Car Sales Prediction (May 2022)

Lamiss Gargouri
Master student in Bahcesehir University
Department of Big Data Analytics and Management.

ABSTRACT The manufacturer sets the price of the new cars in the market, with the state incurring some extra costs as part of taxes. Consumers buying a new car can rest comfortable that the money they spend will be well spent. However, due to rising new car prices and customers' inability to purchase new automobiles due to the absence of cash, used car sales are on the rise worldwide. A method to accurately identify the worthiness of a secondhand automobile utilizing several criteria is required. Although there are websites that provide this service, their technique of prediction may not be the best. Furthermore, several models and algorithms may aid in the prediction of real market value for a used automobile. When purchasing or selling, it's critical to understand their current market value.

INDEX TERMS Exploratory Data analysis, Data visualization, Liner regression, Model Comparison, Regression.

I. INTRODUCTION

Due to the numerous elements that influence a used vehicle's market pricing, determining if the advertised price is accurate is a difficult undertaking. The goal of this research is to create machine learning algorithm that can properly forecast the cost of a used automobile based on its attributes so that buyers can make educated decisions. On a dataset including the selling prices of various brands and models, we build and analyze several learning approaches.
Our findings reveal that linear regression produced adequate results, with the added benefit of a substantially shorter training period than the other approaches.

II. BACKGROUND

A. BASIC CONSEPTS

- **linear regression** produced adequate results, with the added benefit of a substantially shorter training period than the other approaches.

- **Exploratory Data Analysis** is the crucial process of using statistical results and visualizations to do early investigations on data in order to uncover patterns, detect anomalies, test hypotheses, and verify hypotheses.

- **Data visualization** is known as the graphical display of data and information. Data visualization tools make it easy to observe and comprehend patterns and trends in data by employing visual components like graphs and charts.

- **Linear regression** is a linear model in which the input variables (x) and the single output variable (y) have a linear relationship (y).

- **Regression analysis** is a strong statistical tool for examining the connection between two or more variables of interest.

III. Dataset

A. Context

You must model the pricing of automobiles using this dataset and the available independent factors. It will be utilized by management to determine how prices fluctuate in relation to the independent factors. They can then adjust the car's design, commercial strategy, and other factors to fulfill specified pricing targets. Furthermore, the model would assist management in comprehending the price characteristics of a new market.

B. Content

This dataset provides statistics about used automobiles. This information may be utilized for a variety of applications, including price prediction, which demonstrates the usage of regression analysis in Machine Learning.

⇨ The following are the columns in the supplied dataset:

- Car_Name: The name of the vehicle should be entered in this field.
- Year: The year the automobile was purchased should be entered in this field.
- Selling_Price : The price the owner intends to sell the automobile for should be entered in this field.
- Present_Price: This is the car's current ex-showroom pricing.
- Kms_Driven: The distance traveled by the automobile in kilometers.
- Fuel_Type: The car's fuel type.
- Seller_Type: Indicates if the vendor is a business or a private individual.
- Transmission: This setting determines whether the vehicle is manual or automated.
- Owner: The number of past owners of the automobile is specified.

C. Acknowledgements

The data was collected from a publicly accessible website: www.cardekho.com

IV. Importing Libraries & Data Exploration

To begin the project, import libraries like NumPy for data visualization and Matplotlib for data visualization, and then load the data using Pandas. Read csv and save it. Data is entered by typing. The Cars.Head() function displays the first five records of the data set, providing a quick overview of the data frame's rows and columns.

```
#Read dataset
#Get information about Dataset
cars =pd.read_csv('car_data.csv')
cars.head()
```

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
0	ritz	2014	3.35	5.59	27000	Petrol	Dealer	Manual	0
1	sx4	2013	4.75	9.54	43000	Diesel	Dealer	Manual	0
2	ciaz	2017	7.25	9.85	6900	Petrol	Dealer	Manual	0
3	wagon r	2011	2.85	4.15	5200	Petrol	Dealer	Manual	0
4	swift	2014	4.60	6.87	42450	Diesel	Dealer	Manual	0

V. Understanding the Structure of the data

```
cars.describe()
```

	Year	Selling_Price	Present_Price	Kms_Driven	Owner
count	301.000000	301.000000	301.000000	301.000000	301.000000
mean	2013.627907	4.661296	7.628472	36947.205980	0.043189
std	2.891554	5.082812	8.644115	38886.883882	0.247915
min	2003.000000	0.100000	0.320000	500.000000	0.000000
25%	2012.000000	0.900000	1.200000	15000.000000	0.000000
50%	2014.000000	3.600000	6.400000	32000.000000	0.000000
75%	2016.000000	6.000000	9.900000	48767.000000	0.000000
max	2018.000000	35.000000	92.600000	500000.000000	3.000000

