Used Car Price Prediction using XGB Regressor

Vicky Li
UC San Diego
San Diego, CA, USA
yil164@ucsd.edu

Evelyn Huang
UC San Diego
San Diego, CA, USA
xih037@ucsd.edu

Vivian X Zhao
UC San Diego
San Diego, CA, USA
vxzhao@ucsd.edu

	manufacturer	model	year	mileage	engine	transmission	drivetrain	fuel_type	mpg	exterior_color	interior_color	accidents_or_damage	one_owner	personal_use_only	seller_name	seller_rating	driver_rating	driver_reviews_num	price_drop	price
0	Acura	ILX Hybrid 1.5L	2013	92945.0	1.5L I-4 i-VTEC variable valve control, engine...	Automatic	Front-wheel Drive	Gasoline	39-38	Black	Parchment	0.0	0.0	0.0	Iconic Coach	NaN	4.4	12.0	300.0	13988.0
1	Acura	ILX Hybrid 1.5L	2013	47645.0	1.5L I4 8V MPFI SOHC Hybrid	Automatic CVT	Front-wheel Drive	Hybrid	39-38	Gray	Ebony	1.0	1.0	1.0	Kars Today	NaN	4.4	12.0	NaN	17995.0
2	Acura	ILX Hybrid 1.5L	2013	53422.0	1.5L I4 8V MPFI SOHC Hybrid	Automatic CVT	Front-wheel Drive	Hybrid	39-38	Bellanova White Pearl	Ebony	0.0	1.0	1.0	Weiss Toyota of South County	4.3	4.4	12.0	500.0	17000.0
3	Acura	ILX Hybrid 1.5L	2013	117598.0	1.5L I4 8V MPFI SOHC Hybrid	Automatic CVT	Front-wheel Drive	Hybrid	39-38	Polished Metal Metallic	NaN	0.0	1.0	1.0	Apple Tree Acura	NaN	4.4	12.0	675.0	14958.0
4	Acura	ILX Hybrid 1.5L	2013	114865.0	1.5L I4 8V MPFI SOHC Hybrid	Automatic CVT	Front-wheel Drive	Hybrid	39-38	NaN	Ebony	1.0	0.0	1.0	Herb Connolly Chevrolet	3.7	4.4	12.0	300.0	14498.0

Figure 1: First 5 Rows of the Dataset Before Data Cleaning.

Abstract

Forecasting the value of second-hand vehicles represents a critical and engaging field of study. Along with the pandemic, a surge in interest within the second-hand vehicle market has amplified the business opportunities for both purchasers and vendors. Achieving dependable and precise price forecasts matters for many people. In this research, we introduced a supervised machine learning approach that employs the XGB Regressor to evaluate the pricing of second-hand cars after comparing to many other models. Our model training and analysis was grounded on a dataset with data of used cars, collected from Kaggle, and our investigation involved scrutinizing the dataset across various training and testing splits. The RMSE of our best model using XGB Regressor is approximately 45.5% better than the baseline model with linear regression, positioning it as a well-optimized solution for predicting the prices of used cars.

Keywords

Histogram based Gradient Boosting, XGB Regression, Prediction, RMSE, Preprocessing, Regression, Cross-validation, K-Fold Validation

1 Introduction

Do you have a car or are considering to purchase one? If so, would you choose a brand new car or used-car? Our research explores a question that most people will think for at least once in their life times: what factors most significantly affect the prices of used cars? This research question is vital as it directly impacts the decision-making process for countless individuals and businesses involved in the second-hand car market. We believe that the **mileage of a used car, the year it was produced, and any price drop of the car from its initial price** are the most influential factors of the price in which a used car is sold.

To anchor our analysis, we have selected a rich dataset from Kaggle that encompasses a range of variables associated with used cars.

Table 1: Description of Features

Feature	Data Types	Description
manufacturer	object	car manufacturer name
model	object	car model name
year	int64	car production year
mileage	float64	number of miles
engine	object	car engine
transmission	object	car transmission type
drivetrain	object	car's drivetrain type
fuel_type	object	type of fuel
mpg	object	mile per gallon
exterior_color	object	car exterior color
interior_color	object	car interior color
accidents_or_damage	float64	binary
one_owner	float64	binary
personal_use_only	float64	binary
seller_name	object	name of the seller
seller_rating	float64	seller's rating
driver_rating	float64	car rating by drivers
driver_reviews_num	float64	#car reviews by drivers
price_drop	float64	drop from initial price
price	float64	price when sold

Table 1 shows the data types of each feature available in the dataset before data cleaning and transformation. See Figure 1 to see the first 5 rows of the dataset. More details can be found in our Github Repository: *github.com/vickyli1015/DSC148_Final_Project*.

