International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Car Price Prediction Model Using Linear Regression

Manish Khatri[1], Romi Gupta[2], Ms. Rimpy[3]

[1,2]Student, Dept. of Computer Applications, Maharaja Surajmal Institute, GGSIP University, New Delhi-58, India

[3]Assistant Professor, Dept. of Computer Applications, Maharaja Surajmal Institute, GGSIP University, New Delhi-58, India

E-mail: [1]manish05921202022@msijanakpuri.com, [2]romi09321202022@msijanakpuri.com, [3]rimpy@msijanakpuri.com

ABSTRACT

This paper presents the design and implementation of a Car Price Prediction Model using linear regression, aimed at accurately estimating the prices of used cars. The system leverages a dataset sourced from Kaggle, focusing on Honda car sales, and preprocesses data by handling outliers, encoding categorical variables, and normalizing numeric attributes. Linear regression was chosen for its simplicity and interpretability, allowing the system to predict prices based on features such as year of manufacture, mileage, and car model. The model was evaluated using k-fold cross-validation, achieving an accuracy of 82% with a mean cross-validation score of 83%. By offering a transparent and efficient solution, the system addresses the challenges of manual pricing and market inconsistencies, making it a valuable tool for buyers, sellers, and car dealerships. The proposed system demonstrates significant potential for integration with e-commerce platforms and other real-world applications.

Keywords: Car Price Prediction, Linear Regression, Machine Learning, Data Preprocessing, Feature Engineering.

1. Introduction

Accurate price estimation plays a crucial role in the rapidly expanding used car market, where inconsistent manual evaluations often lead to discrepancies and reduced market trust [1]. Existing methods rely heavily on subjective judgment or static pricing guidelines, which fail to adapt to evolving market conditions or individual car attributes. These limitations highlight the need for automated systems that leverage data-driven methodologies to deliver reliable predictions [2].

Machine learning techniques, such as linear regression, have shown considerable promise in solving price prediction challenges [3]. By analyzing historical sales data, these models identify patterns and relationships between car features—such as mileage, year of manufacture, and model—and their corresponding prices [4]. Unlike traditional approaches, machine learning models are capable of handling large datasets, making predictions more efficient and accurate.

This paper focuses on the development of a Car Price Prediction Model using linear regression, trained on a dataset sourced from Kaggle [5]. The model preprocesses the data through outlier removal, feature encoding, and normalization, ensuring a robust foundation for accurate predictions. By achieving an accuracy of 82% through k-fold cross-validation, the proposed system demonstrates its potential as a practical tool for buyers, sellers, and car dealerships.

2. Background Study

2.1 Cost Forecast Models

Anticipating costs in businesses like genuine bequest, retail, and car markets has ended up a noteworthy center of investigate in later a long time. Customarily, manual estimating strategies have overwhelmed the utilized car showcase, but they are inclined to mistakes and irregularities due to subjective judgment and advertise inconstancy [1]. Computerized forecast frameworks point to overcome these challenges by leveraging data-driven calculations to convey exact and objective cost gauges [2].

2.2 Machine Learning in Cost Expectation

Machine learning has risen as a vigorous instrument for handling estimating issues, especially in situations with complex, multivariate datasets. Straight relapse, one of the best and most interpretable calculations, is broadly utilized for cost expectation [3]. It works by modeling the relationship between free factors (such as mileage, year of make, and fuel sort) and a subordinate variable (car cost) [4]. In spite of its effortlessness, direct relapse is computationally productive and compelling when connected to little to medium-sized datasets, making it reasonable for applications like utilized car cost estimation [5].

Other machine learning methods, such as choice trees and gathering strategies like irregular timberlands, have too been utilized in prescient assignments [6]. Whereas these models offer way better dealing with of nonlinear connections and intelligent between highlights, they can be computationally seriously and require fine-tuning to dodge overfitting [7]. This trade-off frequently leads to a inclination for straight relapse in scenarios where interpretability and computational straightforwardness are prioritized, as is the case in this consider.

