Georgia State University

CSC 6780 & DSCI 4780 – Fundamentals of Data Science

Fall 2024

Project Progress Report

CAR RESALE PRICE PREDICTION

The Outliers

Cunjama Bryan

Jayappa Bhavana

Padamata Sai Pavan

Gaikwad Shreya Nilesh

Table of Contents

1 Business Understanding

1.1 Business Problem

The project focuses on Car Resale Price Prediction using machine learning techniques to predict the resale value of used cars. Correct prediction of resale price is of great importance in the automotive industry because it aids sellers in optimizing prices and buyers in making an informed purchasing decision. Various factors act upon resale prices of cars, from registered year, engine capacity, mileage, fuel type, and owner history to even the city in which it is sold. The project will identify and analyze such factors using real data while building a reliable regression model for predicting car resale prices with high accuracy. Thus, this will provide practical insight into how different car attributes affect the car's devaluation and resale value.

The motivation behind this work is the increasing importance of data-driven decision-making in the automotive market, and especially within the used car market segment, which has grown very fast in India. The majority of the sellers do not optimally set the price and, therefore, either overprice their cars to lose potential buyers or underprice them to miss out on profit. On the other hand, buyers have to deal with uncertainty when trying to estimate whether they pay a fair price for the used vehicle. This work tries to overcome these challenges by creating a fair and data-driven pricing model for buyers as well as car sellers in enabling machine learning algorithms to project resale prices.

1.2 Dataset

The dataset used for this project is the 'Car Resale data – 2023' available on Kaggle and can be accessed here. It contains 17,446 rows and 14 features, which include information about various descriptive features such as registered year, engine capacity in cc, transmission type, kilometers driven, owner type, fuel type, maximum power in HP, number of seats, mileage in km per liter, body type and the target variable (resale price), which is useful for building a predictive model for car price estimation.

Key Features:

Descriptive Features:

full_name: The name of the car model (string).

registered_year: The manufacturing year of the car (numeric).

kms_driven: The total kilometers driven by the car (numeric).

fuel_type: The fuel type of the car (categorical: Petrol, Diesel, CNG).

transmission_type**:** The car's transmission type (categorical: Manual, Automatic).

owner_type: The number of previous owners (numeric).

mileage: The mileage of the vehicle (numeric, in km/l).

Target feature:

resale_price: representing the price at which the car is sold in the resale market(numeric).

The dataset, provided in CSV format is suitable for analysis using Python (Pandas), R, or Excel. The above dataset has some categorical columns that may need encoding (e.g., fuel_type, transmission_type). There are also numerical columns that may need scaling or imputation, especially for missing values in attributes like mileage and engine_capacity. This wide range of descriptive features makes conducting in-depth analysis and creating price-prediction models possible. It is perfect for machine learning regression applications investigating the variables influencing car resale values.

1.3 Proposed Analytics Solution

To predict car resale prices, we will start by exploring the dataset to uncover patterns and relationships between the features and the target variable (resale price). This step involves looking at how different attributes, like mileage, engine size, and fuel type, affect the car's resale value. Visual tools such as scatter plots, bar charts, and histograms will help us better understand the data and highlight any important trends.

Next, we'll clean and prepare the data to ensure it's ready for analysis. Any missing information will be filled in appropriately to avoid gaps in the dataset. Categorical information, like whether a car uses petrol or diesel, will be converted into numbers so it can be used by machine learning models. Numerical features, such as mileage and engine capacity, will be adjusted to a consistent scale so that all features contribute fairly to the predictions.

We will use a mix of machine-learning algorithms to build our model:

- **Linear Regression**: A simple method to predict resale prices and give us a clear view of how each feature contributes to the outcome.
- **Random Forest**: A powerful model that combines multiple decision trees to make accurate predictions. It works well with datasets like ours, which have a mix of different types of features.
- **Support Vector Regression (SVR)**: This method uses advanced techniques to find patterns in the data and is particularly useful for making precise predictions.
- **Neural Network (Multilayer Perceptrons)**: This is a more advanced approach that can handle complex relationships between features, allowing us to capture patterns that simpler methods might miss.