2 Data Cleaning

In preparing our dataset for analysis, we executed a comprehensive data cleaning process to ensure the integrity and quality of the data. This step was crucial to minimize errors and biases in our predictions. The initial dataset consisted of 762,091 records and

20 features, including **manufacturer, model, year, mileage, engine, transmission, drivetrain, fuel_type, mpg, exterior_color, interior_color, accidents_or_damage, among others**.

2.1 Outliers Removal

We began by addressing outliers in the **price** column, where values were found to range unreasonably from 1 dollar to 100 million dollars. These outliers were removed to prevent skewing our analysis, retaining prices within realistic bounds. The **mileage** column's outliers were addressed by removing any value at the 98th percentile and above, rationalizing that excessively high mileage is atypical for standard vehicle transactions. The highest mileage we had in the data set was 1119067 miles (46627 times the Earth's circumference). The 98th percentile was chosen because it excludes values that are more than three standard deviations away from the mean assuming the feature holds a normal distribution.

2.2 Dimension Reduction

Irrelevant features, such as **seller_name**, were dropped from the dataset. This decision was based on the preliminary analysis indicating no significant impact of these variables on the vehicle's price.

2.3 Categorical Variables Encoding

Several columns underwent feature engineering:

The **exterior_color** and **interior_color** fields were standardized to a set of predefined colors to reduce variability and complexity. For example, variations of "gray" (including grey, gry, steel, silver, metallic, platinum) were all categorized under a single "gray" label. Only colors that are most common (black, gray, white, red, blue, green, yellow, etc) were extracted and the rest are set to "Other". This shrunk the number of unique colors from 6991 to 11. A smaller number of categories reduces noise and eases the one-hot transformation later in the analysis.

We encode categorical variables like **drivetrain** and **fuel_type** into numerical formats suitable for machine learning algorithms. The original dataset had all kinds of variations for these features and many had the same meaning. We redefined the varieties into a few common types. For instance, fuel types were categorized into 'Gasoline', 'Hybrid', 'Electric', 'Diesel', 'Premium', 'Biodiesel', and 'Other', capturing the essential distinctions relevant to vehicle pricing.

The **transmission** field was transformed into "Automatic", "Manual", or "Other".

2.4 Key Feature Extraction

The original data had all engine data such as horsepower, tank size, transmission set to one string feature: **engine**. New features such as **tank_size** were extracted from existing data, providing additional usable features for our modeling.

2.5 Cleaned Dataset

After cleaning, our dataset was reduced to 741,009 records, each with a refined set of features optimized for predictive modeling. This dataset formed the basis for training our supervised machine learning model, ensuring a robust foundation for accurate and reliable price forecasts. Reference Table 2 for summarized changes.

Table 2: Description of New/Changed Features

Feature	Data Types	Changes Made
price	float64	remove outliers
mileage	float64	remove outliers
seller_name	object	removed
exterior_color_cleaned	object	extract common colors
interior_color_cleaned	object	extract common colors
drivetrain	object	extract drivetrain type
fuel_type	object	extract fuel type
transmission	object	extract transmission type
tank_size	float64	extracted from engine

