

# CYO - Used Vehicle Price Predictor

A. Masmela

2023-10-21

## Contents

<b>Section 1: Project Overview:</b>	<b>2</b>
<b>Section 2: Methods and Techniques</b>	<b>3</b>
2.1. Data Cleaning . . . . .	3
2.2. Data Exploration . . . . .	4
2.3. Model Development . . . . .	5
2.3.1. Benchmark Model . . . . .	5
2.3.2. Linear Regression Model . . . . .	5
2.3.3. Linear Regression Model: Corolla Cars Only . . . . .	6
2.3.4. Machine Learning model: kNN . . . . .	6
2.3.5. Machine Learning model: Random Forest . . . . .	7
2.3.6. Machine Learning model: LightGBM . . . . .	7
2.3.7. Machine Learning model: Xgboost . . . . .	8
<b>Section 3: Results of Model Performance</b>	<b>8</b>
3.1. RMSE . . . . .	9
3.2. Distribution of Residuals . . . . .	9
3.3. Scatter plot of Actuals vs Predicted prices . . . . .	10
3.4. KS statistics and CDF plot . . . . .	11
3.5. Xgboost model validation . . . . .	14
<b>Section 4: Conclusion and Future Work</b>	<b>14</b>
<b>Section 5: References</b>	<b>15</b>

## Section 1: Project Overview:

This project aims to develop a predictive model for used-vehicle prices using historical data from Pakistan. The model is trained and tested with historical values and then evaluated using an ‘out-of-time’ sample. To accomplish this, we utilized the ‘Used Car Price Prediction Pakistan Dataset’ obtained from Kaggle.com. This dataset, with 86,120 data points, was extracted from the popular automotive marketplace website, <https://www.pakwheels.com>. It contains insights into the world of automobiles and features eight distinct attributes.[1]

### Dataset Details:

As mentioned, this dataset contains 86,120 data points, each representing a unique vehicle. The dataset contains eight features, including the target variable, price, and seven independent variables, including make, model, city, year, mileage, engine displacement, and battery capacity. The dataset’s features are described below:

- (a) Make: Name of the manufacturer
- (b) Model: Model of the vehicle
- (c) City: Location where the vehicle was sold
- (d) Year: Year of production
- (e) Mileage: Number of kilometers driven by the vehicle
- (f) Engine Displacement: Engine displacement in cubic centimeters (cc)
- (g) Battery: Capacity of the battery in kilowatt-hours (for electric vehicles)
- (h) Price (Rs): Price of the car in Pakistani Rupees (PKR)

To facilitate the analysis, the project converts the original prices in Pakistani Rupees into US Dollars as of September 2023, using an exchange rate of 0.0034 US dollars per Pakistani Rupee.[2]

**Data Preparation:** As further detailed in section 2.1, the dataset was cleaned, and two subsets were created: a model development dataset comprising data points until the end of 2020 and an ‘out-of-time’ model validation dataset containing data points from 2021 to 2023. As Machine Learning algorithms were utilized in the project, the data frame was prepared, imputing missing values, and removing outliers and vehicle observations with small sample sizes.

**Model Development:** The predictive models developed in this project are designed to predict used-vehicle prices. The out-of-time dataset will simulate the scenario where historical data is available until 2020, and we will evaluate the model’s effectiveness in predicting prices up to 2023. Model predicted values are compared with actual values in section 3, assessing the model’s predictive power against the ‘out-of-time’ data set.

This project utilizes a combination of linear regression techniques and a suite of powerful machine learning algorithms to predict the used-vehicle prices. Random Forest (RF), Light Gradient Boosting Machine (LGBM), and XGBoost (XGB) are the machine learning algorithms chosen for this project.

**Model Evaluation:** As described in detail in section 3, the performance of each model is assessed using several metrics, including:

- (a) Root Mean Square Error (RMSE)
- (b) Distribution of Residuals
- (c) Scatter Plot of Actual vs. Predicted Values
- (d) Kolmogorov-Smirnov (KS) Statistic

**Conclusion:** The project reveals that machine learning algorithms substantially outperform linear regression techniques in the development of this particular predictive model. However, it's important to note that the intrinsic characteristics of vehicles, such as the make, model, year, and engine displacement, explain only a portion of the price of an used vehicle. Additional macroeconomic factors, such as inflation, cost of borrowing, vehicle inventory levels, and commerce policies, play a significant role in shaping prices in the used-vehicle market.

## Section 2: Methods and Techniques

In this section, we provide a comprehensive overview of the methodologies employed in our project. It is divided into three key subsections:

**2.1. Data Cleaning and Transformation:** This initial subsection deep dives into the data cleaning and transformation processes. Here, we outline the steps taken to prepare the dataset for modeling, including imputing missing values, filtering and removing outliers, and other essential preprocessing procedures.

**2.2. Data Exploration:** In this subsection, we conduct a thorough data exploration, where we analyze the distribution of the data by some of the most relevant features. This exploration offers insights into the dataset's characteristics and provides valuable context for our modeling decisions.

**2.3. Modeling Techniques:** This last subsection is dedicated to describing the various modeling techniques we leveraged. We detail the modeling approaches, data transformations, and other techniques used to build and optimize our predictive models. Additionally, we share our initial observations regarding computational effort needed in each approach.

The forthcoming section 3 will provide a comprehensive discussion of the predictive performance of the models, offering a robust evaluation of their effectiveness.

### 2.1. Data Cleaning

In addition to converting the target variable (Price) into a more universally recognizable currency (US dollars), we implemented a series of data cleaning techniques:

- (a) **Feature Selection:** We scrutinized the dataset to identify and retain the features with the fewest missing values, ignoring features such as Battery. This strategic choice aimed to ensure the integrity and completeness of the dataset for our analysis.
- (b) **Missing Value Imputation:** To address missing values with features such as "Engine Displacement", we employed a straightforward yet effective imputation method, using the average value of all other observations in this feature. This approach allowed us to maintain the dataset's continuity and statistical robustness, while keeping an useful predictor.
- (c) **Make Filtering:** We streamlined the dataset by eliminating vehicle makes with small sample set, i.e., fewer than one hundred data points. This curation enhances the overall quality and representativeness of our dataset by focusing on brands with more significant market presence.
- (d) **Outlier Removal:** To enhance the model's resilience to noise and outliers, we leveraged the z-score method with a threshold of three to identify and remove extreme values within the "Price" feature. This technique contributes to a more accurate and reliable model by mitigating the influence of outlier data points.
- (e) **Negative Price Data Points:** We screened the dataset for data points with negative prices, recognizing that such entries are anomalous and not representative of the real-world used-vehicle market. These data points were removed to maintain data integrity and model performance.