Pandas describe() is used to display some basic statistical information of a data frame or a sequence of numeric values, such as percentile, mean, and standard deviation. This method provides a distinct result when used to a sequence of strings, as seen in the examples below. Return Type: Data frame statistical summary.

```
cars.shape
```

```
(301, 9)
```

Return the shape of an array

```
cars.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 301 entries, 0 to 300
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Car_Name       301 non-null    object 
 1   Year           301 non-null    int64  
 2   Selling_Price  301 non-null    float64
 3   Present_Price  301 non-null    float64
 4   Kms_Driven     301 non-null    int64  
 5   Fuel_Type      301 non-null    object 
 6   Seller_Type    301 non-null    object 
 7   Transmission   301 non-null    object 
 8   Owner          301 non-null    int64  
dtypes: float64(2), int64(3), object(4)
memory usage: 21.3+ KB
```

Get a concise summary of the dataframe.

```
cars.isna().any()
```

```
Car_Name         False
Year             False
Selling_Price    False
Present_Price    False
Kms_Driven       False
Fuel_Type        False
Seller_Type      False
Transmission     False
Owner            False
dtype: bool
```

For an array-like object, find missing values. It's used to see if any element is True, maybe throughout an axis.

- ⇨ Our data appears to be complete. There are no NaN values, and the feature types are all correct.

```
[8]  #Print unique values in column (pandas code)
     print(cars['Car_Name'].unique())
     print(cars['Year'].unique())
     print(cars['Selling_Price'].unique())
     print(cars['Present_Price'].unique())
     print(cars['Kms_Driven'].unique())
     print(cars['Fuel_Type'].unique())
     print(cars['Seller_Type'].unique())
     print(cars['Transmission'].unique())
     print(cars['Owner'].unique())
```

```
['ritz' 'sx4' 'ciaz' 'wagon r' 'swift' 'vitara brezza' 's cross'
 'alto 800' 'ertiga' 'dzire' 'alto k10' 'ignis' '800' 'baleno' 'omni'
 'fortuner' 'innova' 'corolla altis' 'etios cross' 'etios g' 'etios liva'
 'corolla' 'etios gd' 'camry' 'land cruiser' 'Royal Enfield Thunder 500'
 'UM Renegade Mojave' 'KTM RC200' 'Bajaj Dominar 400'
 'Royal Enfield Classic 350' 'KTM RC390' 'Hyosung GT250R'
 'Royal Enfield Thunder 350' 'KTM 390 Duke ' 'Mahindra Mojo XT300'
 'Bajaj Pulsar RS200' 'Royal Enfield Bullet 350']
```

Print Unique Values in column (pandas code).

```
cars.isnull().sum()
```

```
Car_Name         0
Year             0
Selling_Price    0
Present_Price    0
Kms_Driven       0
Fuel_Type        0
Seller_Type      0
Transmission     0
Owner            0
dtype: int64
```

The number of missing values in the data collection is returned by this function.

```
[10] cars.columns
```

```
Index(['Car_Name', 'Year', 'Selling_Price', 'Present_Price', 'Kms_Driven',
       'Fuel_Type', 'Seller_Type', 'Transmission', 'Owner'],
      dtype='object')
```

Returns the Columns of our dataset.

```
print(cars.Fuel_Type.value_counts(),"\n")
print(cars.Seller_Type.value_counts(),"\n")
print(cars.Transmission.value_counts())
```

```
Petrol    239
Diesel     60
CNG         2
Name: Fuel_Type, dtype: int64

Dealer        195
Individual    106
Name: Seller_Type, dtype: int64

Manual       261
Automatic     40
Name: Transmission, dtype: int64
```

It returns the value counts of the qualities that make up the object type.