3 Exploratory Data Analysis (EDA)

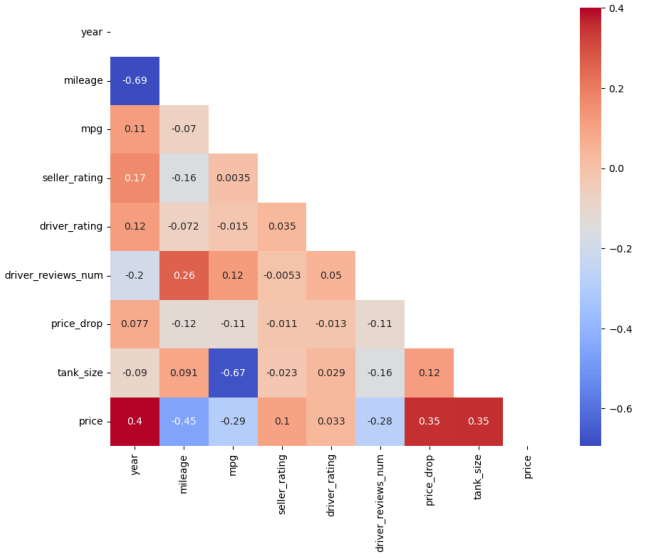

Figure 2: Correlation matrix of numerical features

The correlation matrix in Figure 3 displays correlation between all numerical features in the dataset and the listed prices, indicated by **price**. Features such as **year**, **price_drop**, and **tank_size** exhibit strong positive correlations with the listed price. On the other hand, **mileage** is the primary feature negatively impacting the listed price. Additionally, there is a noticeable negative correlation between the production year and mileage, indicating that newer cars tend to have lower mileage. Similarly, there is a relatively strong negative correlation between **tank_size and mpg (miles per gallon)**, suggesting that larger tank sizes are associated with lower fuel efficiency. The correlation between features are not concerning as they did not exceed magnitude of 0.8, which is the rule of thumb that indicates whether there will be a problem of multicollinearity [3].

In Figure3, we can observe the price distribution for all the available binary features. Overall, across all binary columns, they are

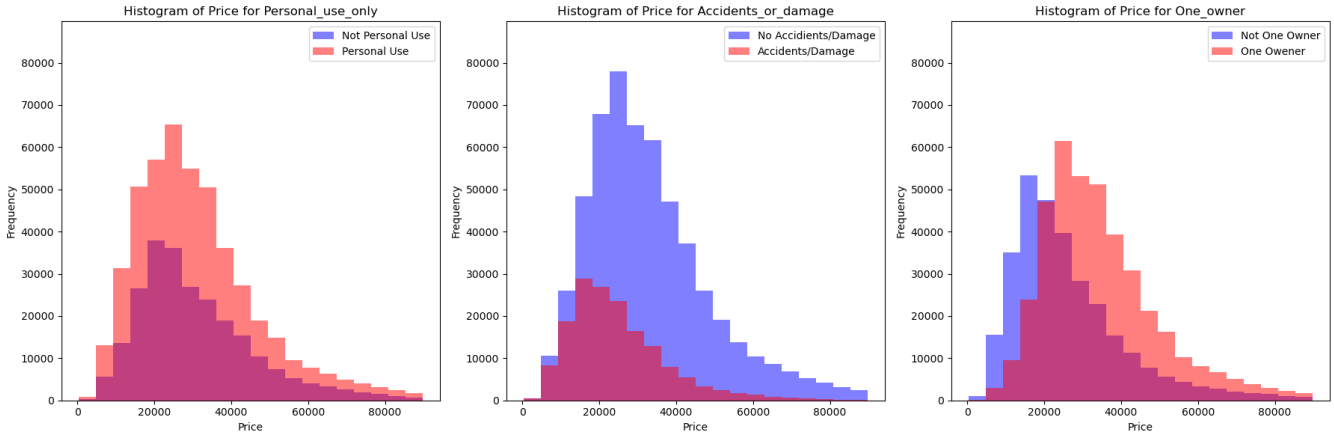

Figure 3: Distribution of price for binary columns (For better visualization displays, we only show listed price that is within 95th percentile)

all slightly left-skewed, meaning that it's more frequent for a car to have relatively lower prices than higher prices regardless of features. However, we can still dive into each category to extract insights. The first feature **Personal_use_only** which denotes whether the car is for personal use has similar distributions in both categories. The frequency for cars that are only personal use is higher for each price level because there are more cars of this type, but overall the percentage in each price level within cars of the 2 kinds are similar. For **Accidents_or_damage**, the proportion of cars with no accident history is much higher than those with accident records. Additionally, cars with accident histories tend to have lower prices, which align with our common intuition. For **One_owner**, which indicates the number of previous owners, we can see an relatively equal proportion across both categories. However, cars with only one previous owner generally with higher listed prices.

In Figure4, we can see that for categorical data that relates to the interior and exterior colors of the car, an intriguing pattern we observed is cars with uncommon colors (bars marked in green) tend to have higher mean price. This can be explained by that for higher-value automobiles, particularly those within specialized niches such as race cars, are more inclined to be customized by their owners for individual preference. For transmission features, cars equipped with manual transmissions have mean price lower than other type. For fuel type, cars that use Diesel fuel are tended to have more higher listed price and wider price range.