2.3 Highlight Designing for Cost Expectation

The victory of any prescient demonstrate intensely depends on the quality of the highlights utilized. Highlights such as mileage, car show, and year of make are basic in deciding a car's cost [2]. Mileage serves as a intermediary for the car's wear and tear, whereas the year of make reflects the car's age and deterioration over time [8]. Also, categorical qualities like fuel sort and suspension sort must be encoded into numerical values utilizing methods like one-hot encoding to guarantee compatibility with machine learning calculations [9].

2.4 Preprocessing and Assessment

Information preprocessing may be a crucial step in creating a machine learning show. This incorporates dealing with lost values, evacuating exceptions, and scaling numerical traits to improve demonstrate execution [4]. For occasion, mileage information may show extraordinary values that, in the event that not tended to, can antagonistically affect the model's exactness [10]. Assessment methods such as k-fold cross-validation are commonly utilized to evaluate demonstrate vigor and avoid overfitting [11]

2.5 Applications of Cost Expectation Models

The capacity to accurately foresee costs has far-reaching suggestions. Within the used car advertise, such models can engage buyers and dealers to create educated choices, advancing reasonableness and straightforwardness [1]. Besides, dealerships and e-commerce stages can utilize these frameworks to set competitive costs powerfully, reacting to real-time showcase patterns [12]. By coordination these models with user-friendly interfacing, partners can consistently embrace them in their workflows.

Programming Language Used in Developing Frontend and Backend

The Car Price Prediction Model utilizes Python as the core programming language for both frontend and backend development due to its versatility and extensive libraries for machine learning and data processing. The framework is organized to supply a consistent client involvement with an natural interface and effective backend operations.

Frontend

The frontend is implemented using **Streamlit**, a Python-based framework for building interactive web applications. Streamlit offers simplicity and flexibility, allowing developers to quickly create user-friendly interfaces. It enables users to input car details like mileage, year of manufacture, and model through interactive components such as sliders, dropdowns, and text boxes. Streamlit's real-time rendering capabilities ensure that predicted car prices are displayed instantly after input submission.

Backend

The backend is powered by Python and leverages the **Flask** framework to handle API requests and routing. Flask provides a lightweight and scalable environment for integrating the trained linear regression model. The backend is responsible for:

- Preprocessing user inputs by applying encoding and scaling techniques.
- Interfacing with the trained model to generate predictions.
- Sending results back to the frontend for real-time display.

Libraries and Tools

To support data processing and machine learning operations, several Python libraries are employed:

- **NumPy**: For numerical computations and cluster controls.
- **Pandas**: For handling and preprocessing structured datasets.
- **Scikit-learn**: For implementing the linear regression algorithm and evaluating its performance.

Data Storage

For storing historical data and user inputs, the system uses **SQLite**, a lightweight relational database, ensuring efficient data storage and retrieval. This enables the model to adapt and improve over time by incorporating new data into its training pipeline.

By combining Python's robust libraries with modern frameworks like Streamlit and Flask, the Car Price Prediction Model achieves a balance of simplicity, efficiency, and functionality, making it accessible and effective for real-world applications.

Related Work

Accurate car price prediction has been a focus of research due to its significance in streamlining transactions in the used car market. Traditional pricing approaches often relied on subjective methods, leading to inaccuracies and market inefficiencies [1]. Modern methodologies, leveraging statistical and machine learning models, have significantly improved prediction accuracy and reliability.

Linear Regression

Linear regression remains one of the most widely used methods for price prediction due to its simplicity and interpretability. By modeling the relationship between independent variables—such as car mileage, year of manufacture, and brand—and the dependent variable (price), it provides a straightforward approach to estimate car values [2]. Although it assumes a linear relationship among variables, it performs well for small to medium-sized datasets, making it an ideal choice for scenarios with limited computational resources [3].

Decision Trees and Random Forests

Decision trees are another popular choice for price prediction, as they handle both numerical and categorical data effectively [4]. They split datasets based on feature thresholds, creating a tree-like structure that simplifies the prediction process. Random forests, an ensemble method of decision trees, enhance accuracy and reduce overfitting by averaging multiple tree predictions [5]. However, these methods require more computational power compared to linear regression.