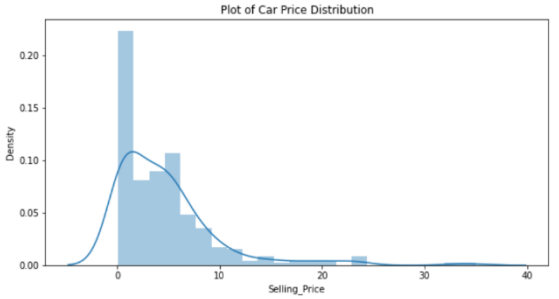
```
cars.Fuel_Type.replace(regex={"Petrol":"0","Diesel":"1","CNG":"2"},inplace=True)
cars.Seller_Type.replace(regex={"Dealer":"0","Individual":"1"},inplace=True)
cars.Transmission.replace(regex={"Manual":"0","Automatic":"1"},inplace=True)
cars[["Fuel_Type","Seller_Type","Transmission"]]=cars[["Fuel_Type","Seller_Type","Transmission"]].astype(int)
```

To make it suitable for regression models, I transformed these object values to numerical values.

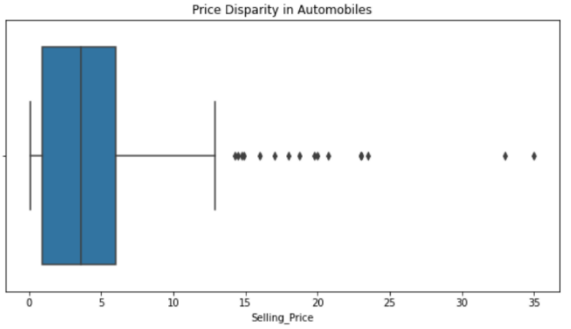
Fuel_Type : 1= petrol
0 = Diesel
2 = CNG
Transmission: 1= Manual
0= Automatic
Seller_Type: 1= Dealer
0 = Individual.

VI. Visualizing the Data

⇨ We will use visualizations to look at different combinations of characteristics while studying the data. This will assist us in better understanding our data and provide some insight into data patterns.



Plot of car price Distribution.



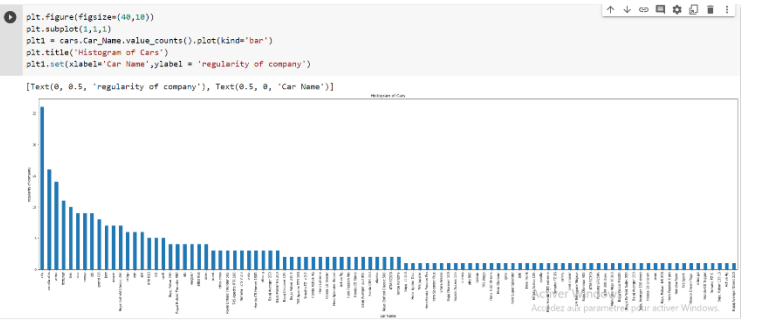
Plot of Price disparsity in cars.

```
print(cars.Selling_Price.describe(percentiles = [0.25,0.50,0.75,0.85,0.90,1]))
```

```
count    301.000000
mean       4.661296
std        5.082812
min        0.100000
25%        0.900000
50%        3.600000
75%        6.000000
85%        8.250000
90%        9.500000
100%      35.000000
max       35.000000
Name: Selling_Price, dtype: float64
```

value 50 should represent the "middle" of the data, commonly known as the median.

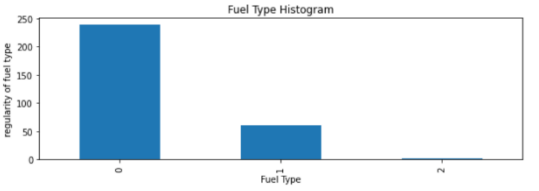
⇨ The data points are not far from the mean, indicating that there isn't a lot of variation in automobile pricing. (90% of the prices are below 9.5, whereas the remaining are between 9.5 and 35).



⇨ City seemed to be the top car type prefered.

```
plt.figure(figsize=(10,3))
plt.subplot(1,1,1)
plt1 = cars.Fuel_Type.value_counts().plot(kind='bar')
plt.title('Fuel Type Histogram')
plt1.set(xlabel='Fuel Type', ylabel = 'regularity of fuel type')
```

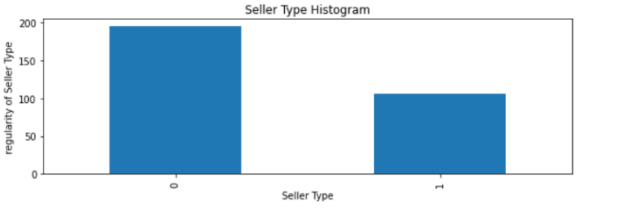
```
[Text(0, 0.5, 'regularity of fuel type'), Text(0.5, 0, 'Fuel Type')]
```



⇨ Number of Diesel fueled cars are more than Petrol and CNG.

```
plt.figure(figsize=(10,3))
plt.subplot(1,1,1)
plt1 = cars.Seller_Type.value_counts().plot(kind='bar')
plt.title('Seller Type Histogram')
plt1.set(xlabel='Seller Type', ylabel = 'regularity of Seller Type')
```

```
[Text(0, 0.5, 'regularity of Seller Type'), Text(0.5, 0, 'Seller Type')]
```



⇨ Number of Dealers cars are more than individual cars.