For model inspiration, we can put emphasis on numerical features that have high correlation with price and put less emphasis on less-correlated features. To achieve this, we can use Lasso regression by setting coefficients of less important features to zero, handles multicollinearity. We can also use XGB Regressor which prioritized the most informative features in the ensemble of decision trees and is suitable for handling skewed data. For categorical data and ratings, since we have many categories, model like Histogram-based Gradient Boosting Regression Tree would be helpful since it can discretize the categories into bins during the histogram construction process.

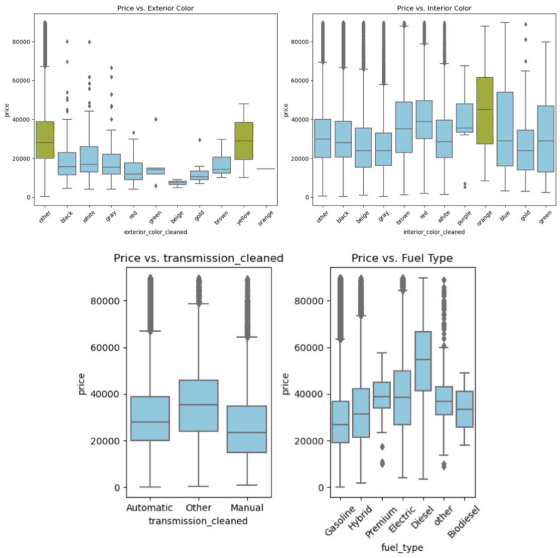

Figure 4: Distribution of price for categorical columns (For better visualization displays, we only show listed price that is within 95th percentile)

4 Pre-processing

In Table 3, we have all features that have missing values and the proportion of their missing value. We decided to do imputation without dropping any of them since none of them exceed 50% missing values and we decided to preserve diversity in dataset instead. For numerical columns with missing values, we decided to impute missing values based on distributions. We first binned the continuous values using 10 bins, then we randomized value within the randomized bins to fill the the null values based on the frequency of the data to fall into each bin. By doing this, we largely preserved the original distribution, accounting for the skewness in the features while ensuring randomness in imputation and fairness. For binary

Table 3: Columns with Missing Values

Column Name	Null Value Proportion	Data Type
accidents_or_damage	0.030855	binary
one_owner	0.037749	binary
personal_use_only	0.030981	binary
seller_rating	0.275075	numerical
mpg	0.190716	numerical
driver_rating	0.040116	numerical
price_drop	0.457415	numerical
tank_size	0.069033	numerical

categorical features, we did similar imputation by randomly assigned based on the likelihood of each categories. After imputation, we performed one-hot encoding to turn categorical data into binary columns for model fitting.

5 Predictive Task

5.1 Task Identification

This task aims to forecast the price of second-hand vehicles, a continuous numerical output, making it a **regression problem**.

5.2 Evaluation Metric

The evaluation of different predictive models revolves around regression metrics. **Root Mean Squared Error (RMSE)** is chosen to be the primary evaluation metric, due to its effectiveness in quantifying the difference between the predicted and actual prices of used cars. Lower RMSE values indicate better model performance.

$$\text{RMSE}(y, \hat{y}) = \sqrt{\frac{\sum_{i=0}^{N-1}(y_i - \hat{y}_i)^2}{N}}$$

RMSE provides a clear measure of how much the predicted values deviate, on average, from the actual values in the dataset. Unlike R-squared or Adjusted R-squared, which are more abstract in indicating the proportion of variance explained by the model, RMSE offers a more intuitive and direct interpretation in the same units as the target variable. This direct comparison between prediction errors made RMSE particularly suitable for assessing and comparing the performance of our predictive models in a practical, real-world context.

5.3 Baseline Model

Linear Regression was chosen as our baseline model. It's one of the simplest forms of regression, providing a straightforward interpretation of how input features affect the target variable. As a basic model, it serves as an effective benchmark for comparing the performance of more complex models.

5.4 Comparative Analysis with Other Models

Apart from the baseline Linear Regression, we explored several models, including **Ridge Regression, Lasso Regression, and Random Forest Regression**. These models were chosen to test various aspects of our dataset, from feature collinearity (addressed by Ridge) to feature selection (handled by Lasso) and non-linear relationships (captured by Random Forest Regression). Each model provided insights into the dataset's characteristics but also revealed limitations in handling its complexity and specific data patterns as what is shown in Table 4, they all have relatively higher RMSE comparing to XGBoost and HBGB Regression Tree. The RMSE scores in Table 4 were obtained from models that were tuned directly using the results from a 5-fold grid search.

