whether the task was regression (predicting car prices) or classification (categorizing car types).

For regression tasks:

- Models analyzed features such as car brand, engine capacity, year, and mileage to predict a continuous value (price).
- Loss functions like Mean Squared Error (MSE) and Mean Absolute Error (MAE) were used to measure accuracy.
- The objective was to minimize prediction errors while ensuring generalization to new data.

For classification tasks:

- Models classified cars into categories such as budget, family, or luxury based on their features.
- The output was a specific category label.
- Performance was assessed using accuracy, precision, recall, and F1-score.

Tree-based models like Random Forest and XGBoost operated by recursively splitting the data based on feature importance, minimizing impurity at each step.

C. Avoiding Overfitting and Underfitting

Overfitting occurs when a model performs well on training data but poorly on new data, while underfitting results from a model being too simple to capture data patterns. To address these issues:

- Cross-validation ensured the model did not memorize the training data.
- Regularization techniques, such as L1 and L2, were applied to Logistic Regression and SVM to prevent excessive complexity.
- Early stopping was used in XGBoost to halt training when performance on validation data ceased to improve.

D. Hyperparameter Tuning

Hyperparameters, such as tree depth in Random Forest or learning rate in XGBoost, significantly impact model performance and must be set prior to training. Two methods were used for tuning:

- **Grid Search**: Exhaustively tested all possible hyperparameter combinations within a specified range.
- **Randomized Search**: Tested a random subset of hyperparameter combinations, improving efficiency for large search spaces.

For Random Forest, hyperparameters like `n_estimators`, `max_depth`, and `min_samples_split` were optimized. For XGBoost, `learning_rate`, `max_depth`, `n_estimators`, and `subsample` were tuned. Scikit-learn's `GridSearchCV` and `RandomizedSearchCV` were used to automate this process, with cross-validation to prevent data leakage. The best hyperparameters were used to retrain the models on the full training set, followed by evaluation on the test set.

E. Summary of the Training Process

The model training process involved:

1) Splitting the preprocessed data into training and test sets.
2) Training multiple models with cross-validation to assess performance.
3) Optimizing hyperparameters using grid and randomized search.
4) Applying regularization and early stopping to prevent overfitting.
5) Testing the final models on unseen data.

This structured approach ensured the car recommendation system was accurate, scalable, and capable of reliable predictions. The trained models supported two primary tasks: predicting car prices and recommending car types, forming the core of the machine learning system.

IX. Model Testing and Evaluation

Model evaluation is a critical stage in the machine learning pipeline, ensuring reliability and applicability in real-world scenarios. This study evaluated models for two tasks: regression for predicting used car prices in PKR and classification for recommending car categories based on user preferences.

A. Regression Evaluation Metrics

Regression models aimed to predict continuous values, specifically used car prices. The following metrics were used:

1) Mean Absolute Error (MAE): MAE measures the average magnitude of prediction errors in the same unit as the target variable (PKR). Lower MAE indicates better performance. Random Forest and XGBoost models exhibited significantly lower MAE compared to Linear Regression.

2) Root Mean Squared Error (RMSE): RMSE squares errors before averaging, making it more sensitive to large errors, which is critical for high-value predictions like luxury vehicles. Random Forest and XGBoost achieved lower RMSE scores than baseline models.

3) R^2 Score (Coefficient of Determination): The R^2 score indicates how well the model explains the variance in the target variable, with higher values (closer to 1) indicating better fit. Random Forest achieved an R^2 of 0.87, while Linear Regression scored approximately 0.65, reflecting its inability to capture complex patterns.

B. Metrics for Classification Evaluation

Classification models predicted car categories such as "budget," "family," or "luxury." The following metrics were utilized:

1) Accuracy: Accuracy measures the proportion of correct predictions. However, it can be misleading with imbalanced data, as the dataset contained more "budget" and "family" cars than "luxury" ones.

2) Precision and Recall: Precision measures the proportion of correct positive predictions, while recall measures the proportion of actual positives detected. High precision is crucial when false positives, such as recommending an expensive car inappropriately, are costly.

3) F1-Score: The F1-Score, the harmonic mean of precision and recall, is effective for imbalanced datasets. Macro-averaged F1 scores were used to assess overall performance across car classes.

4) Confusion Matrix: Confusion matrices revealed misclassification patterns, with most errors occurring between "budget" and "family" cars due to overlapping features. This insight guided further feature engineering.

C. Performance Comparison

The performance of the models is summarized in Table I.