Finally, we'll evaluate how well these models perform by checking their accuracy using metrics like Mean Absolute Error (MAE) and R-squared. These measures will help us understand how close our predictions are to the actual resale prices. We'll also use cross-validation to ensure that our model performs consistently across different parts of the dataset.

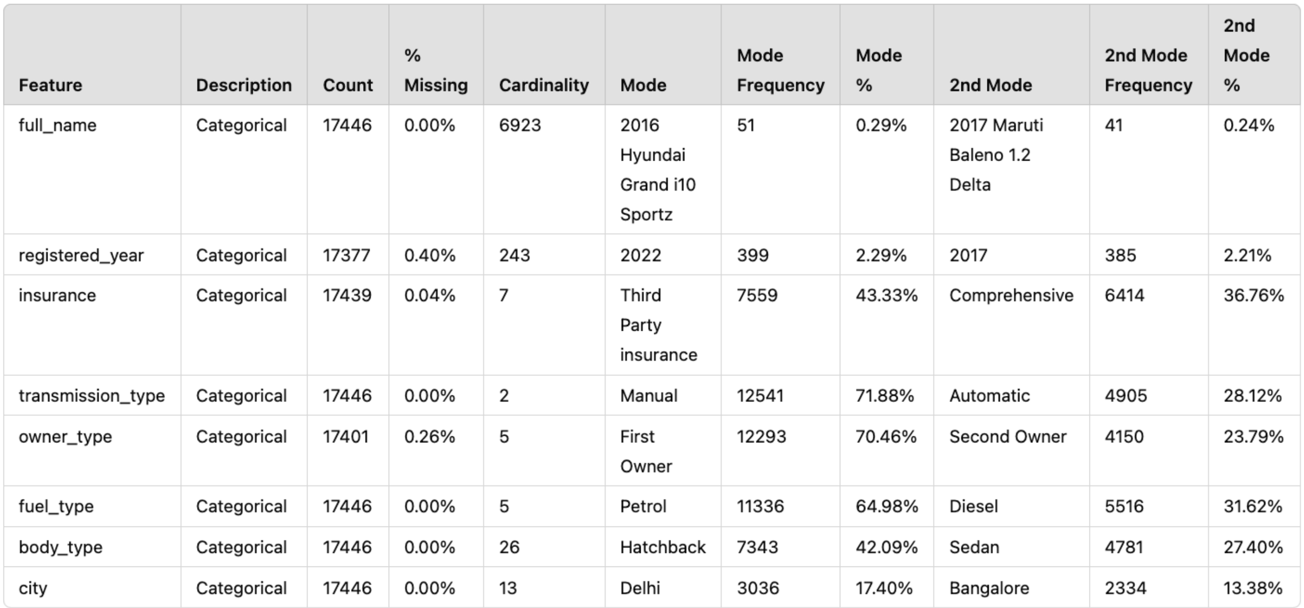
This process will help us create a reliable and easy-to-understand predictive model, making it useful for real-world applications in the car resale market.

2 Data Exploration and Preprocessing

2.1 Data Quality Report

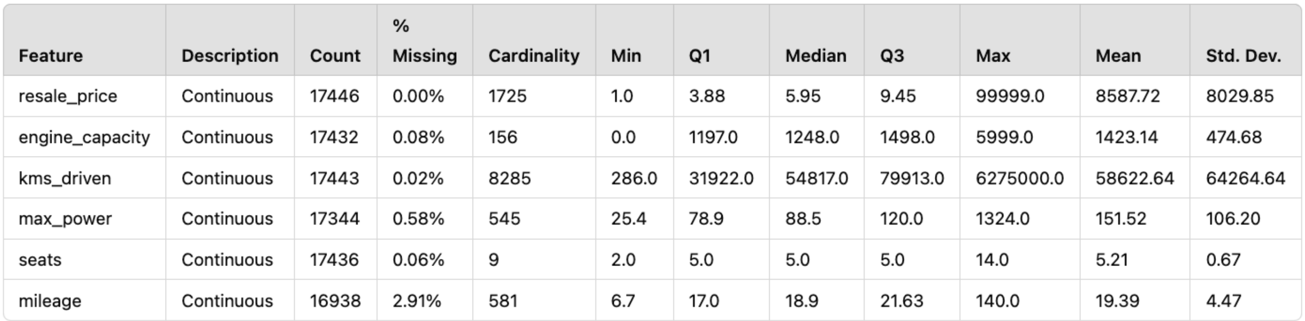
In our data quality analysis, we began by dividing the dataset into two main groups: continuous and categorical features. We identified the continuous columns based on their numeric nature and stored them in the continuous_cols list, which included variables such as resale_price, engine_capacity, and kms_driven. After identifying these columns, we preprocessed them by cleaning any non-numeric characters and converting them to appropriate numeric types. For the categorical features, we used the categorical_cols list, which included columns like full_name, insurance, and fuel_type. We then examined both sets of features to assess their completeness, cardinality, mode distributions, and any missing data, producing data quality reports for each.