Table 4: Compare RMSE of Models tuned by 5-fold Grid-Search

Model	Train	Validation	Test
Mean price	32913.29	32832.65	32844.56
Linear Regression	16754.45	15807.01	16442.54
Ridge Regression	16754.57	15807.07	16442.73
Lasso Regression	16754.51	15806.86	16442.53
Random Forest Regression	11261.65	10695.08	11263.92
HBGB Regression Tree	9327.48	9270.27	9686.97
XGB Regressor	5212.70	8001.22	8162.21

6 Model and Result

Our exploration culminated in focusing on two advanced models: XGB Regression and Histogram-based Gradient Boosting Regression Tree, as they have generally lower RMSE comparing to others. These models were selected for the following reasons:

- **Efficiency and Scalability**: Both models are designed for speed and efficiency, capable of managing large datasets with sophisticated algorithms.
- **Robustness to Outliers and Low risk of Overfitting**: Both models are using ensemble methods that build multiple weak learners and combine their predictions. Moreover, both model can be regulated by many hyper-parameters with hyper-parameter turning. For HGBB Rregression, the binning also reduce the effect of outlier on prediction values. XGBoost have scale_pos_weight specifically for handling imbalanced datasets.
- **Feature Importance and Interpretability**: Both models provide insights into feature importance which account for the correlation between features and predicted value and gave insight to our hypothesis.

We first performed Grid Search to find the best parameters for both models, then adjust the parameters to to mitigate overfitting and lastly we performed a 5-fold cross validation for evaluation. In Table 5, we have included the optimal hyperparameters resulted by Grid Search for both model. We used these hyperparameters to fit the models and make predictions. The results are as below:

- HBGB Regression Tree: {training RMSE = 9327.48, validation RMSE = 9270.27, test RMSE = 9686.97}
- XGB Regression: {training RMSE = 5212.70, validation RMSE = 8001.22, test RMSE = 8162.21}

Based on the results, we observed that both models outperformed the baseline models. However, the XGB Regressor exhibited signs of overfitting, as seen by the significant decline in training RMSE compared to validation and test RMSE. This indicates that the model may have captured specific patterns in the training dataset that

Table 5: Grid-Search Tuning for Both Models

Model	Hyperparameters
HBGB Regression Tree	"learning_rate": 0.2, "max_depth": 7, "max_iter": 300, "max_bins": 150, "random_state": 42
XGB Regressor	'colsample_bytree': 1.0, 'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 300

do not generalize well to unseen data. Therefore, we adjusted the hyperparameters to account for the overfitting by

- Decreasing learning rate to 0.05
- Specify eta = 0.1
- Specify subsample = 0.7

After tuning, the result we were able to achieve a training RMSE of 7391.16, validation RMSE of 8780.28 and test RMSE of 8949.65, while still being able to significantly outperform the baseline models which has a training RMSE of at least 16754.45, validation RMSE of 15807.01 and test RMSE of 16442.54. The RMSE scores for the finalized models are summarized in Table 6.

Reference Table 7 to see how the tunned models are evaluated with 5 folds Cross-Validation and their RMSE. In Table 7, we can see the Cross-Validation scores measured in RMSE for the tunned models. The large RMSE in the first fold may be due to bad splitting, resulting in a high RMSE as it is large for both model. As we can observe, two models have similar performance with XGBoost exhibiting a slightly lower RMSE in each fold and in average(the mean RMSE).

6.1 Final Model Choice: XGB Regressor

After experimenting with various models, we decided to focus on XGB Regressor as our final model for its stability when testing on different train-test-split and random seeds.

XGB Regressor model combines all the advantages of our previous attempts of models. XGB Regressor implements both L1 (Lasso regression) and L2 (Ridge regression) regularization. It also prunes trees using depth-first approach and allows for both pre-pruning (limiting the depth of trees) and post-pruning (removing splits based on the gain), which can lead to the creation of more optimal trees. In conclusion, while the Histogram-based Gradient Boosting Regression model showed promising results, XGB Regressor's overall performance to various data challenges made it the standout choice as our final model. This decision was backed by thorough comparative analysis and validation against a range of models, ensuring confidence in its predictive capabilities for forecasting second-hand vehicle prices.