TABLE I
Model Performance Comparison

Model **F1-Score**	**Task**	**MAE (PKR)**	**RMSE (PKR)**	R^2 **Score**
Linear Regression –	Regression	268,000	437,000	0.65
Random Forest –	Regression	134,000	212,000	0.87
XGBoost –	Regression	128,000	205,000	0.89
Logistic Regression –	Classification	–	–	–
Random Forest 0.71	Classification	–	–	–
XGBoost 0.82	Classification	–	–	–

XGBoost outperformed other models in both tasks, leveraging its ability to model complex patterns and internal regularization. Logistic Regression metrics were not reported in the evaluation table.

D. Error Analysis and Observations

Error analysis revealed several issues:

- **Overfitting**: Observed in unpruned Decision Trees, leading to poor generalization.
- **Underfitting**: Linear Regression failed to capture nonlinear trends.
- **Class Imbalance**: Caused frequent misclassification of "luxury" cars.
- **Outliers**: Price data outliers, addressed through filtering, impacted initial model performance.

Visualization tools, such as residual plots and confusion matrices, facilitated these findings and informed model tuning.

E. Summary of Evaluation Phase

The evaluation phase confirmed the efficacy of ensemble methods like Random Forest and XGBoost for both regression and classification tasks, supported by robust cross-validation and metric comparisons. However, limitations such as class imbalance and noisy inputs were identified, suggesting future improvements like applying Synthetic Minority Oversampling Technique (SMOTE) or advanced feature engineering. These findings lay a strong foundation for further enhancement and practical deployment of the car recommendation system.

X. Results and Interpretation

The results from the trained machine learning models provide valuable insights into the accuracy and reliability of the car recommendation system. This section presents the outcomes of both regression and classification models, interprets the significance of features, and analyzes model behavior using statistical metrics and visualizations. Beyond evaluating performance metrics, this analysis explores feature relationships, identifies key drivers of predictions, and highlights areas for improvement to enhance real-world applicability.

A. Regression Results: Car Price Prediction

For the regression task, models were trained to predict used car prices based on features such as engine capacity, brand, year, transmission type, fuel type, and mileage. XGBoost and Random Forest outperformed other models, achieving the lowest errors and highest R^2 scores.

1) Quantitative Results: The performance of regression models is summarized in Table II.

TABLE II
REGRESSION MODEL PERFORMANCE FOR CAR PRICE PREDICTION

Model	MAE (PKR)	RMSE (PKR)	R^2 Score
Linear Regression	268,000	437,000	0.65
Random Forest	134,000	212,000	0.87
XGBoost	128,000	205,000	0.89

These results demonstrate that ensemble methods, such as XGBoost and Random Forest, significantly outperform Linear Regression due to the nonlinear relationships in car price data. XGBoost excelled in predicting prices for mid-range and budget cars.

2) Residual Analysis: Residual analysis, examining the difference between predicted and actual prices, showed that most predictions were closely aligned with actual values, with errors centered around zero. However, the models struggled with luxury and rare cars, likely due to factors such as brand prestige, custom modifications, or seller motivations, which are difficult to quantify.

B. Classification Results: Car Category Recommendation

For classification, models categorized cars into types such as Budget, Family, or Luxury based on features like price, fuel efficiency, and engine power. XGBoost achieved the highest performance, followed by Random Forest.

1) Confusion Matrix: The confusion matrix for XGBoost, shown in Table III, highlights classification performance:

TABLE III
CONFUSION MATRIX FOR XGBOOST CLASSIFICATION

Actual	Predicted	Budget	Family	Luxury
Budget		402	23	5
Family		18	345	37
Luxury		3	28	196

The matrix indicates:

- High accuracy for Budget and Family cars.
- Some confusion between Family and Luxury cars, likely due to overlapping features like large engines in some family cars.
- Accurate classification of Sports and Off-road cars due to their distinct features.

Overall accuracy was 84%, with an F1-score of 0.82, indicating balanced performance across categories.

2) Classification Report: The classification report provided the following metrics:

- Budget cars: Precision = 0.91
- Family cars: Recall = 0.88
- Luxury cars: F1-score = 0.81

The model performed well on common categories but showed room for improvement on rare categories. Techniques like Synthetic Minority Oversampling Technique (SMOTE) could enhance performance by addressing class imbalance.

C. Feature Importance Analysis

Feature importance analysis for XGBoost and Random Forest revealed key drivers of model decisions.

For price prediction (regression):

- Car Age (Year): 0.29
- Engine Capacity (cc): 0.21
- Mileage (km): 0.15
- Brand: 0.12
- Transmission and Fuel Type: Lower impact

These results align with market trends, where car age, engine power, and mileage significantly influence price, while brand reputation has a moderate effect.