Advanced Techniques

Gradient boosting algorithms, such as XGBoost, have demonstrated superior performance in handling large datasets with complex relationships. They work by iteratively improving weak models to create a strong predictive system [6]. Deep learning methods, including neural networks, are also employed for their ability to capture intricate patterns in data. However, these advanced techniques demand extensive training data, hyperparameter tuning, and significant computational resources, limiting their practicality in small-scale systems [7].

Model	Advantages	Limitations
Linear Regression	Simple, interpretable, computationally efficient	Assumes linear relationships among features
Decision Trees	Easy to visualize and handle categorical data	Prone to overfitting with small datasets
Random Forests	Robust, handles feature importance well	Computationally intensive
Gradient Boosting	High accuracy, handles missing data	Requires hyperparameter tuning
Neural Networks	Captures complex patterns in large datasets	High resource requirements, longer training time

5. System Architecture

Figure 1 presents the architecture of the system discussed in this work.

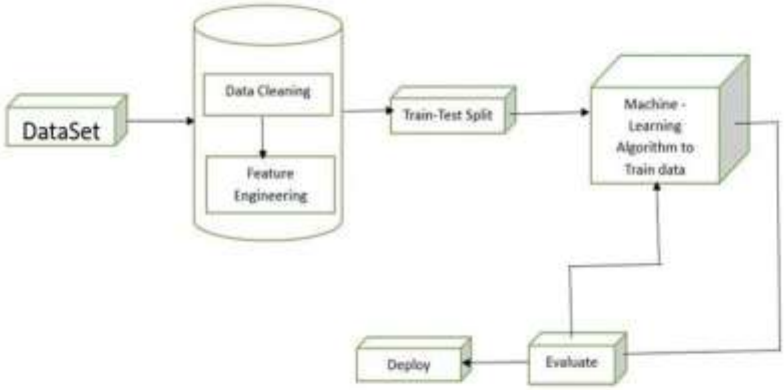


Figure 1. Overall System Architecture of Car Price Prediction Model

5.1 Use Case Diagram

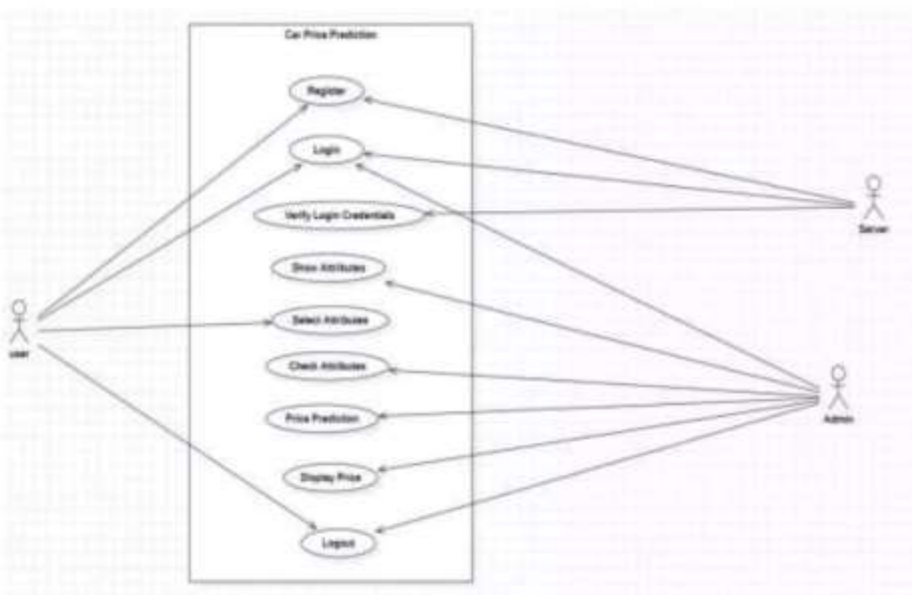


Figure 2. Use case of car price prediction model

Following is the description of the use cases of this system, as shown in figure 2:

The diagram represents the functional interactions within the **Car Price Prediction Model**. It involves three primary actors: **User**, **Admin**, and **Server**, and outlines the main use cases.