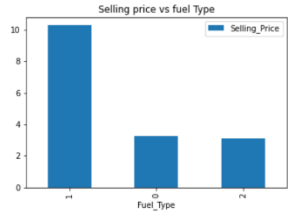
```
[43] df = pd.DataFrame(cars.groupby(['Year'])['Selling_Price'].mean().sort_values(ascending = False))
     df.plot.bar()
     plt.title('Selling price vs Year')
```

```
Text(0.5, 1.0, 'Selling price vs Year')
```

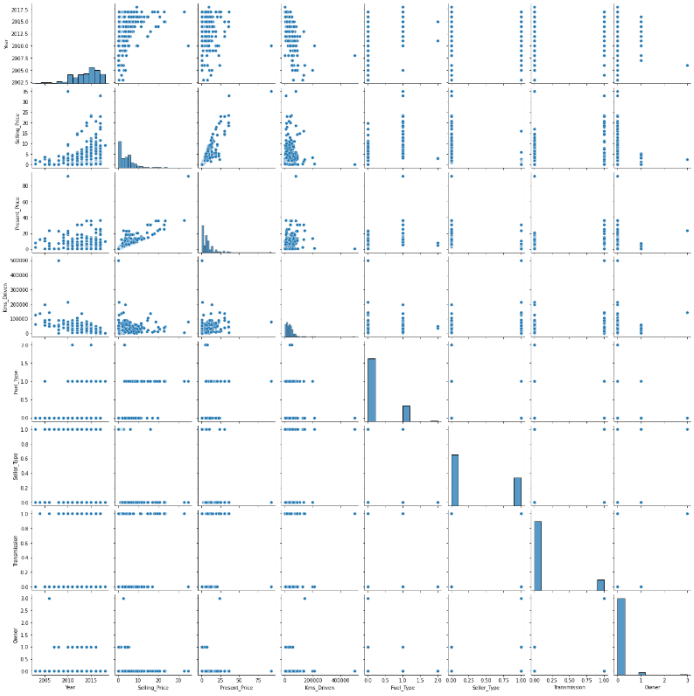
⇨ The newer the car, the more expensive it is.

```
df = pd.DataFrame(cars.groupby(['Fuel_Type'])['Selling_Price'].mean().sort_values(ascending = False))
df.plot.bar()
plt.title('Selling price vs fuel Type')
```

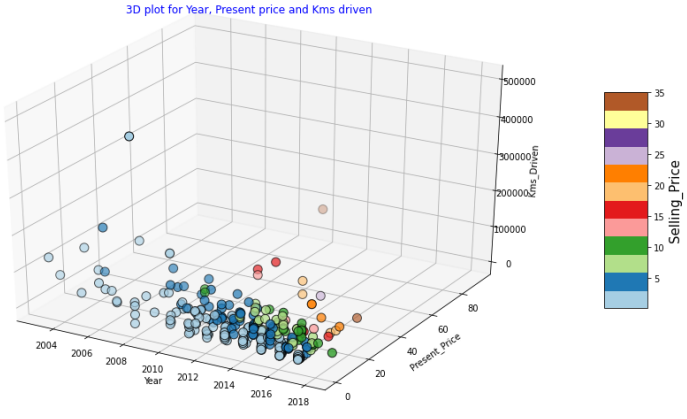
```
Text(0.5, 1.0, 'Selling price vs fuel Type')
```