Table 1. Data Quality Report for Categorical Features



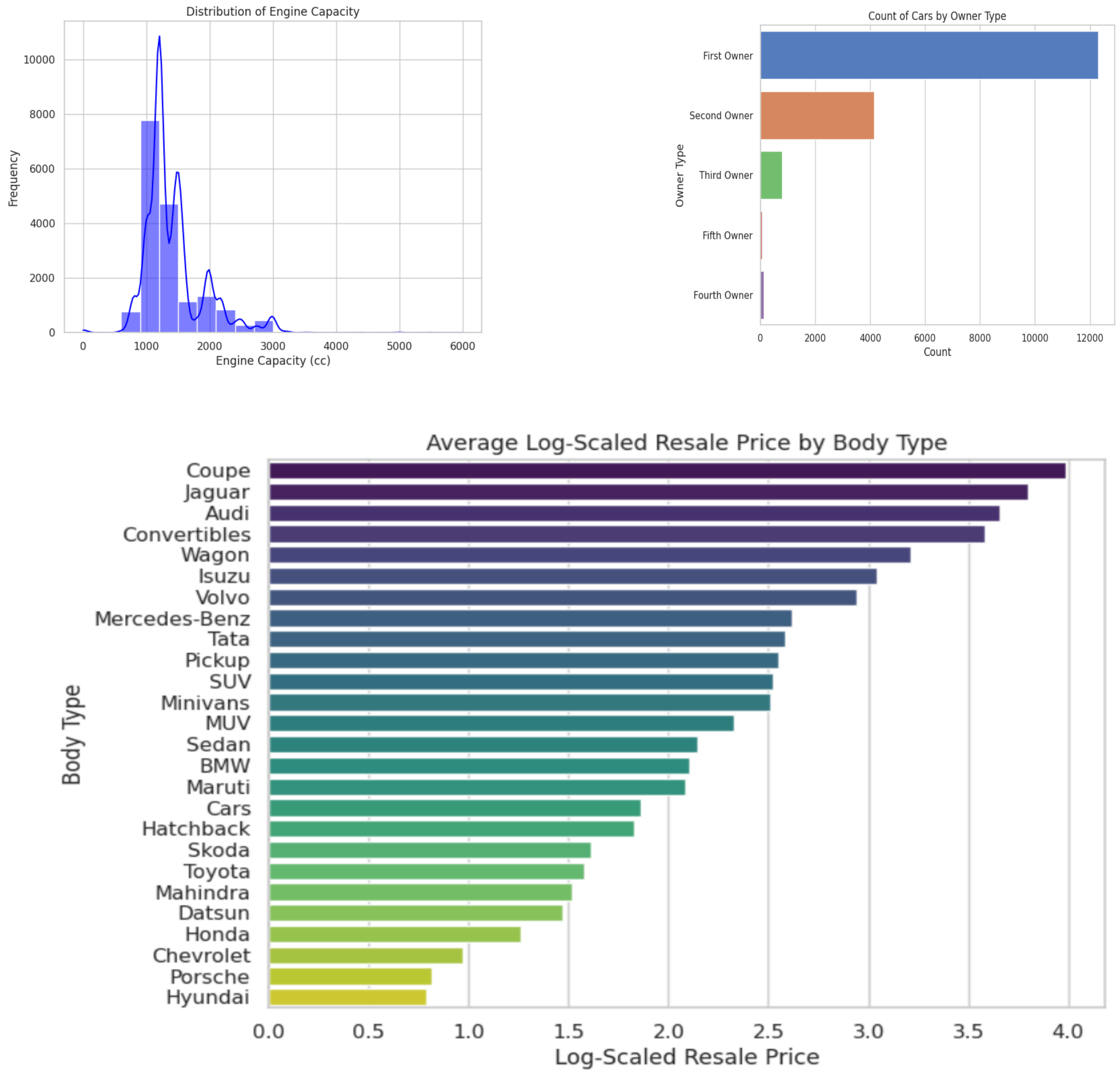
Feature	Description	Count	% Missing	Cardinality	Mode	Mode Frequency	Mode %	2nd Mode	2nd Mode Frequency	2nd Mode %
full_name	Categorical	17446	0.00%	6923	2016 Hyundai Grand i10 Sportz	51	0.29%	2017 Maruti Baleno 1.2 Delta	41	0.24%
registered_year	Categorical	17377	0.40%	243	2022	399	2.29%	2017	385	2.21%
insurance	Categorical	17439	0.04%	7	Third Party insurance	7559	43.33%	Comprehensive	6414	36.76%
transmission_type	Categorical	17446	0.00%	2	Manual	12541	71.88%	Automatic	4905	28.12%
owner_type	Categorical	17401	0.26%	5	First Owner	12293	70.46%	Second Owner	4150	23.79%
fuel_type	Categorical	17446	0.00%	5	Petrol	11336	64.98%	Diesel	5516	31.62%
body_type	Categorical	17446	0.00%	26	Hatchback	7343	42.09%	Sedan	4781	27.40%
city	Categorical	17446	0.00%	13	Delhi	3036	17.40%	Bangalore	2334	13.38%