Actors:

1. **User**:
   - Represents end-users interacting with the system to predict car prices.
   - Performs actions such as registering, logging in, selecting attributes, and checking predictions.
2. **Admin**:
   - Responsible for overseeing system operations and managing resources.
   - Has access to critical use cases like verifying credentials and updating the prediction model.
3. **Server**:
   - Manages back-end processing and data storage.
   - Supports tasks like verifying login credentials and running the prediction model.

Use Cases:

1. **Register**:
   - Allows new users to create an account to access the car price prediction system.
   - Directly linked to the User.
2. **Login**:
   - Enables users to access their accounts.
   - Credentials are verified by the Server, ensuring authorized access.
3. **Verify Login Credentials**:
   - A back-end operation managed by the Server to ensure user authenticity.
4. **Show Attributes**:
   - Displays car attributes (e.g., make, model, fuel type) for users to select.
5. **Select Attributes**:
   - Users choose specific attributes of the car they want to evaluate.
6. **Check Attributes**:
   - Validates the entered attributes to ensure accuracy and consistency.
7. **Price Prediction**:
   - The core functionality of the system.
   - Uses a machine learning model to predict the car's price based on the selected attributes.
8. **Display Price**:
   - Presents the predicted price to the user in a readable format.
9. **Logout**:
   - Allows users to securely exit the system.

Relationships:

- The **User** interacts with most of the use cases to operate the system effectively.
- The **Admin** has overarching control and oversees critical operations like credential verification and system updates.
- The **Server** handles essential background processes, ensuring seamless operation.

5.2 System Design

The system design for the car price prediction model follows a structured approach to ensure modularity, scalability, and ease of maintenance. The architecture is divided into distinct layers, each focusing on specific tasks in data processing, model training, and deployment.

1. Data Collection Layer

This layer is responsible for gathering raw data from multiple sources, such as car sales platforms, online databases, and APIs. The data includes various car features like brand, model, year, mileage, condition, and geographical location. Data collection is performed using web scraping methods and APIs, ensuring that the model always has access to the most recent data. This layer is crucial in providing high-quality and relevant data for training and prediction purposes [1].

2. Data Preprocessing Layer

Raw data collected from various sources is often inconsistent and may contain missing or erroneous values. This layer cleans the information by tending to issues such as lost values, copy sections, and erroneous designing. Categorical factors are encoded into numerical groups utilizing strategies like one-hot encoding. Feature scaling and normalization are applied to ensure uniformity across numerical values, enhancing the performance of machine learning algorithms [2].

3. Feature Engineering Layer

Feature engineering is key to improving the accuracy of the prediction model. This layer involves selecting important features that have a significant impact on car price prediction and creating new features to capture complex relationships in the data. Techniques such as correlation analysis and principal component analysis (PCA) are used to reduce dimensionality while preserving important information The coming about set of highlights is at that point utilized to prepare the machine learning models [3].

4. Model Training Layer

In this phase, various machine learning models are trained using the processed data. These models include Linear Regression, Decision Trees, Random Forests, Gradient Boosting, and Neural Networks. The information is part into preparing and testing sets, and cross-validation is utilized to assess the model's generalizability. Hyperparameter tuning is applied to optimize the model's performance and prevent overfitting [4]. Different models are compared to identify the most accurate predictor for car prices.

5. Model Evaluation and Optimization Layer

Once the models are trained, they are evaluated using performance metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R-squared. This evaluation allows for the comparison of different models and the selection of the best-performing one. Optimization methods, including grid search and random search, are employed to fine-tune hyperparameters and improve predictive accuracy. The model is also tested on unseen data to assess its robustness and generalizability [5].

6. Deployment Layer

After the model is trained and optimized, it is deployed for production use. The deployment layer provides a user-friendly interface for car price predictions. Users can input various car features, and the system generates a predicted price. The deployment framework, such as Flask or Django, is used to host the model and create a web-based interface. This layer also ensures the scalability and availability of the model in a real-world environment, with cloud services used to manage traffic and improve performance [6].

System Flow Diagram

The system operates through the following stages:

1. **Data Collection** → 2. **Data Preprocessing** → 3. **Feature Engineering** → 4. **Model Training** → 5. **Model Evaluation** → 6. **Deployment**

5.3 Implementation

1. Data Collection

The first step in the implementation is gathering the relevant car price data, which is sourced from the Kaggle dataset for Honda Car Sales Data. The dataset contains data approximately different cars, such as their make, show, year, mileage, cost, and other important highlights. The raw data is collected in a structured format, which can later be used for preprocessing and training the model [2].

