Used Car Selling Price - Linear Regression

Artificial Intelligence CS-617-A

Avalons

Sacred Heart University
School of Computer Science & Engineering
The Jack Welch College of Business & Technology

Submitted To:
Dr. Reza Sadeghi

Fall 2022

Project Report of Used Car Selling Price - Linear Regression

Team Name

Name of the Team **Avalons**

Team Members

1. Sambasiva Rao Chennamsetty chennamsettys@mail.sacredheart.edu(Team Head)
2. Arif Pasha Shaik shaiks11@mail.sacredheart.edu(TM)
3. Jagadishwar Reddy Velma velmaj@mail.sacredheart.edu (TM)
4. Sai Hrithik Peddi peddis4@mail.sacredheart.edu (TM)
5. Vamsi Kiran Kakkera kakkerav@mail.sacredheart.edu (TM)

Description of Team Members

1. **Sambasiva Rao Chennamsetty**

   I completed my Bachelor's in Information Technology. I had 3+ years of experience as a full-stack developer with Java programming as a backend. I like to work with a team with more commitment to work.

2. **Arif Pasha Shaik**

   I have completed my Bachelor's in Information Technology, I have done a couple of internships on Visual Basic .net, and Business Analytics: Data mining and Data warehousing.And I love working in a team that has its full dedication.

3. **Jagadishwar Reddy Velma**

   I hold 7+ years of experience in SQL Database Administration. I am here to learn and improve better development skills which help me to become an extensive experienced Core Developer.

4. **Sai Hrithik Peddi**

   I am a graduate student at sacred Heart University. I have completed my Undergraduate in Computer Science. After, I worked as an Android Developer at Sensorise Digital services for 6 months. I'm very passionate about my work role.

5. **Vamsi Kiran Kakkera**

   I have done my Bachelor's degree in the stream of computer science. I'm having work Experience of 2.5 years in the AWS cloud as an Associate Developer. I've chosen this team as they are very coordinative and discuss everything with the team members.

1 Table of Contents

1 Introduction

As the world evolving in all directions significantly, the economic gaps between the people are still exist. The livelihood of different people from different financial backgrounds are changing a lot. When it comes to the comfortable travel the cars are playing a vital role. Also, considering the COVID pandemic, most of the lower- and middle-income group of people also attracting to travel in a safe environment and not willing to choose public transport.

- At the same time the car manufacturers also increased the price of the new cars, which is directly affecting the buying capability of low-income group people.
- Hence, most of the people are looking at the used cars now.
- There are few people who cannot afford to buy new luxury car, but they wish to travel in it. For those, this used cars are the sunlight in dark. [1]
- This used cars has become an opportunity for the business. And it's going to generate a decent revenue for business as well.

1.1 Research Question

- Which variables are significant in predicting the price of a used car?
- How well those variables describe the price of a car?

1.2 GitHub Repository

https://github.com/samba-chennamsetty/used-car-selling-price-linear-regression

2 Dataset Description

2.1 URL of Dataset

Old Car Selling Price with Linear Regression | Kaggle [2]

2.2 Dataset Explanation

- This dataset contains information about used cars listed on www.cardekho.com [3]
- This data can be used for a lot of purposes such as price prediction to exemplify the use of linear regression in Machine Learning.

2.3 Features of Dataset

The columns are in the given dataset is as follows:

1. **Car_Name:** This column should be filled with the name of the car.
2. **Year:** This column should be filled with the year in which the car was bought.
3. **Selling_Price:** This column should be filled with the price the owner wants to sell the car at.
4. **Present_Price:** This is the current ex-showroom price of the car.
5. **Kms_Driven:** This is the distance completed by the car in km.
6. **Fuel_Type:** Fuel type of the car.
7. **Seller_Type:** Defines whether the seller is a dealer or an individual.
8. **Transmission:** Defines whether the car is manual or automatic.
9. **Owner:** Defines the number of owners the car has previously had.

3 Related Work

3.1 Pro's

The advantages we have over the other related works are

- We are following Linear Regression to make it to the point for easy analysis.
- Considering the best prediction relational fields.
- Portraying the many visualities of impact with each feature.
- Plan to build multiple models based on companies.
- We use better data-cleaning techniques.

3.2 Con's

The disadvantages we have over the other related works are

- We don't have multiple regression i.e, based on many features compared to the source project we referred.

4 Project Plan

The project plan has the below steps in it.