Table 2. Data Quality Report for Continuous Features



Feature	Description	Count	% Missing	Cardinality	Min	Q1	Median	Q3	Max	Mean	Std. Dev.
resale_price	Continuous	17446	0.00%	1725	1.0	3.88	5.95	9.45	99999.0	8587.72	8029.85
engine_capacity	Continuous	17432	0.08%	156	0.0	1197.0	1248.0	1498.0	5999.0	1423.14	474.68
kms_driven	Continuous	17443	0.02%	8285	286.0	31922.0	54817.0	79913.0	6275000.0	58622.64	64264.64
max_power	Continuous	17344	0.58%	545	25.4	78.9	88.5	120.0	1324.0	151.52	106.20
seats	Continuous	17436	0.06%	9	2.0	5.0	5.0	5.0	14.0	5.21	0.67
mileage	Continuous	16938	2.91%	581	6.7	17.0	18.9	21.63	140.0	19.39	4.47

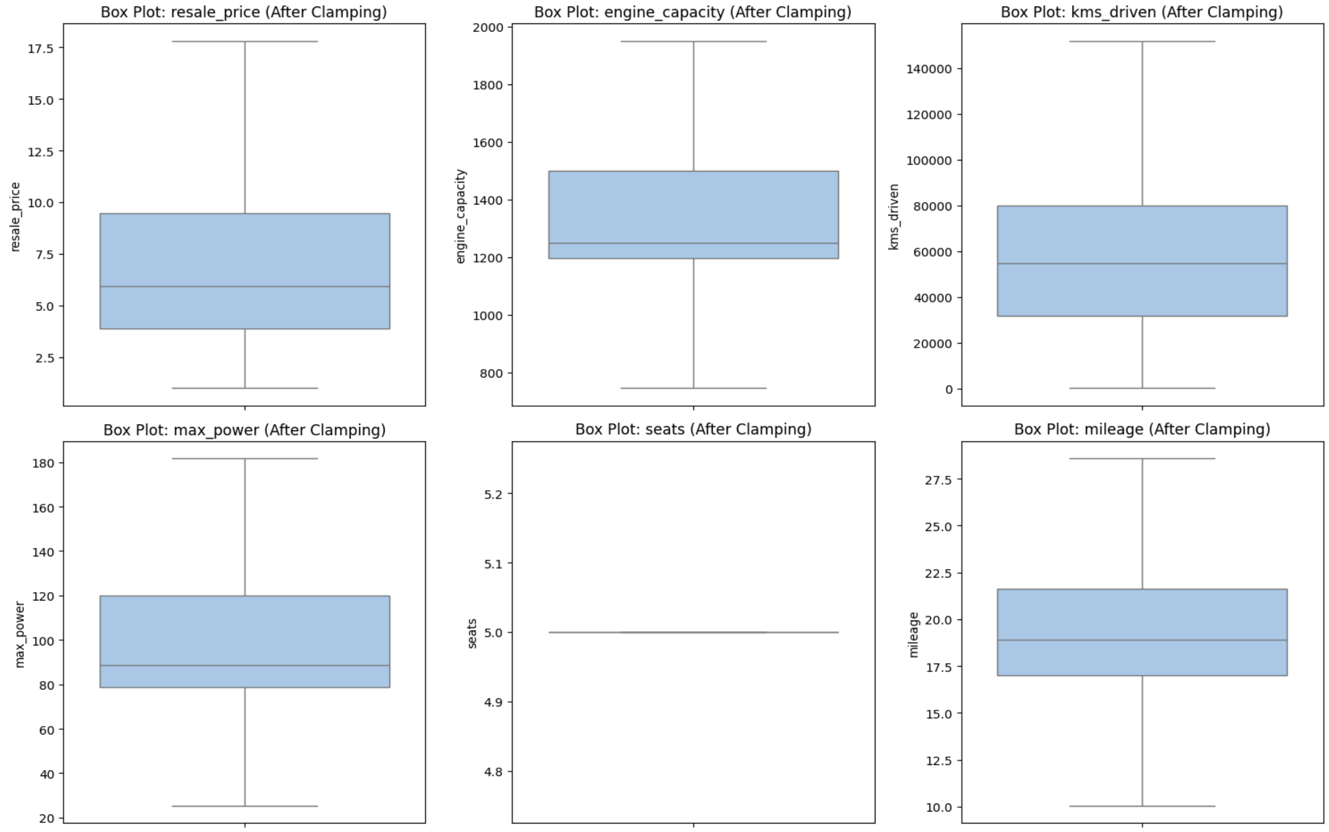
Figure 1. Visualizations of Categorical and Continuous Features in Dataset



2.2 Missing Values and Outliers

Missing Values: The missing values in the dataset were checked, and it was observed that some features contained missing data. Further, the percentage of missing values in every feature was determined in descending order of priority for attention. These analyses give an idea about how much data is missing; thus, they help to inform decisions that might be made in the future regarding the exclusion or imputation of features.

Outliers in the continuous variables resale_price, engine_capacity, kms_driven, max_power, seats, and mileage were found using the IQR method. Lower and upper bounds for each feature were computed and observations outside these bounds were considered as outliers. Quantification has also been performed regarding the count and percent of outliers for each variable. That means to say it shows those features which have large deviations because for robust modeling, those features have to be either transformed or dropped. Finally, a report on all the results after analysis is the last recommendation.



2.3 Normalization