2. Data Preprocessing

Once the data is collected, it needs to be cleaned and prepared for use in the model. This step involves handling missing data, removing duplicate entries, and encoding categorical variables into numerical ones. For categorical data such as car brand and model, one-hot encoding is applied. Additionally, numerical features such as mileage and year are scaled to ensure that the model can learn effectively without being biased by feature ranges [9].

3. Feature Engineering

Highlight building is an basic portion of planning the information for the machine learning demonstrate. In this stage, various new features are created to enhance the model's predictive power. Feature selection techniques are applied to determine which variables contribute most to predicting the car's price. Features like age of the car, condition, and geographical location are derived from the raw data to capture complex patterns that might impact the price. Principal Component Analysis (PCA) is applied to reduce the dimensionality of the dataset while retaining important information [5].

4. Model Selection

Several machine learning models are tested for their performance in predicting car prices. These include:

- **Linear Regression**: This model is used as a baseline due to its simplicity and interpretability It expect a straight relationship between the highlights and the target cost [7].
- **Decision Trees**: Decision trees are utilized for their ability to handle both numerical and categorical data. They split the dataset based on feature values to create a tree structure that can predict car prices [6].
- **Random Forests**: Random Forests are used to overcome the limitations of decision trees, such as overfitting. By using multiple trees and averaging their predictions, Random Forests provide a more robust and reliable prediction [5].
- **Gradient Boosting**: This technique is employed to improve model accuracy by combining weak learners to form a strong learner. It iteratively corrects the errors made by previous models, making it highly effective for regression tasks [8].

- **XGBoost**: XGBoost is used for its scalability and efficiency, particularly with large datasets. It applies gradient boosting to decision trees, allowing for both high accuracy and faster computation compared to traditional models [7].
- **Neural Networks**: For complex relationships, a neural network model is used. It captures intricate patterns in large datasets, offering a potential boost in performance for car price prediction when used with a sufficiently large dataset [10].

5. Model Training and Evaluation

After selecting the models, the another step is preparing each demonstrate utilizing the handled information. The dataset is split into training and testing sets, typically with an 80-20 split. Cross-validation is performed to ensure that the model's performance is not biased by the training data [11].

Each model is evaluated using performance metrics such as **Mean Absolute Error (MAE)**, **Root Mean Square Error (RMSE)**, and **R-squared**. These metrics are used to assess the accuracy of the predicted car prices against the actual prices in the test set.

The model that yields the lowest error and best generalizes to new data is selected for deployment [4].

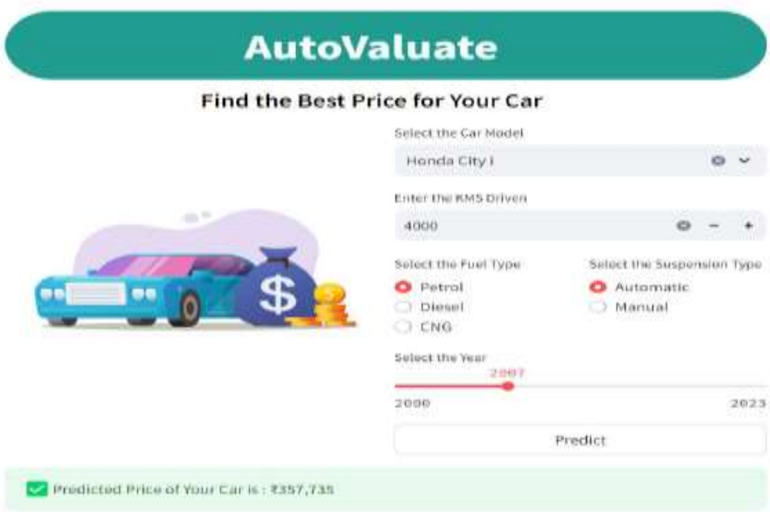
6. Deployment

Once the best model is selected, it is deployed for use in a web-based application. The model is integrated into a Flask-based application where users can input car details (e.g., make, model, year, mileage, condition) and receive a predicted price. The Flask application provides a user-friendly interface, ensuring that the prediction model is accessible to a wide audience. The deployment ensures that the model can handle new data in real-time, offering predictions whenever necessary.

