Used Car Price Prediction

Name:	**Ishanya**
Registration No./Roll No.:	21329
Institute/University Name:	IISER Bhopal
Program/Stream:	EECS
Problem Release date:	August 17 , 2023
Date of Submission:	November 19 , 2023

1 Introduction

New cars are typically bought close to their manufacturer's suggested retail price (MSRP), ensuring price transparency. But the used car market is complex. The goal is to identify key price-influencing factors and create a model that estimates used car prices. This benefits both buyers, to make informed decisions, and sellers, to strategically price their vehicles for maximum value or a quick sale.

2 Data Description

Training data: 5417 training instances and Test data: 602 test instances.

Categorical Features: Brand, Location, Transmission, Fuel Type, Owner Type.

Numeric Features: Year, Kilometers Driven, Mileage, Engine, Power, Seats.

Target variable: Selling Price (Numeric) of the Used Car

Problem Type: Regression because we are estimating real-valued outputs based on input features.

3 Exploratory Data Analysis and Visualisation

3.1 Loading the Data

Loaded the data provided by our professor. It consisted of training data, training data targets , and test data. Concatenated train and target data to visualize the data.

3.2 Data Cleaning

Units in 'Mileage,' 'Engine,' and 'Power' features were removed, converting the features data to pure numeric values. Null values were substituted for 0 values.

Both train and test datasets had missing values in the 'Mileage,' 'Engine,' 'Power,' and 'Seats' features. Thorough data cleaning addressed these null values, exploring imputation methods such as mean, mode, and KNN imputation. Given the discrete nature of the categories in 'Mileage,' 'Engine,' 'Power,' and 'Seats,' mode imputation was chosen for its simplicity and suitability. This approach helps maintain the original distribution and is robust to outliers.

The 'Brand' feature was split into new features: 'Brand1,' 'Model,' and 'Version.' During this process, it was observed that the test data included 'smart' and 'Ambassador' brands, which had no corresponding information in the train data. To maintain consistency, data instances associated with 'smart' and 'ambassador' were dropped. Additionally, it was verified that all data points in the test set had corresponding information in the train data.

3.3 Data Visualisation

The distribution of 'Selling Price' in the 'train' dataset is positively skewed (Skewness: 3.383963) with heavy tails (Kurtosis: 17.793260), suggesting the presence of outliers or extreme values.

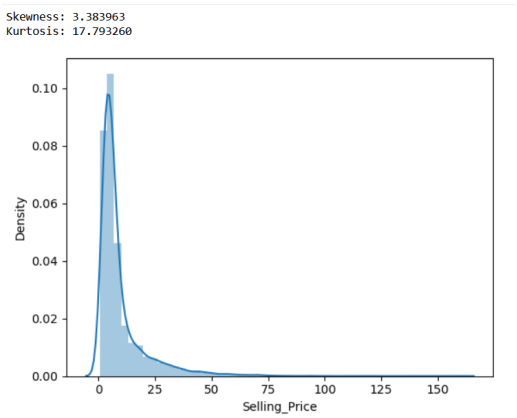

Figure 1: Distribution of 'Selling Price'

A new column 'Age' was created by subtracting the 'Year' column from 2020, assuming the newest used car was from 2019. Plots were generated to observe the frequency of cars based on 'Location,' 'Fuel Type,' 'Owner Type,' 'Year,' 'Brand1,' and 'Age.' Both the train and test data instances exhibited similar patterns in these plots. Also, made a heatmap for the Correlation Matrix.