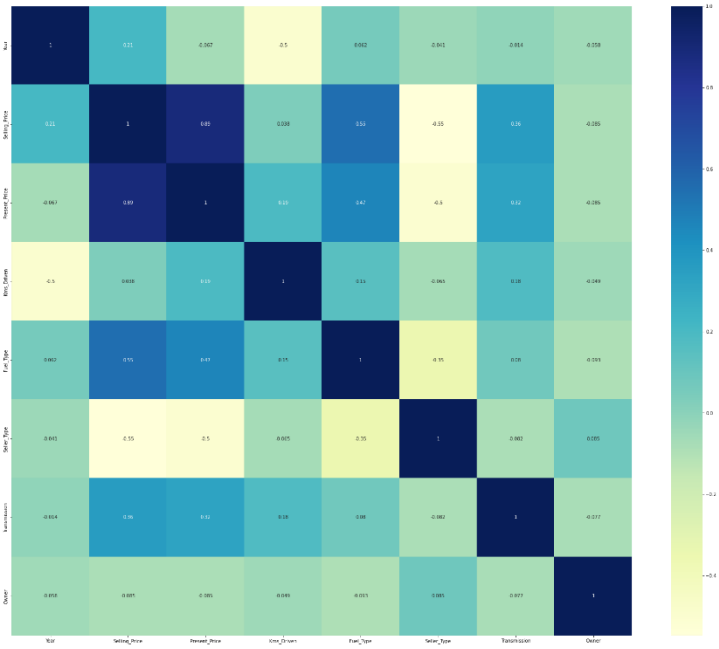
- ⇨ Diesel cars are more expensive than CNG and petrol cars.
- ⇨ While between petrol and CNG cars there is not a big difference.



- ⇨ Before using regression model I took a visual look at the characteristics and their relationships.



- ⇨ Following the creation of the 3D plot critique elements that influence selling price, we can observe that the majority of automobiles gather around the year 2010, with low current prices and low kilometers traveled.



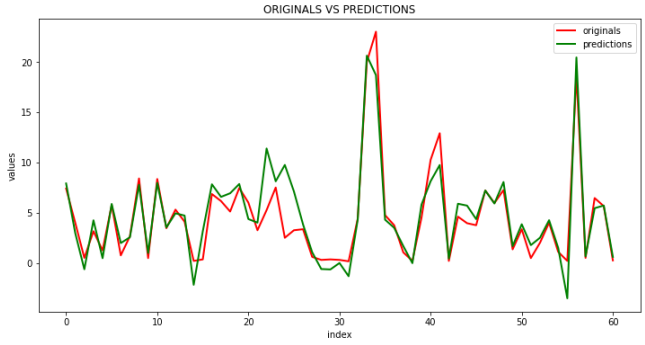
- ⇨ A correlation heatmap is a heatmap that depicts a two-dimensional correlation matrix between two discrete dimensions, with colored pixels representing data on a monochromatic scale.
- ⇨ The heatmap was created to illustrate relationship of two variables, one on each axis. I can see whether there are any trends in frequency with one or even both variables by looking at how cell colors vary across each axis.

VII. Regression Model

- ⇨ It's now time to put regression models to work.

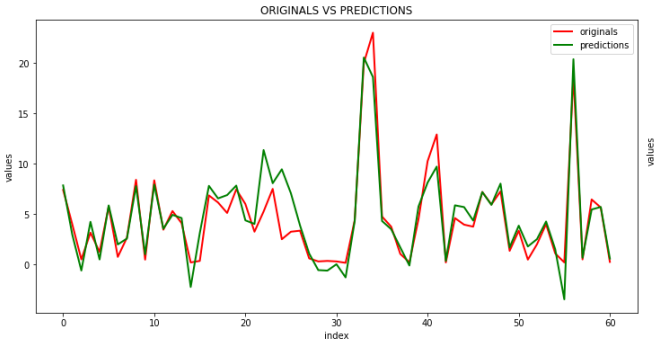
A. Linear Regression

- ⇨ A linear model is one in which the input variables (x) and the single output variable (y) are assumed to have a linear relationship (y).



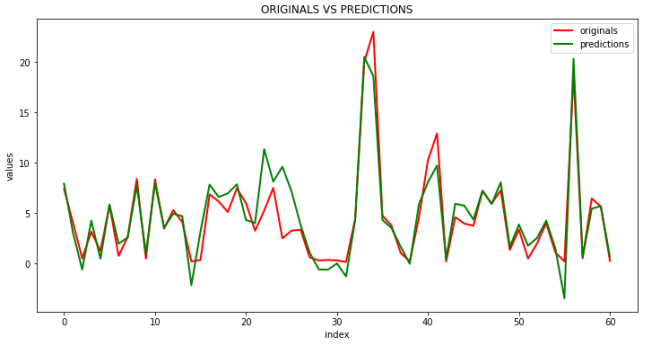
B. Lasso Regression Model

⇨ Lasso regression is a linear regression technique that employs shrinkage.