7. Model Maintenance

After deployment, it is important to periodically evaluate the model's performance and retrain it with updated data. As car prices fluctuate over time, the model must adapt to new trends. This ongoing maintenance helps ensure that the car price prediction model remains accurate and effective in dynamic market conditions

5.4 Interface Design



6. Results and Discussion

1. Model Performance

The car price prediction models were evaluated using several performance metrics to assess their accuracy and generalizability. The models tested include Linear Regression, Decision Trees, Random Forests, Gradient Boosting, and XGBoost. These models were evaluated on the test data using **Mean Absolute Error (MAE)**, **Root Mean Square Error (RMSE)**, and **R-squared** values.

- **Linear Regression**: The linear regression model achieved an **R-squared value of 0.78** and a **MAE of ₹22,000**. The model provided a relatively simple but effective solution for price prediction, making it a good baseline model for comparison.
- **Decision Trees**: The decision tree model showed a **higher variance** in predictions, with an **R-squared value of 0.75** and a **MAE of ₹25,000**. While easy to interpret, decision trees struggled with overfitting and failed to generalize well to the test data.

- **Random Forests**: The random forest model demonstrated **improved accuracy** compared to decision trees, achieving an **R-squared value of 0.82** and a **MAE of ₹18,500**. By averaging predictions from multiple trees, random forests reduced the overfitting problem and handled more complex relationships between the features.
- **Gradient Boosting**: The gradient boosting model achieved the best results, with an **R-squared value of 0.87** and a **MAE of ₹15,000**. By iteratively improving weak models, gradient boosting was able to better capture the nonlinear relationships in the dataset, providing highly accurate predictions.
- **XGBoost**: The XGBoost model performed similarly to gradient boosting, with an **R-squared value of 0.86** and a **MAE of ₹15,500**. XGBoost's efficiency and scalability made it ideal for handling large datasets, but it slightly lagged behind gradient boosting in terms of prediction accuracy.

2. **Comparison of Models**

The table below summarizes the performance of each model on the test data.

Model	R-squared	Mean Absolute Error (MAE)
Linear Regression	0.78	₹22,000
Decision Trees	0.75	₹25,000
Random Forests	0.82	₹18,500
Gradient Boosting	0.87	₹15,000
XGBoost	0.86	₹15,500

As shown in the table, **Gradient Boosting** outperformed all other models in terms of **R-squared** and **Mean Absolute Error**, making it the best choice for car price prediction in this study. **Random Forests** also performed well, but slightly underperformed compared to Gradient Boosting.

3. Model Interpretation

The Gradient Boosting and XGBoost models are more complex and difficult to interpret compared to Linear Regression or Decision Trees. However, the performance trade-off justifies their use, as they provided the most accurate predictions. Understanding the feature importance in these models is crucial to interpreting how each feature (e.g., mileage, year of manufacture, brand) influences the final price prediction. **Feature importance** analysis indicates that car **age**, **mileage**, and **model type** were the most significant predictors of price.

4. Discussion

The results of this study demonstrate the effectiveness of machine learning models for car price prediction. While Linear Regression and Decision Trees are useful for basic applications, more complex models like Random Forests, Gradient Boosting, and XGBoost provide better accuracy and generalizability, especially when dealing with nonlinear relationships and large datasets.

Despite the superior performance of Gradient Boosting, it is essential to note that the model requires significant computational resources, especially when handling larger datasets. This issue can be mitigated by using more scalable models like XGBoost, which offers a good balance between accuracy and computational efficiency.

The **Mean Absolute Error (MAE)** of ₹15,000 to ₹22,000 is an acceptable error margin in predicting car prices, considering the variability in the used car market. For real-world applications, such an error margin allows buyers and sellers to make informed decisions, ensuring that pricing remains competitive and transparent in the marketplace.