By implementing these data preprocessing steps, we optimized the dataset for our analysis, ensuring that our predictive models are built on a solid foundation of clean, complete, and representative data.

## 2.2. Data Exploration

After cleaning the data, we conducted a comprehensive data exploration, discovering the following insights:

- Distribution by Make:** Toyota, Suzuki, and Honda emerged as the dominant vehicle manufacturers, each manufacturer contributing around 20,000 data points to the dataset.
- Distribution by Model:** The dataset prominently features the Corolla, Civic, and City as the most prevalent car models, with a presence of nearly 15,000, 10,000, and 6,000 vehicles, respectively.
- Distribution by City and Average Price by City:** Among the cities, Lahore, Karachi, and Islamabad emerged as the major hubs for car sales, each hosting over 10,000 transactions. While analyzing the average prices by city, we found that prices are relatively consistent across all urban centers.

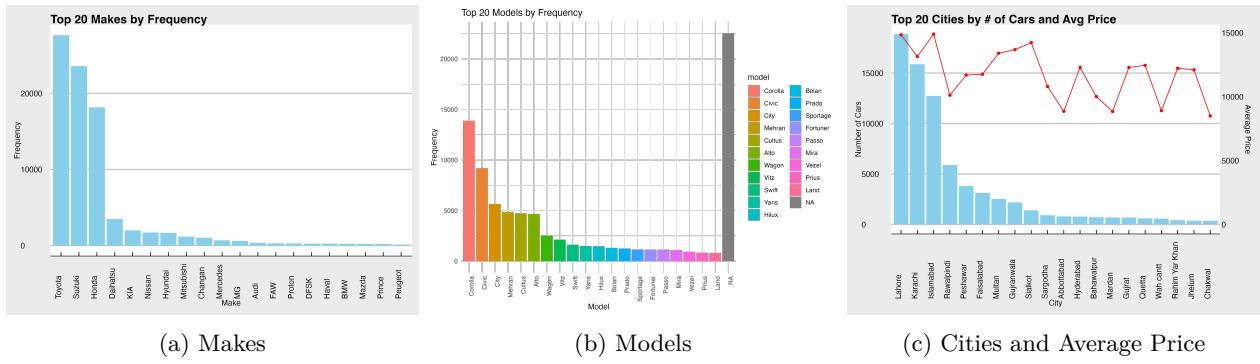
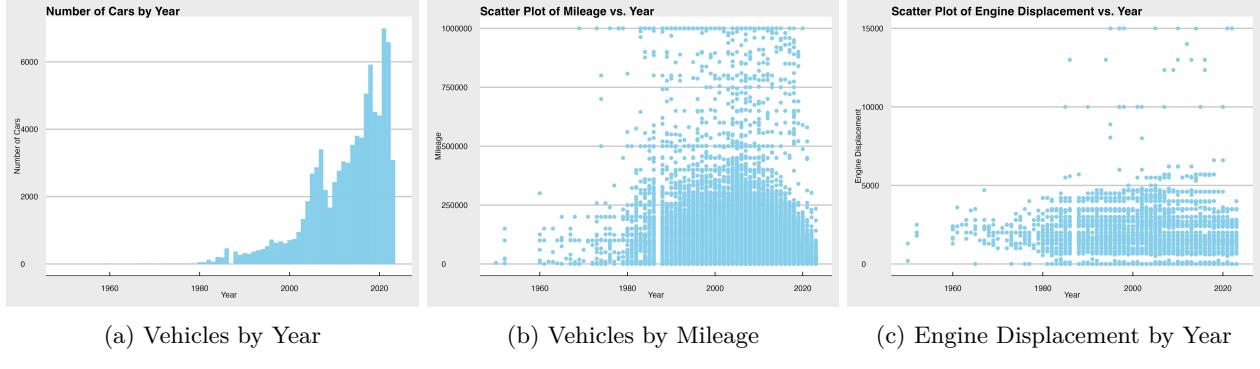


Figure 1: Dataset Distributions

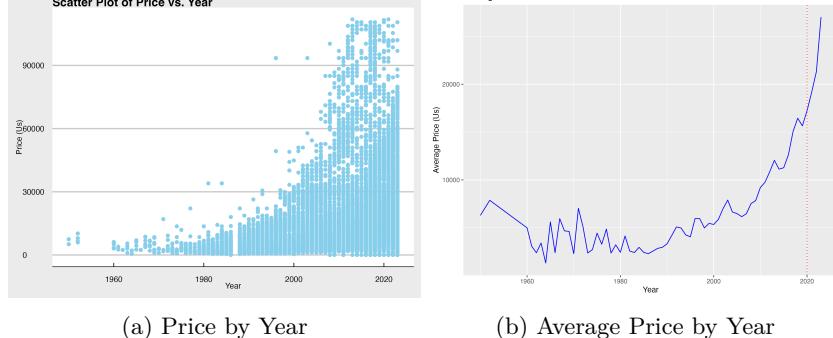
- Distribution of Cars by Year:** A notable trend is the substantial increase in the number of used vehicles over time, with fewer observations in the early '80s and a peak of 6,000 vehicles per model year in the 2020s. Interestingly, economic contractions and high unemployment rates seem to have left their mark, with years like 2008 and 2020 displaying particularly low observations.
- Distribution of Cars by Mileage:** Vehicles in the dataset typically have mileage ranging from 25,000 to 50,000 kilometers, with more recent models exhibiting lower mileage, as expected. However, outliers at the high end of the distribution reveal mileage exceeding 50,000 and even 100,000 kilometers, even in models as recent as 2019.
- Distribution of Cars by Engine Displacement vs. Year:** Across model years, engine power remains relatively stable, with engine capacities consistently below 5,000 cc for all models. While there are rare outliers reaching as high as 15,000 cc, most vehicles adhere to this range.
- Distribution of Price:** As anticipated, newer vehicles have higher prices. Nonetheless, some vehicles from recent years exhibit surprisingly low prices, even just a few hundred dollars. This phenomenon could be attributed to total-loss or reposessed vehicles, or those with significant wear-and-tear. The majority of vehicles fall below the \$30,000 price range, resulting in a modest overall average price of approximately \$10,000. Nonetheless, there are a few exceptional outliers since 2010, surpassing \$90,000 in price.
- Average Price per Year:** The average price trend exhibits an upward trajectory since the '80s. However, a surge occurs after the economic recovery following the 2008 Financial Crisis. In this period, average prices experienced a substantial rise, increasing from around an average of \$7,000 in