1. Data-preprocessing
2. Model building
3. Optimizing Model
4. Model Evaluation

4.1 Data Preprocessing

We import the dataset initially as below and look for the head rows in the dataset

Step 1: Reading and Understanding the Data

Let's start with the following steps:

- Importing data using the pandas library
- Understanding the structure of the data

In [1]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
car = pd.read_csv("UsedCarDetails.csv")
```

In [4]:
```
car.head()
```

Out[4]:

	name	year	selling_price	km_driven	fuel	seller_type	transmission	owner
0	Maruti 800 AC	2007	60000	70000	Petrol	Individual	Manual	First Owner
1	Maruti Wagon R LXI Minor	2007	135000	50000	Petrol	Individual	Manual	First Owner
2	Hyundai Verna 1.6 SX	2012	600000	100000	Diesel	Individual	Manual	First Owner
3	Datsun RediGO T Option	2017	250000	46000	Petrol	Individual	Manual	First Owner
4	Honda Amaze VX i-DTEC	2014	450000	141000	Diesel	Individual	Manual	Second Owner

- Using the shape function, we find the number of rows and columns in the dataset.
- We use columns function to view the columns in the function.
- We use info function to know all the details of the car data set with their datatype.

In [8]:
```
car.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4340 entries, 0 to 4339
Data columns (total 7 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   selling_price  4340 non-null   int64 
 1   km_driven      4340 non-null   int64 
 2   fuel           4340 non-null   object
 3   seller_type    4340 non-null   object
 4   transmission   4340 non-null   object
 5   owner          4340 non-null   object
 6   age            4340 non-null   int64 
dtypes: int64(3), object(4)
memory usage: 237.5+ KB
```

In [9]:
```
car.shape
```

Out[9]:
```
(4340, 7)
```

In [10]:
```
car.columns
```

Out[10]:
```
Index(['selling_price', 'km_driven', 'fuel', 'seller_type', 'transmission',
       'owner', 'age'],
      dtype='object')
```

Data Cleaning

Adding a new variable for calculating the age of the car.
As part of this we clean the unwanted data and make the data right and good for the model with error free.

Adding new variable

In [5]:
```
# adding new variable 'current-year' to the car dataframe to calculate car age.
car['current']= 2022
```

Adding new variable age column

In [6]:
```
# calculating current age.
car['age']=car['current']-car['year']

car.head()
```

Out[6]:

	name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	current	age
0	Maruti 800 AC	2007	60000	70000	Petrol	Individual	Manual	First Owner	2022	15
1	Maruti Wagon R LXI Minor	2007	135000	50000	Petrol	Individual	Manual	First Owner	2022	15
2	Hyundai Verna 1.6 SX	2012	600000	100000	Diesel	Individual	Manual	First Owner	2022	10
3	Datsun RediGO T Option	2017	250000	46000	Petrol	Individual	Manual	First Owner	2022	5
4	Honda Amaze VX i-DTEC	2014	450000	141000	Diesel	Individual	Manual	Second Owner	2022	8

Step 2 : Data Cleaning and Preparation

Drop all non required or repeative data

In [7]:
```
car.drop(['current','year','name'],axis=1,inplace=True)
car.head()
```

Out[7]:

	selling_price	km_driven	fuel	seller_type	transmission	owner	age
0	60000	70000	Petrol	Individual	Manual	First Owner	15
1	135000	50000	Petrol	Individual	Manual	First Owner	15
2	600000	100000	Diesel	Individual	Manual	First Owner	10
3	250000	46000	Petrol	Individual	Manual	First Owner	5
4	450000	141000	Diesel	Individual	Manual	Second Owner	8

Duplicate Data Check

Checking if there is any duplicate data and dropping the entire duplicate row if any

Duplicate Check

In [14]:
```
car_dub=car.copy()
# Checking for duplicates and dropping the entire duplicate row if any
car_dub.drop_duplicates(subset=None, inplace=True)
```

In [15]:
```
car_dub.shape
```

Out[15]:
```
(3498, 7)
```

In [16]:
```
car.shape
```

Out[16]:
```
(4340, 7)
```

Insights

- The shape after running the drop duplicate command is not same as the original dataframe.

Identifying junk values

Checking value_counts() for entire dataframe.

This will help to identify any Unknow/Junk values present in the dataset.

In [20]:
```
for col in car:
    print(car[col].value_counts(ascending=False), '\n\n\n')
```