For classification:

- Price and Engine Capacity were the primary drivers.
- Family cars: Moderate engine size and price.
- Luxury cars: High price, large engines, often automatic transmission.

D. Visual Interpretation and Insights

Visualizations provided deeper insights:

- Scatter plots showed XGBoost's predictions closely matched actual prices for mid-range cars.
- Feature importance bar charts confirmed the dominance of engine capacity and car age.
- Correlation heatmaps revealed that price increases with engine capacity but decreases with mileage.

These visualizations confirm that the models' predictions are both accurate and logically consistent with real-world car market dynamics.

E. Strengths and Limitations

Strengths:

- High accuracy on real-world car data.
- Interpretability through feature importance and visualizations.
- Robust performance across diverse car types.

Limitations:

- Reduced accuracy for rare or imported cars.

- Sensitivity to outlier listings.
- Inability to account for subjective preferences (e.g., color or style).

F. Summary

XGBoost and Random Forest demonstrated strong performance in predicting car prices and recommending car categories, with high accuracy and interpretability. The models are well-suited for real-world deployment but could be improved by:

- Incorporating user behavior data (e.g., click patterns).
- Addressing class imbalance for rare car categories.
- Exploring image recognition for assessing car condition.

This system provides a robust foundation for a car recommendation system, with clear pathways for further enhancement.

XI. Challenges in Developing a Machine Learning-Based Car Recommendation System

Building an intelligent car recommendation system using machine learning involves multiple stages, from data collection to model deployment. While this study demonstrated the feasibility of such a system, several challenges were encountered, requiring strategic solutions to ensure accuracy, reliability, and usability.

A. Handling Noisy and Incomplete Data

Real-world datasets often contain missing, inconsistent, or inaccurate data. Issues such as omitted details, formatting inconsistencies, and outliers (e.g., unrealistic mileage or pricing) necessitated extensive preprocessing. Techniques like median imputation, statistical filtering, and text normalization were employed to refine the dataset.

B. Overcoming the Cold-Start Problem

Newly listed cars or first-time users lack sufficient historical data, complicating accurate predictions. Content-based filtering was used to mitigate this issue, but incorporating hybrid models with collaborative filtering could further improve recommendations for new users or unique vehicles [5].

C. Ensuring Model Interpretability

Ensemble models like XGBoost and Random Forest, while accurate, function as “black-box” algorithms, making it challenging to explain predictions. Techniques such as SHAP (SHapley Additive exPlanations) values and feature importance rankings were implemented to enhance transparency, though these added computational complexity.

D. Addressing Class Imbalance

The dataset was skewed, with most cars labeled as “budget” or “family” and fewer as “luxury” or “sports.” This imbalance led to biased predictions for underrepresented categories. Methods like Synthetic Minority Oversampling Technique (SMOTE), adjusted class weights, and alternative evaluation metrics were used to improve model balance.

E. Managing Scalability and Computation Time

As the dataset size increased, training times for complex models grew significantly. Optimization strategies, including early stopping, parallel processing, and efficient hyperparameter tuning, were applied. Further improvements, such as model pruning, could enhance scalability.

F. Capturing Subjective User Preferences

Preferences like brand loyalty, design appeal, or transmission type are inherently subjective and difficult to quantify. Future models could leverage sentiment analysis, behavioral tracking, or personalized feedback to better capture these preferences, while addressing associated ethical considerations.

G. Addressing Ethical and Privacy Concerns

Using personal data for recommendations raises ethical issues, including data security, user consent, and potential biases in recommendations. Implementing transparent data practices and responsible AI policies is critical to ensure fairness and maintain user trust.

H. Conclusion

Developing a robust recommendation system requires addressing data quality, improving model generalization, balancing interpretability with accuracy, and ensuring ethical AI practices. While this study successfully tackled many of these challenges, further research and refinement are necessary for effective real-world deployment.

XII. Future Development: Improving Machine Learning-Driven Car Recommendation Systems

The current machine learning-driven car recommendation system demonstrates significant potential for predicting car prices and suggesting vehicle categories based on structured attributes. However, there is considerable scope for enhancement to improve accuracy, personalization, scalability, and user experience. The following key areas outline potential improvements for future development.

A. Incorporating Real-Time Customer Feedback

To enhance responsiveness, integrating real-time user feedback is essential. The current system relies on static data, limiting its ability to adapt to individual preferences dynamically. By analyzing user behavior, such as clicks, favorites, and watchlists, the model can refine recommendations in real time. Employing reinforcement learning techniques will enable the system to adapt to positive or negative feedback, similar to platforms like Netflix and Amazon.