(a) Vehicles by Year      (b) Vehicles by Mileage      (c) Engine Displacement by Year

Figure 2: Distributions by Year

2012 to an average of approximately \$17,000 in 2019. The world pandemic caused a temporary dip in average prices, but as consumer purchasing power rebounded, and production and inventory levels contracted, average prices soared, nearing almost \$23,000 by the end of 2023.



(a) Price by Year      (b) Average Price by Year

Figure 3: Price Distributions

## 2.3. Model Development

### 2.3.1. Benchmark Model

Following our data exploration, we established a simple benchmark model to serve as a baseline for evaluating all developed models. This naïve benchmark assumes that all vehicles have the average vehicle price of approximately \$10,000. While this is a simplistic assumption, it provides a reference point against which all developed models in our project should demonstrate superior performance.

The results of this benchmark model, alongside those of each developed model, will be discussed in detail in section 3.

### 2.3.2. Linear Regression Model

Our initial modeling approach involved the use of traditional linear regression. In this linear regression model, we focused on numerical variables, as linear models are effective at handling this type of variables. In these models, predictions are made by calculating the numerical relationship between input features and target variables, allowing for a quantification of the impact of input features on particular changes on the

target variable. However, linear models tend to have challenges with non-numerical data, such as categorical variables, as they lack a direct numerical relationship. [yale-website]

Our initial step consisted of building a correlation matrix among these numerical variables. The correlation matrix revealed significant insights, most notably, a positive correlation between engine displacement and price (0.49). Furthermore, we observed a slightly less substantial correlation between the year and price (0.34). These two features, i.e., engine displacement and year, displayed weak correlations with each other, indicating their independence and their potential to contribute distinct value to the model. Lastly, mileage exhibited a modest negative correlation with price (-0.12) by itself and displayed a negative correlation with the year (-0.26). Therefore, mileage can be valuable in differentiating vehicles with similar car model, engine displacement and year, serving as a relevant additional feature in the model.

X	Price_Us	year	mileage	Engine_displacement
Price_Us	1.0000000	0.3444260	-0.1282792	0.4932434
year	0.3444260	1.0000000	-0.2683262	-0.1381026
mileage	-0.1282792	-0.2683262	1.0000000	0.0773300
Engine_displacement	0.4932434	-0.1381026	0.0773300	1.0000000

### 2.3.3. Linear Regression Model: Corolla Cars Only

In our initial linear regression model, we did not include specific car models as a factor. However, upon segmenting the data to focus solely on the most prevalent vehicle model, the Corolla, we observed a shift in the correlation dynamics, with substantially strengthened relationships. The correlation between the year and price exhibited the highest correlation coefficient of 0.79, while a negative correlation of -0.37 was observed between mileage and price. In this subsegment, year and mileage displayed a relatively strong correlation with each other, as older vehicles tend to accumulate more mileage. Furthermore, due to the inherent similarity in engine displacement among Corolla models, the correlation between price and engine displacement exhibited a weaker relationship, with a correlation coefficient of 0.14.

X	Price_Us	year	mileage	Engine_displacement
Price_Us	1.0000000	0.7934767	-0.3715621	0.1494276
year	0.7934767	1.0000000	-0.2861786	-0.0173860
mileage	-0.3715621	-0.2861786	1.0000000	0.0084351
Engine_displacement	0.1494276	-0.0173860	0.0084351	1.0000000

### 2.3.4. Machine Learning model: kNN

Later in the project, we incorporated machine learning algorithms. Our initial choice was k-Nearest Neighbors (kNN), a versatile algorithm primarily known as a classification algorithm, where it assigns a class label to a new data point based on the majority class among its k-nearest neighbors in the training dataset. However, kNN can also be used for regression tasks, where it predicts continuous variables.[3]

In the context of this project, we leveraged kNN for predictive modeling. To prepare the data for kNN, we carried out several critical transformations. Firstly, we addressed missing values by imputing them with the respective feature's average values. Moreover, we applied label encoding to text-based features, converting them into integers, which enabled kNN to use them as predictors. This preprocessing step was particularly essential for categorical variables, ensuring the algorithm's compatibility. Additionally, in our kNN model, we opted for a k-value of five, indicating the number of nearest neighbors considered for predictions.