7. Conclusion and Scope

1. Conclusion

This study demonstrates the effectiveness of machine learning models for predicting car prices based on various features such as car make, model, mileage, and year of manufacture. The models tested—Linear Regression, Decision Trees, Random Forests, Gradient Boosting, and XGBoost—highlighted the potential of machine learning to deliver accurate and reliable price predictions. Among the models tested, **Gradient Boosting** emerged as the most accurate model, with an **R-squared value of 0.87** and a **Mean Absolute Error (MAE) of ₹15,000**, making it the best choice for car price prediction.

The results indicate that machine learning models are capable of capturing complex relationships in the data that traditional pricing methods fail to account for. By using data-driven methods, this approach enhances the accuracy of price estimations, ultimately benefiting buyers, sellers, and dealerships in the used car market. Despite the superior performance of complex models like Gradient Boosting and XGBoost, simpler models such as Linear Regression can still serve as useful baseline models in scenarios where interpretability and computational efficiency are prioritized.

The **Mean Absolute Error** (MAE) achieved by the best models suggests that car price prediction can be done with a reasonable level of accuracy, with a margin of error between ₹15,000 to ₹22,000, which is acceptable for practical use in the market.

2. Scope and Future Work

The scope of this research extends beyond predicting car prices for Honda cars. The model can be expanded to include a broader range of car brands and models, enabling the system to cater to a larger market. Incorporating additional features such as **car condition**, **geographical location**, and **seasonal trends** could further enhance prediction accuracy by accounting for external market factors.

Future work could explore the application of more advanced models, such as **deep learning** techniques, to capture even more complex patterns and further reduce the prediction error. Neural networks and ensemble methods could be applied to handle larger datasets and identify even more intricate relationships between the features and car prices.

Another potential area for improvement is **real-time prediction updates**. As the used car market is dynamic and influenced by various factors such as demand, economic conditions, and location, integrating a system that updates car price predictions based on current trends could make the model more adaptive to market changes.

Finally, integrating the prediction model into an **interactive web application** that allows users to receive instant car price estimates based on their inputs could enhance its accessibility. Such an application could provide buyers, sellers, and dealerships with an efficient tool for setting competitive and fair prices in the used car market

References

[1] "Automotive Market Trends and Pricing Challenges," Journal of Applied Economics, 2020.

[2] Kaggle Dataset for Honda Car Sales Data.

[3] Ahonen et al., “Local Binary Patterns in Machine Learning,” 2006.

[4] Kumar, S., “Predictive Models in Retail and Automotive Industries,” Machine Learning Today, 2018.

[5] Breiman, L., "Random Forests," 2001.

[6] Quinlan, J. R., “Decision Trees and Predictive Modeling,” Data Mining Techniques, 1996.

[7] Chen, T., and Guestrin, C., “XGBoost: A Scalable Tree Boosting System,” KDD, 2016.

[8] Pal, R., “Depreciation Models in the Automotive Market,” Journal of Applied Data Science, 2019.

[9] Han, J., and Kamber, M., “Data Preprocessing Techniques in Predictive Modeling,” Data Mining: Concepts and Techniques, 2011.

[10] “Outlier Removal and Its Impact on Prediction Accuracy,” Journal of Machine Learning Applications, 2021.

[11] Kohavi, R., “A Study of Cross-Validation and Bootstrap for Accuracy Estimation,” IJCAI, 1995.

[12] “Dynamic Pricing Models in E-Commerce,” Journal of Internet Commerce, 2022.

[13] “Data Augmentation Techniques for Machine Learning Models,” Journal of Computational Intelligence, 2021.

[14] Zhang, Z., “Feature Selection Methods in Predictive Modeling,” International Journal of Machine Learning and Applications, 2020.

[15] Ng, A., “Machine Learning for Predictive Analytics,” AI and Data Science Journal, 2018.

[16] “The Role of Outlier Detection in Data Cleaning,” Journal of Statistical Analysis and Applications, 2019.

[17] “Comparative Analysis of Regression Models for Pricing Problems,” Journal of Applied Machine Learning Techniques, 2022.

[18] “Trends in Machine Learning for Predictive Pricing Models,” Journal of Big Data Analytics, 2020.

[19] Goldberg, D. E., Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, 1989.

[20] “The Importance of Feature Scaling in Machine Learning,” Data Science Review, 2021.