Numerical features were normalized according to the following three methods in pre-processing: Min-Max scaling, a logarithmic transformation, and using the Z-score. First, Min-Max normalization was applied to a range from 10 to 100,000 to scale the values of different features. Then, logarithmic scaling was employed to handle skewness and compress wide ranges in the data, being mindful of adjusting for small or zero values accordingly without causing computational errors. The features were then normalized around zero using Z-score normalization with a unit variance. A new DataFrame was stored after normalization, and that had provided clean and uniform input for subsequent analyses.

2.4 Transformations

The new feature 'price_per_km' captures the relation of resale price with kilometers driven in value per unit distance. This transformation enables to put resale prices into context; taking usage into account might show something about pricing trends and could be useful for model interpretability. Further investigation will be carried out on the derived feature to ascertain its predictive importance using both the original dataset and the prepared dataset.

2.5 Feature Selection

Feature selection is a crucial stage in the machine learning process that aims to improve model performance by removing unnecessary or redundant features while finding the most pertinent ones. Full_name and city are examples of non-informative columns that are eliminated during feature selection since they don't provide the prediction model with useful numerical information. LabelEncoder is also used to encode categorical variables like insurance, owner_type, fuel_type, body_type, and transmission_type in order to convert them into a numerical format that can be used with Random Forest methods. By using this method, the dataset is guaranteed to contain just those attributes that have a substantial impact on the model's ability to predict. A more efficient and accurate machine learning model is produced by reducing noise, increasing computing efficiency, and minimizing overfitting risks by concentrating on pertinent characteristics.