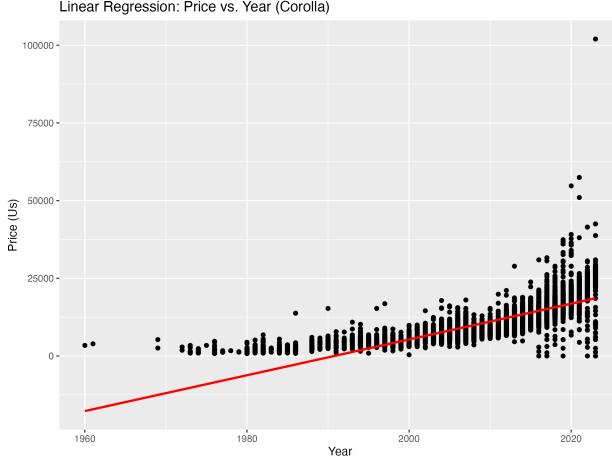


Figure 4: Linear Regression: Corolla Price vs. Year

### 2.3.5. Machine Learning model: Random Forest

We also harnessed more robust ML algorithms tailored for regression scenarios. We initiated this phase with Random Forest (RF), an ensemble learning method renowned for its adaptability in handling both classification and regression tasks. RF, as a collection of decision trees, excels at capturing complex patterns in the data.[4] One notable advantage of RF is its compatibility with categorical variables, making it suitable for our specific use case, where “make” and “model” were vital predictors. In our data preparation process, apart from addressing missing values through imputation, we employed a practical strategy for handling categorical features. Specifically, we grouped together sub-categories with low frequencies, which strengthened the robustness of the RF algorithm, allowing it to focus on materially predominant sub-categories, enhancing its ability to make accurate predictions.

### 2.3.6. Machine Learning model: LightGBM

Furthermore, we integrated the Light GBM (LGBM) algorithm into our machine learning model. LGBM is a powerful gradient boosting framework that boasts exceptional speed and efficiency, making it well-suited for a range of machine learning applications.[5] In preparation for LGBM, we converted our data frames into matrices, a requisite step that not only optimizes computational and memory efficiency but also facilitates parallel processing, as matrices can be effortlessly distributed for parallel computation.

LGBM provides numerous adjustable parameters for various use-cases. One of these parameters enables us to define the model as a regression model, thus distinguishing it from classification tasks, such as those typically associated with other algorithms such as k-Nearest Neighbors (kNN). To ensure an equitable comparison with our other models, we employed the Root Mean Square Error (RMSE) as the evaluation metric, a choice that accentuates the model’s performance by penalizing the influence of outliers. Our strategic selection of a learning rate of 0.1 served to mitigate overshooting issues, a common concern in gradient boosting models. Maintaining a relatively shallow depth of six levels further guarded against overfitting. Lastly, we opted for one hundred boosting rounds, permitting the algorithm to iteratively correct errors from preceding trees. Although a high number of boosting rounds can extend the training time, it’s important to note that the impact on training time was relatively minimal due to the dataset’s manageable size, consisting of approximately 86,000 observations. This parameter configuration empowered LGBM to deliver robust and efficient regression predictions.

### 2.3.7. Machine Learning model: Xgboost

Lastly, we incorporated Xgboost, a highly efficient gradient boosting algorithm known for its speed and accuracy.[6] [7] Both Light GBM and Xgboost demonstrated remarkable efficiency, requiring minimal computational time due to the dataset's manageable size of approximately 86,000 observations. As detailed in section 5, our machine-learning-based models produced impressive results.

In addition to the data transformations performed for Light GBM, which involved converting data frames into matrices and handling categorical variables, we followed a similar preprocessing approach for Xgboost. However, in the case of Xgboost, we retained the target variable within the data frame, eliminating the need for extensive feature reordering prior to algorithm execution. Given the regression task at hand, we specified the objective as “reg:squarederror,” instructing the model to minimize the squared error loss during training. Leveraging the dataset’s manageable scale, we comfortably set the number of boosting rounds to five hundred without demanding a significant computational power. To safeguard against overfitting, we constrained the depth of each tree to six levels. Lastly, we opted for a relatively low learning rate for the boosting rounds, ensuring that the model learned progressively and effectively. As noted below, Xgboost have an exponential improvement in RMSE as the number of boosting rounds increases.

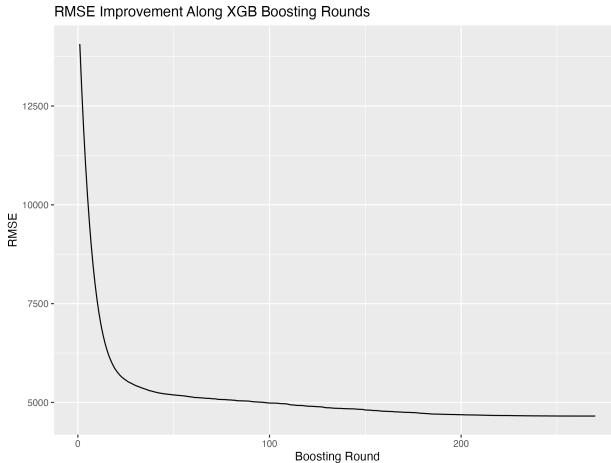


Figure 5: Xgboost: RMSE Improvement by boosting round

One of the key elements for algorithms is explaining the importance of each feature in the model. In the case of Xgboost, we can extract the feature importance from the model. The following table shows the top features in the model, with the engine displacement being the feature yielding the highest gain.

Feature	Gain	Cover	Frequency
Engine_displacement	0.5292714	0.2373571	0.2371925
year	0.3408893	0.0820317	0.1754457
mileage	0.0799039	0.2595829	0.2268111
X	0.0499354	0.4210283	0.3605507

## Section 3: Results of Model Performance

To assess the performance of each model, we employed four evaluation methodologies: Root Mean Square Error (RMSE), the Distribution of Residuals, a graphical representation through Scatter Plots illustrating the differences between actual and predicted prices, and the Kolmogorov-Smirnov (KS) statistic to compare the cumulative distribution functions (CDFs) of actual and predicted prices.

### 3.1. RMSE

In the context of RMSE evaluation, the benchmark model, which simply averages all observations' prices, yielded an RMSE of USD\$11,304.45. Consequently, our models were expected to achieve lower RMSE values to demonstrate their superior performance.

Both linear regression models exhibited promising results, with an RMSE of USD\$8,492.50 when considering all car models and USD\$4,852.83 when focusing on one of the top-selling models, the Corolla. Encouraged by the performance of the segmented linear regression model, we sought to create a simplified segmentation strategy based on the correlation between the model year and price. However, instead of running three separate models for each segment, we incorporated the clusters as an additional feature to evaluate whether it enhanced the linear model's performance. Unfortunately, the RMSE of this new predictive model remained similar to that of the initial linear model, resulting in an RMSE of USD\$8,391.56. Aside from the unsatisfactory outcomes, this code displayed high instability, leading us to exclude it from our R script.