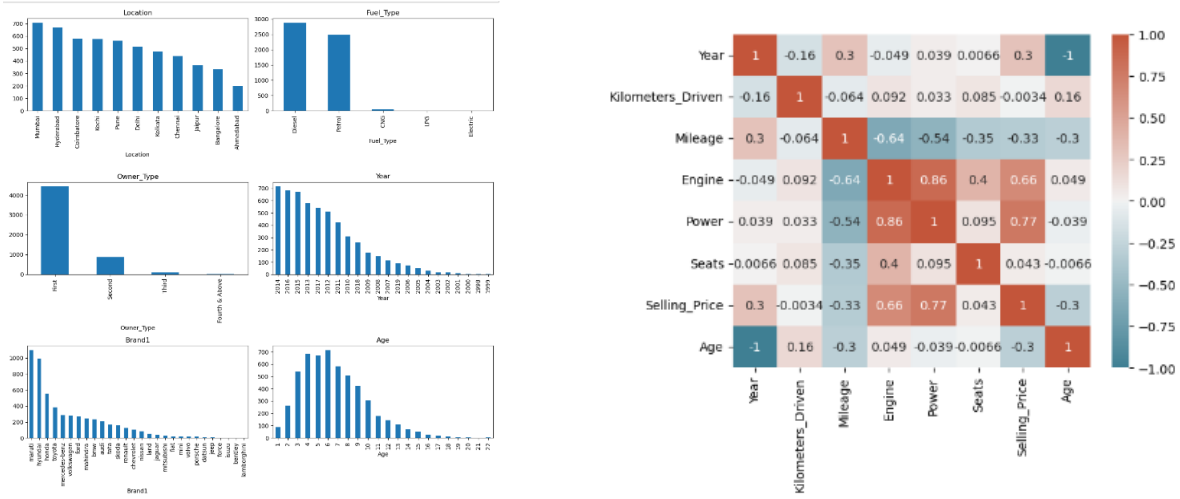

Figure 2: Features vs Car Count and Heatmap of Correlation Matrix

3.4 Encoding and Scaling

Experimented with both one-hot encoding and label encoding on all the categorical features, ultimately selecting one-hot encoding for the final model due to its superior performance in capturing categorical information and preventing ordinal assumptions.

Scaled the numeric data using RobustScaler from SciKit Learn to mitigate the impact of outliers and enhance the model's robustness.

3.5 Feature Selection Techniques

Experimented with feature selection techniques using Extra Trees Regressor to rank features and Mutual Information for regression techniques. The Extra Trees Regressor helped identify key variables, while Mutual Information enhanced predictive capabilities by capturing dependencies between features and the target variable.

4 Methods

Performed data splitting into training, testing, and validation sets with a ratio of 80:10:10, using the train test split function. This involved creating training sets (X_train_temp, y_train_temp) with 80% of the data, a test set (X_test, y_test) with 10%, and a validation set (X_val, y_val) with the remaining 10%. The process ensured a systematic partitioning of the dataset for effective model training, testing, and validation.

I have experimented with various techniques and I have explained it below.

For Type 1 analysis, performed mode imputation, feature ranking was conducted using Extra Trees Regressor, followed by hyperparameter tuning through GridSearchCV. The models employed include Linear Regression, Lasso Regression, Ridge Regression, Decision Tree, Random Forest, Gradient Boosting, XGBoost, SVR, KNN, and AdaBoost.

In Type 2 analysis, performed mode imputation, used label encoder, feature ranking was performed using Extra Trees Regressor. The models used in this type include Linear Regression, Lasso Regression, Gradient Boosting (GBR), XGBoost (XGB), K-Nearest Neighbors (KNN), and Random Forest.

For the Type 3 analysis, performed mode imputation, feature ranking was performed using Extra Trees Regressor for Linear Regression, Lasso Regression, Random Forest, Gradient Boosting, SVR, KNN, and XGBoost. The experiment involved not using the 'Location' feature. Subsequently, hyperparameter tuning was implemented for each model using GridSearchCV.

For Type 4 analysis, performed mode imputation, Mutual Information was employed for feature selection, and the models included were Linear Regression, Lasso Regression, Ridge Regression, Decision Tree, Random Forest, Gradient Boosting, and XGBoost. Additionally, hyperparameter tuning was implemented using GridSearchCV.

For the Type 5 analysis, KNN imputation was performed for handling missing values. Feature ranking using Extra Trees Regressor was conducted, followed by hyperparameter tuning through GridSearchCV. The models involved in this analysis include Linear Regression, Lasso Regression, Ridge Regression, Decision Tree, Random Forest, Gradient Boosting, XGBoost, SVR, KNN, and AdaBoost. Additionally, experimentation was carried out by retaining the location feature in the dataset.