3. Model Selection and Evaluation

3.1 Evaluation Metrics

Evaluation metrics are necessary to qualify the performance of regression models, and to make some judgments about their predictive accuracy and reliability. In our project, we have utilized four important metrics: MSE, RMSE, MAE, and the R^2 Score.

MSE calculates the average of the squared differences between actual and predicted values. It gives heavy penalties for larger errors.
The RMSE, being the square root of MSE, gives an error measure in the same units as that of the target variable and hence is more interpretable for practical purposes.
MAE is the average of absolute differences between actual and predicted values. It is an intuitive measure that is less sensitive to outliers than MSE or RMSE.
R^2 Score is the measure of the coefficient of determination that tells what proportion of variance in the target variable the model is explaining, and a value closer to 1 indicates a better performance.
All together, these metrics comprehensively judge the model, pointing out strengths and further opportunities for predictive accuracy and consistency.

3.2 Models

Linear Regression: In the car resale price prediction project, Linear Regression was employed as a baseline model to establish the relationship between features such as mileage, engine capacity, car age, and the resale price. This algorithm predicts the target variable by fitting a straight line through the data, minimizing the difference between actual and predicted values. To prepare the data, categorical variables were encoded using LabelEncoder, and the dataset was split into training and testing sets for objective evaluation. Performance metrics like MSE, RMSE, MAE, and R^2 score were used to assess the model, which provided a basic understanding of how the features influence the target variable. While effective for capturing linear trends, the model’s limitations in handling non-linear relationships highlighted the need for more complex approaches in subsequent steps.

Random Forest Regressor: Building on the insights gained from Linear Regression, Random Forest was employed as a more advanced model to address its limitations in capturing non-linear relationships between features such as mileage, engine capacity, car age, and resale price. Random Forest, an ensemble learning technique, constructs multiple decision trees and averages their predictions to enhance accuracy and robustness against overfitting. Using the preprocessed dataset, where categorical variables were encoded and numerical features were imputed, the model was trained and evaluated on separate training and testing sets. Key performance metrics, including MSE, RMSE, MAE, and R^2 score, demonstrated the model's ability to effectively capture complex patterns in the data, making it a powerful choice for this predictive task.

XGBoost Regressor: The XGBoost model was employed for car resale price prediction in our project due to its powerful ability to handle complex, non-linear relationships in the data. XGBoost, an ensemble method based on gradient boosting, builds decision trees sequentially, where each new tree attempts to correct the errors of the previous ones. This iterative approach enables the model to capture intricate patterns and interactions between variables such as car age, mileage, fuel type, and city of sale. By fine-tuning key hyperparameters like **n_estimators** (set to 100) and **learning_rate** (set to 0.1), the model efficiently balances bias and variance, leading to accurate predictions. XGBoost's robust performance, coupled with its ability to handle large and diverse datasets, makes it an ideal choice for predicting car resale prices, ensuring that sellers can set competitive prices and buyers can make informed purchasing decisions.

Sequential Neural Network: A Sequential Neural Network model was developed enabling the capturing of intricate relationships and complex nonlinear patterns in the data. The architecture of the model consists of three hidden layers with 128, 64, and 32 neurons, respectively, and also includes dropout layers to prevent overfitting and batch normalization for stable training. Categorical features were one-hot encoded, and numerical features were scaled to ensure proper distribution of input. The model is trained by using the Adam optimizer and Mean Squared Error as the loss function. Performing early stopping to stop training in case the validation loss stops improving reduces the risk of overfitting.

3.3 Evaluation

3.3.1 Evaluation Settings and Sampling

Evaluation settings and sampling strategies bear a very important role in understanding the performance of machine learning models, ensuring that their generalization to unseen data is appropriate. In the code, a train-test split strategy is implemented that splits the dataset into 80% for training and 20% for testing, which maintains a clear separation of data used for learning and evaluation. This is to ensure that the model is tested on data that it has not seen during its training phase, hence an unbiased estimate. Besides, for categorical variables, label encoding is performed, which converts them into numerical values while retaining their categorical nature. Such steps will ensure that the dataset is preprocessed and ready for meaningful evaluation.