B. Addition of User Reviews and Sentiment Analysis

User-generated content, such as reviews and forum discussions, provides valuable insights into subjective attributes like comfort and reliability. Natural Language Processing (NLP) techniques, such as sentiment analysis using VADER or BERT, can capture opinions about specific car models. Combining structured data with unstructured user sentiment will improve recommendation accuracy and increase user trust.

C. Foray into Deep Learning and Neural Collaborative Filtering

Traditional machine learning algorithms like Random Forest and XGBoost perform well but struggle to capture complex user preferences. Deep learning approaches, such as Neural Collaborative Filtering (NCF), can enhance personalization by modeling latent user-item interactions. These models leverage historical behavior and multidimensional data to deliver more sophisticated recommendations [5].

D. Deployment as a Web Application

Transitioning from a research prototype to a consumer-facing application requires deploying the system as a web application. A front-end built with React or Vue.js, paired with a back-end using Flask or Django, can enable real-time user interaction. Hosting on cloud platforms like AWS or Google Cloud ensures scalability. Incorporating a feedback loop will further refine future recommendations.

E. Personalization Using User Profiles and Context Awareness

The current system does not track user preferences over time. Implementing persistent user profiles will enable more tailored recommendations. Features like saving previous searches, learning user trade-offs (e.g., fuel economy versus engine power), and providing context-aware suggestions (e.g., recommending SUVs during winter) will enhance user engagement and satisfaction.

F. Enhanced Data Collection and Real-Time Market Updates

Augmenting the data pipeline with real-time information from sources like PakWheels, OLX, and manufacturer websites will ensure recommendations reflect current market trends. Integrating APIs for car history, fuel prices, and maintenance costs will provide additional context for decision-making.

G. Conclusion

Implementing these enhancements will transform the car recommendation system into a highly personalized, real-time, and scalable solution. Key improvements include:

- Integration of real-time user feedback
- Sentiment analysis of user-generated content
- Adoption of deep learning recommendation models
- Web-based deployment for accessibility
- Context-aware, personalized user profiles
- Dynamic market data collection and updates

These advancements will evolve the system from a static research model into an intelligent, user-friendly platform, revolutionizing how consumers navigate and select cars in a dynamic market.

XIII. Conclusion

The digital transformation of commerce, particularly in the automotive sector, has significantly altered consumer decision-making processes. With a multitude of options available online, navigating car listings efficiently can be overwhelming for buyers. This study aimed to address this challenge by developing and evaluating a machine learning-driven car recommendation system designed to assist users in selecting vehicles based on their budget, preferences, and needs.

By employing supervised learning techniques on real-world automotive data, specifically the PakWheels Used Car Dataset (October 2022), this research demonstrated that machine learning models can effectively predict car prices and classify vehicles into categories such as budget-friendly, family-oriented, and luxury models. The foundation of this system relied on transforming raw data into actionable insights through extensive preprocessing, feature engineering, and model training using regression and classification algorithms.

A. Summary of Findings

The study found that ensemble learning models, particularly Random Forest and XGBoost, performed significantly better than linear regression and simpler classification models. These algorithms proved highly effective in capturing complex data patterns, handling diverse feature types, and delivering high accuracy while maintaining interpretability through visualization and feature importance analysis.

For price prediction, the XGBoost Regressor achieved an R^2 score of 0.89, demonstrating a strong ability to estimate car prices based on parameters such as engine capacity, mileage, brand, and manufacturing year. This model can be instrumental in helping buyers assess whether a vehicle is fairly priced on used car marketplaces.

In the classification task, the XGBoost Classifier attained an accuracy of 84% and a macro F1-score of 0.82, ensuring balanced performance across various car categories. The classification results were validated using confusion matrices and precision-recall metrics, confirming the model's robustness in recommending cars aligned with user preferences.

Furthermore, the research highlighted the importance of data preprocessing steps such as handling missing values, removing outliers, and encoding categorical variables. These processes played a crucial role in ensuring meaningful and reliable model outputs. Additionally, techniques like cross-validation and hyperparameter tuning enhanced the model's generalization capability, reducing the risks of overfitting and underfitting.