5 Evaluation Criteria

The model evaluation involved optimizing hyperparameters via grid search, considering metrics such as Mean Squared Error, Mean Absolute Error, Root Mean Squared Error, and R^2 Score. These criteria served as benchmarks to assess the accuracy and fit of the regression models in predicting used car prices. Regression techniques were visualized through plots, providing insights into their predictive performance and alignment with actual used car prices.

Table 1: Performance of Type 1 combination

Classifier	MSE	MAE	RMSE	R^2 Score
Linear Regression	36.14	3.75	6.01	0.71
Lasso Regression	41.50	3.90	6.44	0.67
Random Forest	10.71	1.61	3.27	0.91
Gradient Boosting	10.39	1.57	3.22	0.92
SVR	16.71	1.98	4.09	0.87
KNN	13.38	1.91	3.66	0.89
XGBoost	9.21	1.59	3.03	0.93
Decision Tree	21.30	2.04	4.62	0.83
Ridge Regression	36.18	3.75	6.02	0.71
AdaBoost	26.24	3.11	5.12	0.79

Table 2: Performance of Type 2 combination

Model	MSE	MAE	RMSE	R^2 Score
Linear Regression	35.86	3.63	5.99	0.6921
Random Forest Regression	10.94	1.43	3.31	0.9061
Ridge Regression	35.87	3.63	5.99	0.6921
Gradient Boosting Regression	10.2	1.35	3.19	0.9124
Lasso	38.49	3.62	6.2	0.6696
XGBoost	7.56	1.21	2.75	0.9351
K-Nearest Neighbors	13.27	1.57	3.64	0.886

Table 3: Performance of Type 3 combination

Classifier	MSE	MAE	RMSE	R^2 Score
Linear Regression	36.14	3.75	6.01	0.71
Lasso Regression	41.50	3.90	6.44	0.67
Random Forest	11.01	1.60	3.32	0.91
Gradient Boosting	10.39	1.57	3.22	0.92
SVR	16.71	1.98	4.09	0.87
KNN	13.38	1.91	3.66	0.89
XGBoost	9.21	1.59	3.03	0.93

Table 4: Performance of Type 4 combination

Model	MSE	MAE	RMSE	R^2 Score
Linear Regression	38.89	3.73	6.24	0.6905
Lasso Regression	38.83	3.73	6.23	0.6909
Ridge Regression	38.87	3.72	6.23	0.6906
Decision Tree	28.64	2.08	5.35	0.7720
Random Forest	17.88	1.76	4.23	0.8577
Gradient Boosting	16.89	1.73	4.11	0.8655
XGBoost	16.10	1.73	4.01	0.8718

Table 5: Performance of Type 5 combination-With KNN Imputer

Model	MSE	MAE	RMSE	R^2 Score
Linear Regression	35.28	3.67	5.94	0.7182
Lasso Regression	41.15	3.88	6.41	0.6712
Ridge Regression	35.56	3.65	5.96	0.7159
Random Forest	14.62	1.61	3.82	0.8832
Gradient Boosting	14.33	1.51	3.79	0.8855
AdaBoost	27.29	3.10	5.22	0.7819
SVR	16.43	1.85	4.05	0.8687
KNN	18.01	2.04	4.24	0.8561
Decision Tree	20.61	2.08	4.54	0.8353
XGBoost	10.04	1.44	3.17	0.9198

6 Discussions and Conclusions

XGBoost, Random Forest, and Gradient Boosting emerge as top-performing models across various combinations and regression techniques, consistently demonstrating superior predictive accuracy with low error metrics and high R^2 scores. Their robustness and ability to handle complex relationships

make them standout choices for optimal model performance. The final test data was run XGBoost, Random Forest, and Gradient Boosting to predict the test data used car prices.

References

1. https://cs229.stanford.edu/proj2019aut/data/assignment_308832_raw/26612934.pdf
2. https://ieeexplore.ieee.org/document/9696839
3. https://scikit-learn.org/stable/modules/linear_model.html