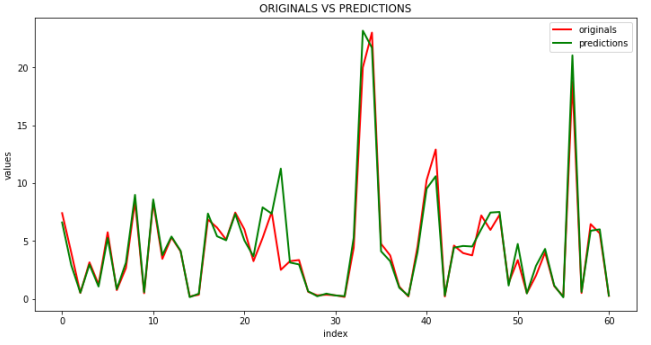
C. Ridge Regression Model

⇨ Ridge regression is a model tuning technique that may be used to analyze data with multicollinearity.



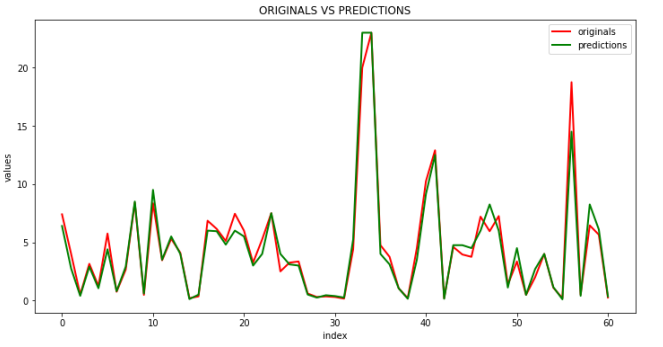
D. Random Forest Regression

⇨ Random Forest Regression is a supervised learning approach for regression that use the ensemble learning method.



E. Decision Tree Regression

⇨ Decision tree builds regression or classification models in the form of a tree structure.



VIII. Data frame

```
Model = ["LinearRegression","Lasso","Ridge","DecisionTreeRegressor","RandomForestRegressor"]
results=pd.DataFrame({'Model': Model,'R Squared': r_2,'CV score mean': CV})
results
```

	Model	R Squared	CV score mean
0	LinearRegression	0.848455	0.837659
1	Lasso	0.849557	0.840418
2	Ridge	0.852677	0.838269
3	DecisionTreeRegressor	0.908531	0.882759
4	RandomForestRegressor	0.950037	0.834148

⇨ The final data frame provides an opinion on model scores, and the charts assist us in determining which models are more successful.

IX. Conclusion

It was hoped to gain new viewpoints by running different models and then comparing their performance. The goal of this study was to forecast used automobile prices. The dataset was unearthed, and characteristics were thoroughly investigated using data visualizations and exploratory data analysis. The relationship between characteristics was investigated. Predictive models were used in the last step to forecast automobile prices.

REFERENCES

Brownlee, J. (2020, August 27). *Recursive feature elimination (RFE) for feature selection in Python*. Machine Learning Mastery. Retrieved May 10, 2022, from https://machinelearningmastery.com/rfe-feature-selection-in-python/

Car price prediction using Machine Learning Techniques. (n.d.). Retrieved May 10, 2022, from https://temjournal.com/content/81/TEMJournalFebruary2019_113_118.pdf

Gokce, E. (2020, January 10). *Predicting used car prices with machine learning techniques*. Medium. Retrieved May 10, 2022, from https://towardsdatascience.com/predicting-used-car-prices-with-machine-learning-techniques-8a9d8313952

goyalshalini93. (2019, April 26). *Car price prediction (linear regression - RFE)*. Kaggle. Retrieved May 10, 2022, from https://www.kaggle.com/code/goyalshalini93/car-price-prediction-linear-regression-rfe/notebook

New cars, car prices, buy & Sell used cars in India. CarDekho. (n.d.). Retrieved May 10, 2022, from https://www.cardekho.com/

Patil, P. (2021, December 18). *What is exploratory data analysis?* Medium. Retrieved May 10, 2022, from https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15

Robinson, S. (2021, June 7). *Linear regression in python with scikit-learn*. Stack Abuse. Retrieved May 10, 2022, from https://stackabuse.com/linear-regression-in-python-with-scikit-learn/

Technicallife-sujeet/linear-regression-prediction. Jovian. (n.d.). Retrieved May 10, 2022, from https://jovian.ai/technicallife-sujeet/linear-regression-prediction

Zach. (2021, September 28). *How to perform one-hot encoding in Python*. Statology. Retrieved May 10, 2022, from https://www.statology.org/one-hot-encoding-in-python/