Subsequently, as we transitioned into machine learning algorithms, we began using k-Nearest Neighbors (kNN), an algorithm not inherently designed for continuous variable prediction. As anticipated, the RMSE of this model failed to surpass the benchmark, registering an RMSE of USD\$14,892.31. We later shifted to machine learning algorithms explicitly designed for regression tasks. Our first choice was the Random Forest (RF) algorithm, which allowed us to incorporate more predictive categorical variables, such as the vehicle model. The RF model delivered an RMSE of USD\$5,206.83. Despite its effectiveness, it demanded more computational resources, which was notable even with our relatively small dataset.

The Light GBM model provided a comparable RMSE of USD\$6,299.64 while consuming significantly less computational time. Finally, the Xgboost model demonstrated remarkable predictive power, achieving a similarly low RMSE of USD\$5,462.64, all while executing five hundred boosting rounds with remarkable efficiency. It's essential to emphasize that this efficient performance was feasible due to our dataset's modest size, allowing for relatively low computational time.

Model	RMSE
Benchmark	11304.452
Linear Regression	8492.500
kNN	14892.442
Random Forest	5206.832
LightGBM	6299.642
XGBoost	5462.640

### 3.2. Distribution of Residuals

The distribution of residuals is a critical aspect, representing the disparity between actual and predicted values. In an accurate model, one expects the residuals to form a tight normal distribution centered around zero. The subsequent plots illustrate the distribution of residuals for each of the models developed.

In the benchmark model, the distribution of residuals does not conform to a normal distribution and exhibits a significant right skew, indicating substantial underprediction in high-priced instances.

The distribution of residuals in the first linear regression model is predominantly centered around zero, although it features thin tails on both ends and a slight skew towards positive residuals. A similar slight skew towards positive residuals is observed in the linear regression model focused solely on Corolla vehicles. The tighter distribution around zero in the latter model might be a reflection of the generally lower prices associated with this particular car model.

The distribution of residuals in the k-Nearest Neighbors (kNN) model, which is not well-suited for this type of prediction, displays a pronounced skew towards negative residuals, indicating substantial underprediction of prices. Furthermore, this distribution does not conform to the normal distribution, highlighting that the kNN algorithm and approach are inappropriate for this use-case.

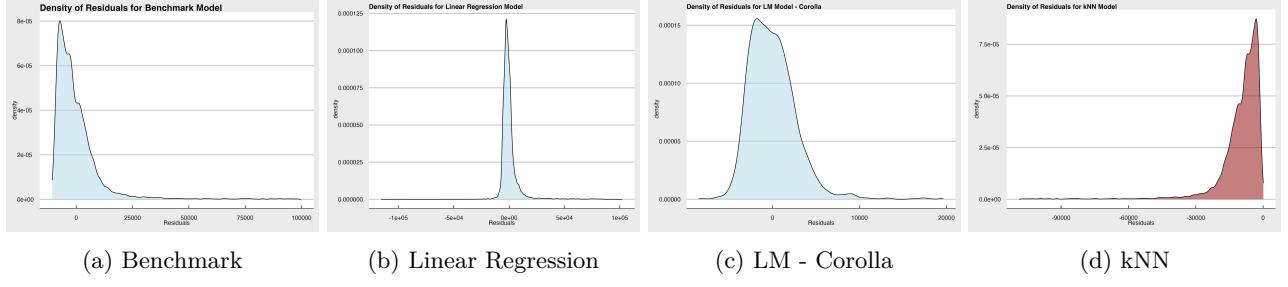


Figure 6: Density of Residuals

In contrast, the appropriate machine learning algorithms for this use-case, namely Random Forest (RF), Light GBM (LGBM), and XGBoost (XGB), all yield tight, normal distributions centered around zero with similarly proportioned tails on both ends, highlighting their superior performance for this task.

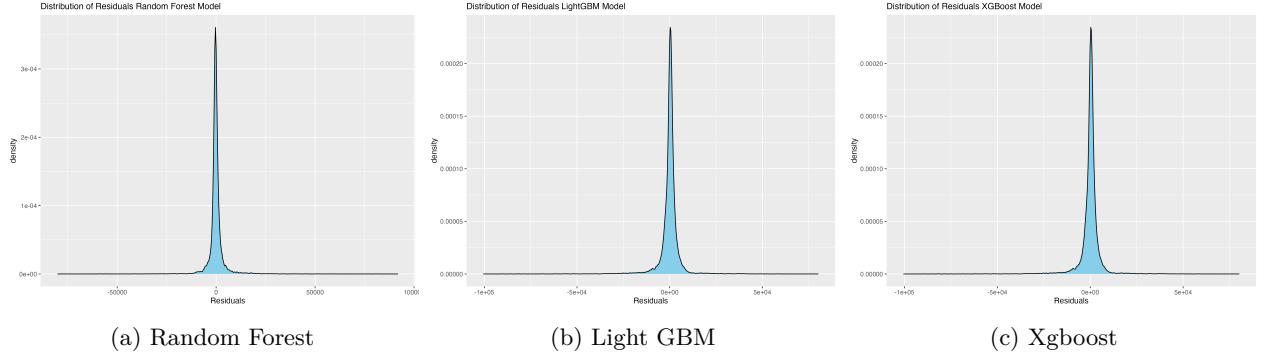


Figure 7: Density of Residuals

### 3.3. Scatter plot of Actuals vs Predicted prices

The scatter plot illustrating Actual vs. Predicted prices provides an intuitive visualization of the disparity between predictions and actual values. A perfect model would have all predicted and actual values falling into a diagonal. In the case of the benchmark model, which simply averages all observations, the scatter plot appears as a flat horizontal line across the actual values, as shown below.