7 Literature

The task of predicting used-car prices has been studied by others before. The papers "Used Car Price Prediction using K-Nearest Neighbor Based Model" by Dr R.Ashok Kumar K.Samruddhi1 et al. [1] and "Used Car Price Prediction using Supervised Learning Techniques" by Mukkesh et al. [2] are examples of such studies. Dr R.Ashok Kumar K.Samruddhi1 et al utilized a single classification model approach, while Mukkesh et al. conducted a comparative study of multiple regression techniques. Our work presented how the dataset was meticulously cleaned and pre-processed with innovative techniques like outlier removal, dimension reduction, and feature extractions. Also, we used a complex model which addressed most problems we met (such as overfitting) in the process of model selection. All works provided a well functioning prediction model for the task. Our result was similar to Mukkesh et al.'s results where it concluded that **ensemble methods and decision trees will create more accurate predictions**.

Table 6: Compare RMSE of Models with Finalized Hyperparameters

Model	Train	Validation	Test
Mean price	32913.29	32832.65	32844.56
Linear Regression	16754.45	15807.01	16442.54
Ridge Regression	16754.57	15807.07	16442.73
Lasso Regression	16754.51	15806.86	16442.53
Random Forest Regression	11261.65	10695.08	11263.92
HBGB Regression Tree	9327.48	9270.27	9686.97
XGB Regressor	7391.16	8780.28	8949.65

Table 7: Cross-Validation

Model	RMSE (in Each Fold)	Mean RMSE
HBGB Regression Tree	[14956.47 10728.65, 10125.50, 8871.40, 9122.84]	10760.97
XGB Regression	[14981.31 9663.17 10210.65, 8164.67, 8422.68]	10228.50

8 Conclusion

In Table 8, we utilized XGB Regressor build-in function to get the "importance score" for each features. In addition, we calculated Permutation Feature Importance (PFI) which measures the change in a model's performance metric when the values of a feature are randomly shuffled. Note that the XGBoost importance score are scaled to sum up to 1 so the values does not account for the negative/positive correlation. PFI indicates the decrease in a model's

Table 8: Feature Importance

Features	XGB Importance Score	PFI
driver_reviews_num	0.054	0.21
mileage	0.037	0.25
tank_size	0.036	0.22
year	0.031	0.302
manufacturer_Porsche	0.177	0.221
manufacturer_Mercedes-Benz	0.064	0.0767

performance metric caused by permuting the values of a specific feature.

What is surprising is that among the original feature, **manufacturer_Porsche** is the most important feature found by the XGBoost built-in function, followed by **driver_review_number, mileage, tank_size, year**. For PFI, **year** is the most important features followed by **mileage, tank_size and manufacturer_Porsche** which are different from what we have hypothesized in the beginning that mileage, year and price_drop are the most important ones. We can see the overlapping of important feature using different approaches, generally, manufacturer_Porsche, mileage,driver_review_number and tank_size are the most importance features. The importance score of manufacturer_Porsche measures the importance of the dummy variable for Porsche and can be explained by the fact that Porsche cars usually are having higher listed price and the dummy variable account for bringing up the price for this specific brand. Among the features, the surprising finding would be the importance of driver_review_number.

The key takeaway is that **features that related to the functionality of the cars cast a great impact on the listed price**. On the other hand, the **color of a car** has less effect on listed price. Comparing to seller ratings, driver's comments on the cars is more valuable. When it comes to modeling, we can see the power of XGBoost in completing different tasks and the options for fine-tuning, making it a great fit for a wide range of data types.

References

[1] Dr R.Ashok Kumar K.Samruddhi1. 2020. Used Car Price Prediction using K-Nearest Neighbor Based Model. Retrieved March 14, 2024 from https://ijirase.com/assets/paper/issue_1/volume_4/V4-Issue-2-629-632.pdf

[2] Mukkesh Ganesh Pattabiraman Venkatasubbu. 2019. Used Cars Price Prediction using Supervised Learning Techniques. Retrieved March 14, 2024 from https://www.researchgate.net/profile/Mukkesh-Ganesh/publication/343878698_Used_Cars_Price_Prediction_using_Supervised_Learning_Techniques/links/5f461ab492851cd30230688b/Used-Cars-Price-Prediction-using-Supervised-Learning-Techniques.pdf

[3] Robert Signorile. 2021. CHAPTER 8: MULTICOLLINEARITY. Retrieved March 14, 2024 from "https://www.sfu.ca/~dsignori/buec333/lecture%2016.pdf"