B. Implications of the Study

The successful implementation of a machine learning-based recommendation system in the automotive industry presents several key implications:

- **Enhanced Consumer Decision-Making**: Users can access data-driven insights that improve transparency, reducing reliance on dealers or agents who may provide biased or inaccurate information.
- **Optimized Market Dynamics**: Sellers can price their vehicles more competitively based on market trends, and platforms can enhance user experience by promoting relevant listings.
- **Foundation for Personalized Recommendations**: Future advancements can leverage this system to develop more personalized and adaptive recommendation engines that learn from user behavior over time.

This study aligns with the broader trend of integrating artificial intelligence into e-commerce, as seen in industries like retail, travel, and entertainment. Applying similar AI-driven strategies in the automotive sector represents a natural evolution in digital marketplaces [3].

C. Limitations and Considerations

Despite its effectiveness, the study identified several limitations that warrant further attention:

- **Data Quality and Diversity**: The dataset contained some inconsistencies and imbalances, particularly in underrepresented categories like luxury and sports cars.
- **Cold-Start Problem**: The system struggled with new or rare car listings, affecting its recommendation accuracy.
- **Subjectivity in Consumer Preferences**: Factors such as aesthetics, brand perception, and emotional attachment, which influence buyer decisions, are difficult to quantify using structured data.
- **Scalability and Real-Time Adaptability**: The current model operates on static data and batch learning. For practical deployment, it would require real-time learning capabilities and continuous updates.

These challenges do not diminish the study's contributions but rather highlight potential avenues for further research and development.

D. Recommendations for Future Research

To improve the system's capabilities and expand its scope, the following future enhancements are suggested:

- **Hybrid Recommendation Models**: Combining collaborative filtering with content-based approaches could improve personalization, particularly for new users or uncommon listings [5].
- **Sentiment Analysis**: Integrating natural language processing to analyze reviews and forum discussions could provide insights into subjective aspects like reliability and user satisfaction.
- **Real-Time User Feedback Integration**: Enabling the system to learn from user interactions (e.g., clicks, favorites, ratings) could enhance its accuracy over time.
- **Web-Based Deployment**: Developing an interactive web application using frameworks like Flask or Django could facilitate real-time recommendations and dynamic listing updates.
- **Context-Aware Recommendations**: Factoring in elements such as location, time, and environmental conditions could refine suggestions for users in different scenarios.

By implementing these improvements, the recommendation system could evolve into a more intelligent, scalable, and user-friendly platform, offering tailored and adaptive suggestions based on diverse consumer needs.

E. Concluding Remarks

This research validates the potential of machine learning in developing an effective recommendation system for used cars. By leveraging structured data, robust algorithms, and systematic evaluation techniques, the system exhibited strong performance in both predictive and classification tasks.

Machine learning has the power to transform the automotive marketplace, turning raw data into actionable insights that enhance user decision-making. As online vehicle purchasing becomes increasingly common, AI-driven tools like this recommendation system will play a crucial role in improving efficiency, transparency, and user satisfaction.

This study marks a significant step toward the advancement of intelligent, data-driven automotive platforms and lays the groundwork for future innovations in AI-powered recommendation systems.

References

[1] "Application of Hybrid Algorithms for Car Recommendation System," *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, vol. 10, no. 12, pp. 47784, Dec. 2022, doi: 10.22214/ijraset.2022.47784. [Online]. Available: https://www.ijraset.com/research-paper/application-of-hybrid-algorithms-for-car-recommendation-system

[2] "An App-Based Recommender System Based on Contrasting Automobiles," *Processes*, vol. 11, no. 3, p. 881, Mar. 2023, doi: 10.3390/pr11030881. [Online]. Available: https://www.mdpi.com/2227-9717/11/3/881

[3] "Artificial Intelligence in Recommender Systems," *Complex & Intelligent Systems*, Springer, Oct. 2020. [Online]. Available: https://link.springer.com/article/10.1007/s40747-020-00212-w

[4] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, Jun. 2005. [Online]. Available: https://ieeexplore.ieee.org/document/1423975

[5] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender Systems Handbook*, Springer, 2011, pp. 1–35. [Online]. Available: https://link.springer.com/chapter/10.1007/978-0-387-85820-3_1

[6] C. C. Aggarwal, "Recommender systems: The textbook," Springer, 2016. [Online]. Available: https://link.springer.com/book/10.1007/978-3-319-29659-3

[7] "Pakistan Car Reviews," Kaggle, 2022. [Online]. Available: https://www.kaggle.com/datasets/zusmani/pakistan-car-reviews/data

[8] "Pakistan's Automobiles Database by PakWheels," Kaggle, 2022. [Online]. Available: https://www.kaggle.com/code/yasirarfat/pakistan-s-automobiles-database-by-pakwheels/notebook