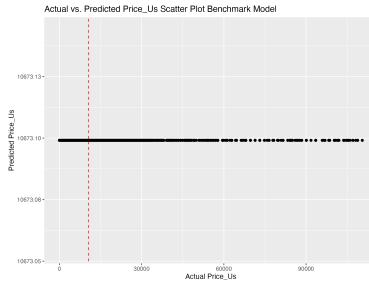


Figure 8: Scatter Plot Predicted vs. Actuals: Benchmark

As illustrated in the following plots, the linear regression model including all car models exhibits data points falling around the diagonal line for lower prices, i.e., less than USD\$30,000. However, it tends to underpredict higher prices, which are all predicted below USD\$50,000. When filtering for Corolla car models, the regression model shows improved performance, yet it continues predicting some prices as negative.

The initial machine learning model, kNN, designed for classification rather than continuous variables, displays the weakest performance, consistently underpredicting prices by a substantial magnitude. This underprediction grows exponentially as prices exceed USD\$30,000. This model's poor performance was also evident in the distribution of residuals, which is skewed towards negative residuals, as previously discussed.

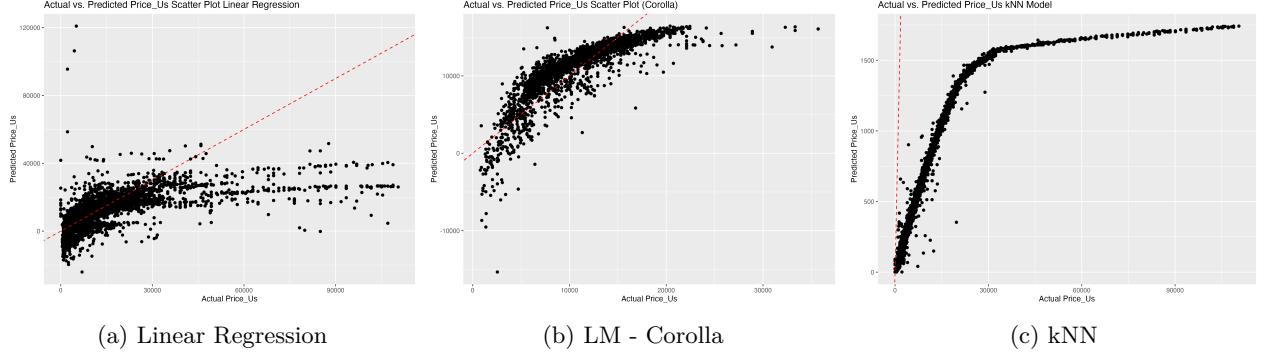


Figure 9: Scatter Plot Predicted vs. Actuals

On the other hand, both the Random Forest (RF) and Light GBM (LGBM) models yield satisfactory scatter distributions of predicted vs. actual values, with data points clustered around the diagonal. RF appears to outperform LGBM, with predictions evenly distributed on both sides of the diagonal. LGBM tends to exhibit more underpredictions in the high-end price range. However, the enhanced performance of RF may not offset the additional computational time required for model training.

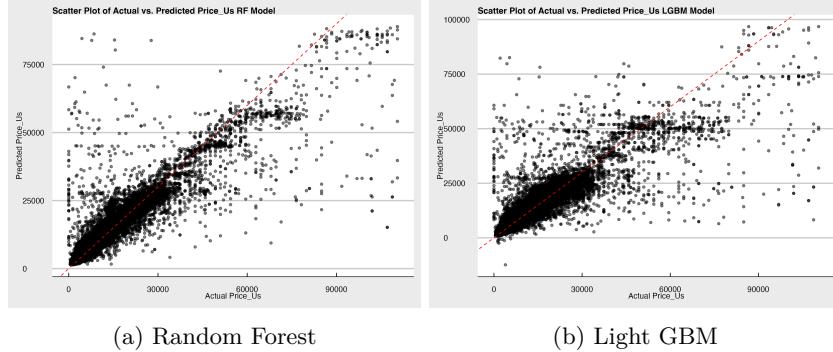


Figure 10: Scatter Plot Predicted vs. Actuals

Lastly, for the Xgboost model, we color-coded overpredictions in blue and underpredictions in green for clarity. This model's performance is deemed satisfactory. The color-coding aids in understanding the model's predictive power. Ultimately, each observation represents either a gain or a loss for the asset owner. The Xgboost model exhibits a similar distribution around the diagonal for accurate values, displaying good predictive ability for vehicles priced up to USD\$60,000. Similar to the LGBM model, the Xgboost model tends to underpredict vehicles with high prices. The underprediction in the high-end price range is also evident in the distribution of residuals, which is skewed towards negative residuals, as previously discussed. Therefore, further development of the modelthe high-end price range is warranted to identify additional factors that influence price in this range.

### 3.4. KS statistics and CDF plot

The KS (Kolmogorov-Smirnov) statistic is a test that compares the cumulative distribution functions (CDFs) of two datasets. [8] In our analysis, we leveraged the KS statistic to compare the CDF of actual prices with the CDF of predicted prices. The KS metric quantifies the maximum vertical distance between the



Figure 11: Scatter Plot Predicted vs. Actuals: Xgboost

cumulative distribution functions (CDFs) of these two distributions, providing valuable insights into the model’s performance.

The cumulative distribution function (CDF) of a dataset is a mathematical function that describes the probability that a random outcome variable takes on a value less than or equal to a specific value.[9] [10] In the context of our analysis, the CDF of actual prices shows how the prices are distributed in the dataset, and the CDF of predicted prices shows how the model’s predictions are distributed. The KS statistic calculates the maximum vertical distance between these two CDFs, helping us understand how closely the model’s predictions match the actual price distribution.

As mentioned above, in our use-case, we leveraged the KS statistic to assess how well our model’s predictions align with the actual data distribution. A lower KS value indicates a better fit, suggesting that the model’s predicted values closely resemble the actual values. As illustrated in the table below, the benchmark model, which simply averages all observations, yielded a KS of 0.54, indicating a poor fit. In contrast, the linear regression model, which considers all car models, achieved a KS of 0.17, indicating a better fit. The linear regression model focused solely on Corolla car models yielded a similar KS of 0.18. The kNN model, which is not well-suited for this type of prediction, yielded a KS of 0.96, indicating a poor fit. The machine learning algorithms, Random Forest (RF), Light GBM (LGBM), and Xgboost (XGB), all yielded superior KS values of 0.04, indicating a strong fit.