3.3.2 Hyper-parameter Optimization

The XGBoost model in our pipeline incorporates key hyperparameters to ensure effective learning and accurate predictions. The n_estimators parameter is set to 100, specifying that the model will build 100 decision trees sequentially, with each tree correcting the errors of the previous ones. This strikes a balance between underfitting, which could result from too few trees, and overfitting, which might occur with too many trees. Additionally, the learning_rate is set to 0.1, a common choice for controlling the contribution of each tree to the final prediction. This gradual learning process helps the model generalize better and prevents overfitting, making it suitable for complex datasets.

Several hyper-parameter optimizations were performed for the SNN in the project of car resale price prediction in order to enhance accuracy and generalization. The architecture was designed with three hidden layers with 128, 64, and 32 neurons, respectively, ReLU for non-linear relationships, and a linear activation function for continuous output. Dropout layers with rates of 0.3 and 0.2 were included to reduce overfitting, while the Adam optimizer with a learning rate of 0.001 provided efficient and stable convergence.

Early stopping with patience of 3 epochs was also done to enhance the performance, which means training stops when the validation loss plateaus, usually at around 4–5 epochs. The batch size was set to 32 after experimentation in order to optimize memory usage and convergence speed. These hyper-parameter choices allowed the SNN to capture intricate patterns in the data while maintaining robust generalization, making it a highly effective model for car resale price prediction.

3.3.3 Evaluation

The evaluation of the models for car resale price prediction was conducted using metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R^2 Score. These metrics provide insights into the accuracy and generalization capabilities of each model.

Linear Regression:

As a baseline model, Linear Regression showed limited ability to capture complex relationships in the data. It produced the highest errors (MSE: 9.2416, RMSE: 3.0400, MAE: 2.0748) and a relatively low R^2 score of 0.6052. This highlights its limitations in modeling non-linear patterns and complex interactions between features.

Random Forest Regressor:

Random Forest significantly improved over Linear Regression, capturing non-linear relationships and reducing errors. It achieved an MSE of 3.5330, RMSE of 1.8796, and MAE of 0.9801, with an R^2 score of 0.8491. These results demonstrate its effectiveness in balancing accuracy and robustness while minimizing overfitting.

XGBoost Regressor:

XGBoost delivered competitive performance, with an MSE of 3.8056, an MAE of 0.9927, and an R² score of 0.8390. While slightly behind Random Forest, its iterative boosting process effectively captured non-linear relationships, making it a strong alternative for the task.

Sequential Neural Network (SNN):

The SNN, trained for 10 and 6 epochs, showed promising results with an MSE of 4.4649, RMSE of 2.1130, and MAE of 1.1795. It achieved an R² score of 0.8093, reflecting its ability to capture intricate non-linear patterns. However, its performance was slightly behind that of Random Forest and XGBoost, suggesting that further optimization may be needed for better generalization.

Summary of Results:

Model	MSE	RMSE	MAE	R2 Score
Linear Regression	9.2416	3.0400	2.0748	0.6052
Random Forest	3.5330	1.8796	0.9801	0.8491
XGBoost	3.8056	1.9515	0.9927	0.8390
Sequential NN	4.4649	2.1130	1.1795	0.8093

4. Results and Conclusion

The Random Forest Regressor outperformed all other models in this evaluation, establishing itself as the superior choice for capturing complex relationships within the data. Its ability to handle non-linear patterns and interactions between features is a key advantage, allowing it to achieve the lowest mean squared error (MSE) of 3.5330, a root mean squared error (RMSE) of 1.8796, and a mean absolute error (MAE) of 0.9801. The high R² score of 0.8491 further demonstrates its effectiveness in explaining the variance in the target variable, indicating that it provides a more accurate and reliable prediction compared to other models.

The strength of Random Forest lies in its ensemble approach, which combines the predictions of multiple decision trees to enhance robustness and reduce the risk of overfitting. This method not only improves accuracy but also allows the model to generalize better across different datasets. In contrast, simpler models like Linear Regression struggle to capture such complexities, resulting in higher error rates and lower predictive power. Overall, Random Forest's superior performance is attributed to its capability to model intricate relationships effectively while maintaining high levels of accuracy and reliability.