Model	KS_Statistic
Benchmark	0.6334541
Linear Regression	0.1771010
kNN	0.9628945
Random Forest	0.0465649
LightGBM	0.0471183
XGBoost	0.0495972

To visualize this statistic, we created CDF plots that display the predicted prices’ cumulative distribution alongside the actual prices’ CDF. In an ideal scenario, a well-fitting model would have its CDF distribution closely following the CDF distribution of actual prices.

The benchmark model’s CDF shows imperfections, with predicted values, i.e., average price, forming a vertical line noticeably distant from the CDF distribution of actual prices, therefore, resulting in a relatively high KS of 0.54.

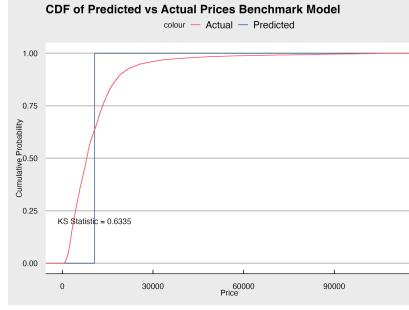
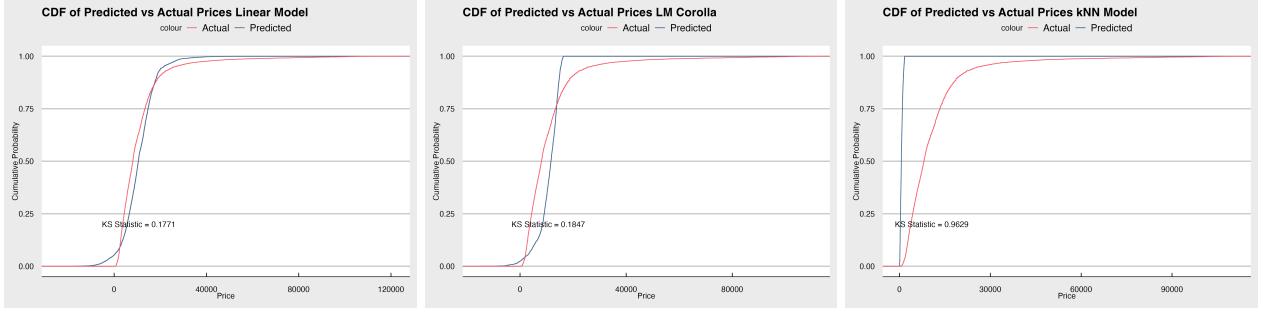


Figure 12: KS (Kolmogorov-Smirnov) statistic: Benchmark

The linear regression model significantly improves this fit as evident in the closely matched CDF distributions of predicted and actual prices. This improvement is also observed in the linear model specifically for Corolla car models. As mentioned, these models yielded similar KS values of 0.17 and 0.18, respectively.

As mentioned, the kNN model exhibits the worst KS at 0.96, clearly illustrated by the considerable discrepancy between CDF distributions. This result is consistent with the model's poor performance in the scatter plot of actual vs. predicted prices and the distribution of residuals, as previously discussed.



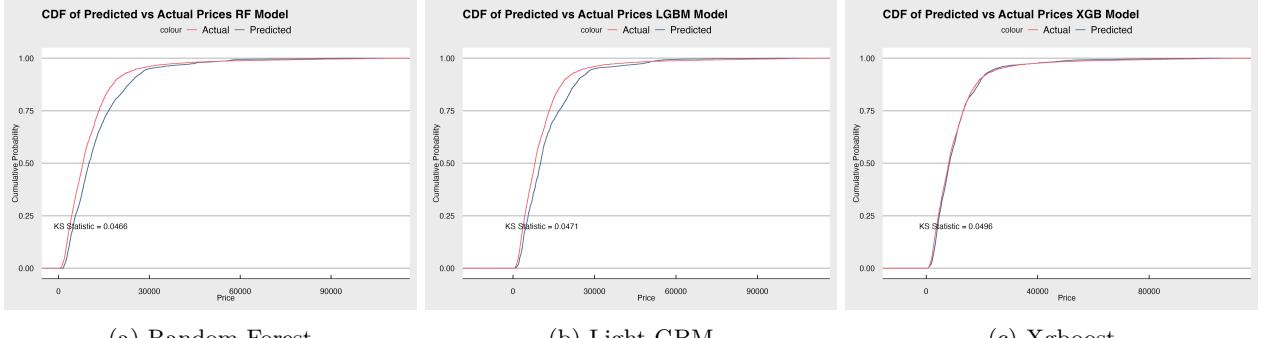
(a) Linear Model

(b) LM - Corolla

(c) kNN

Figure 13: KS (Kolmogorov-Smirnov) statistic

In contrast, machine learning algorithms designed for regression and continuous variable prediction, such as RF, LGBM, and Xgboost, deliver superior KS values of 0.04, as previously noted. Their CDF distributions nearly overlap the CDF distribution of actual prices, with Xgboost's distribution being almost identical, indicating their strong predictive power. This result is consistent with the models' performance in the scatter plot of actual vs. predicted prices and the distribution of residuals, as previously noted.



(a) Random Forest

(b) Light GBM

(c) Xgboost

Figure 14: KS (Kolmogorov-Smirnov) statistic

### 3.5. Xgboost model validation

Based on the comprehensive analysis conducted above, and considering the Xgboost model's superior performance, we selected the Xgboost model for validation against an out-of-time sample. As previously described, Xgboost demonstrated optimal computational efficiency, with tightly distributed residuals, a low RMSE, and a satisfactory scatter plot showing predicted prices closely aligned with actual values. The model also exhibited a superior KS statistic, indicating a near-identical cumulative distribution function (CDF) of predicted values compared to the CDF of actual prices.

The out-of-time dataset consisted of approximately 10% of the dataset, representing approximately 8,600 observations. This out-of-time sample was selected from the most recent observations, i.e., from 2020 to 2023, to ensure that the model was tested against the most recent data. The out-of-time sample was also selected to ensure that the model was tested against a sample that was not used in the model's training phase. As highlighted in the data exploration section, vehicle prices saw a significant increase during this period due to supply chain constraints and overall inflation, therefore, making this sample particularly relevant for testing the model's predictive power.

When the Xgboost model was applied to the out-of-time test data, the RMSE increased substantially. While the distribution of residuals maintained its normal distribution, it exhibited a slightly broader spread compared to the training phase. The scatter plot of predicted versus actual prices maintained a symmetric distribution around the expected diagonal, although the points deviated further from this line. Additionally, the KS statistic substantially increased to 0.57, as the CDF distribution shifted to the right.

These results suggest that there are external factors influencing car prices during this time period that were not considered in the model's training phase. Elements such as inventory levels, production dynamics, and customer purchasing power can impact vehicle prices beyond the model's scope. Including these factors in the model's training phase would likely improve the model's predictive power.

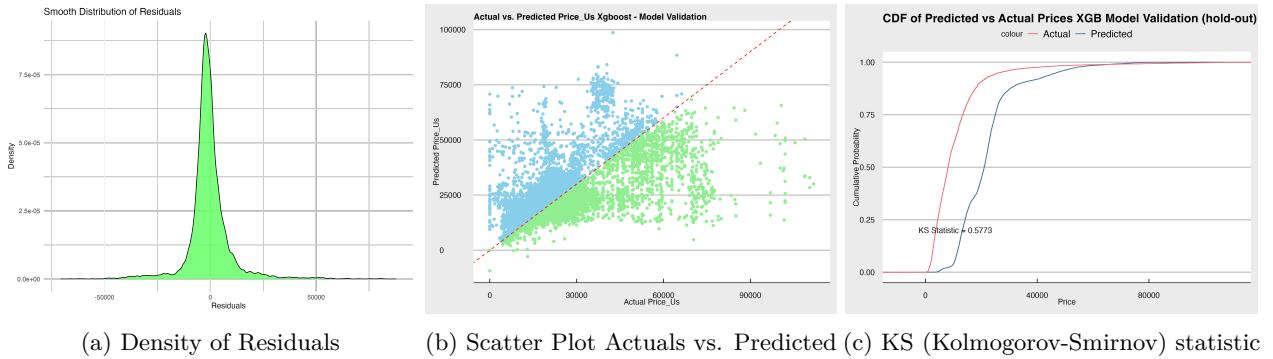


Figure 15: Model Validation: Out-Of-Time

## Section 4: Conclusion and Future Work

We explored seven distinct approaches for predicting car prices, with machine learning algorithms exhibiting remarkable superior performance, surpassing conventional models like linear regression. Nevertheless, harnessing the potential of machine learning algorithms requires more extensive data preprocessing efforts. We also learned that not all machine learning algorithms yielded favorable results. Specifically, algorithms not suited for continuous variable prediction, such as kNN, demonstrated poor performance, even falling behind a basic benchmark of simple average prices. Among the machine learning algorithms, Random Forest (RF) incurred the highest computational effort, while Light GBM and Xgboost showcased comparable performance with highly efficient computational demands.

These machine learning algorithms have applications extending beyond car price prediction, finding utility in forecasting the values of assets such as real estate and commercial equipment. Similarly, they can be adapted to predict stock prices and commodity values within financial markets.

Finally, it is vital to recognize the limitations of these algorithms: they are only as good as the data they are trained on. In our case, the dataset was limited to a specific country, Pakistan, and a specific time period, 1983 to 2023. Furthermore, the dataset did not include macroeconomic factors that can influence car prices, such as the exchange rate between the Pakistani Rupee and the US Dollar, and the presence of exogenous factors subject to potential shifts due to unforeseen macroeconomic conditions. Exploring these macroeconomic factors would be an exciting direction for future endeavors as a data scientist.

## Section 5: References

- [1] Muhammad Awais Tayyab. Used Car Price Prediction - Pakistan Dataset. URL: <https://www.kaggle.com/datasets/muhammadawaistayyab/used-car-price-prediction-pakistan/data>
- [2] Pakistani Rupee to US dollar 2023 exchange rate. URL: <https://www.google.com/search?client=safari&rls=en&q=exchange+rate+pak+rupee+to+us+dollar&ie=UTF-8&oe=UTF-8>
- [3] Analytics Vidhya. (2023). A Complete Guide to K-Nearest Neighbors (Updated 2023). URL: <https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/>
- [4] Understand Random Forest Algorithms With Examples (Updated 2023). URL: [<https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/>]    <https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/>
- [5] R Documentation: `predict.lgb.Booster`. URL: [<https://search.r-project.org/CRAN/refmans/lightgbm/html/predict.lgb.Booster.html>],    <https://search.r-project.org/CRAN/refmans/lightgbm/html/predict.lgb.Booster.html>
- [6] Tianqi Chen, Tong He, Michaël Benesty. XGBoost presentation. URL: <https://cran.r-project.org/web/packages/xgboost/vignettes/xgboostPresentation.html>.
- [7] Analytics Vidhya. (2021). Gradient Boosting Algorithm: A Complete Guide for Beginners. URL: <https://www.analyticsvidhya.com/blog/2021/09/gradient-boosting-algorithm-a-complete-guide-for-beginners/>
- [8] Arize. Kolmogorov Smirnov Test: When and Where To Use It. URL: <https://arize.com/blog-course/kolmogorov-smirnov-test/>
- [9] Rafael A. Irizarry. (2019). Advanced Data Science: Statistics and Prediction Algorithms Through Case Studies. Section 1 Distributions. URL: <http://rafalab.dfc.harvard.edu/dsbook-part-2/summaries/distributions.html>
- [10] Rafael A. Irizarry. (2023). Introduction to Data Science: Data Analysis and Prediction Algorithms with R. Chapter 13 Probability. URL: <http://rafalab.dfc.harvard.edu/dsbook/probability.html>