```
300000     122
250000     107
350000     104
550000      82
150000      81
          ...
2595000      1
368000       1
248000       1
641000       1
865000       1
Name: selling_price, Length: 445, dtype: int64



70000     202
80000     197
120000    192
60000     189
50000     171
         ...
35925       1
40771       1
30500       1
55800       1
112198      1
Name: km_driven, Length: 770, dtype: int64



Diesel      1762
Petrol      1676
CNG           37
LPG           22
Electric       1
Name: fuel, dtype: int64
```

```
Individual          2753
Dealer               712
Trustmark Dealer      33
Name: seller_type, dtype: int64


Manual       3187
Automatic     311
Name: transmission, dtype: int64


First Owner             2157
Second Owner             964
Third Owner              285
Fourth & Above Owner      75
Test Drive Car            17
Name: owner, dtype: int64


5     336
10    332
7     327
8     315
9     290
4     285
6     273
11    244
12    205
13    167
3     156
14    127
15    114
16     93
17     60
2      45
18     37
19     22
20     17
21     16
22     12
24      9
23      9
25      3
26      2
27      1
30      1
Name: age, dtype: int64
```

Insights

- There seems to be no Junk/Unknown values in the entire dataset.

* We found that there is no Junk or Unknown values exists in the data set.

Data Exploration

Univariate Analysis:

Univariate analyses are used extensively in quality-of-life research. Univariate analysis is defined as analysis carried out on only one (“uni”) variable (“variate”) to summarize or describe the variable. However, another use of the term “univariate analysis” exists and refers to statistical analyses that involve only one dependent variable and which are used to test hypotheses and draw inferences about populations based on samples, also referred to as univariate.

We find the univariate using distplot and boxplot graphs with below code. Here we’re using only uni one feature for the analysis.

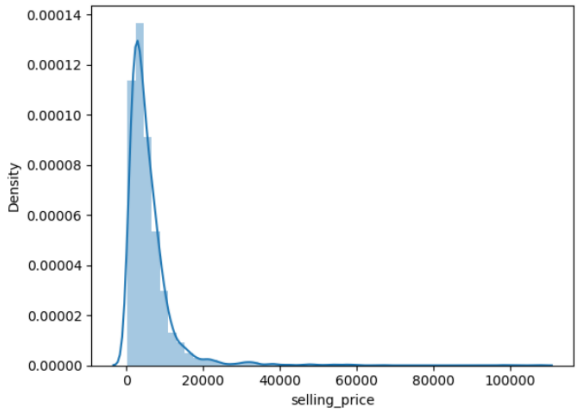
Step 2 : Data Visualization

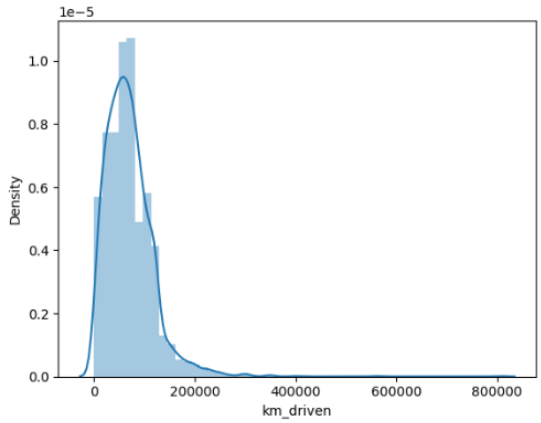
Univaraite Analysis

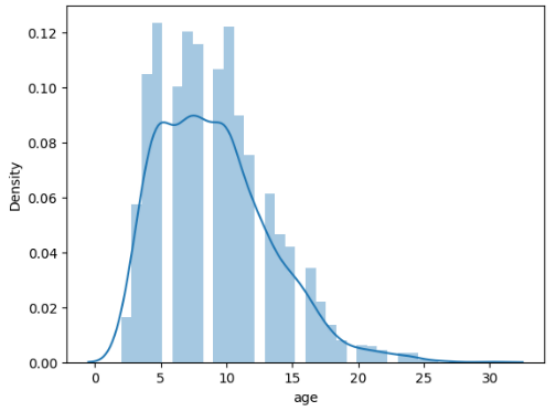
In [23]:
```
# variables for plotting.
cont_col =[ 'selling_price','km_driven', 'age']
```

distplot

In [24]:
```
for i in cont_col:
    sns.distplot(car[i])
    plt.show()
```





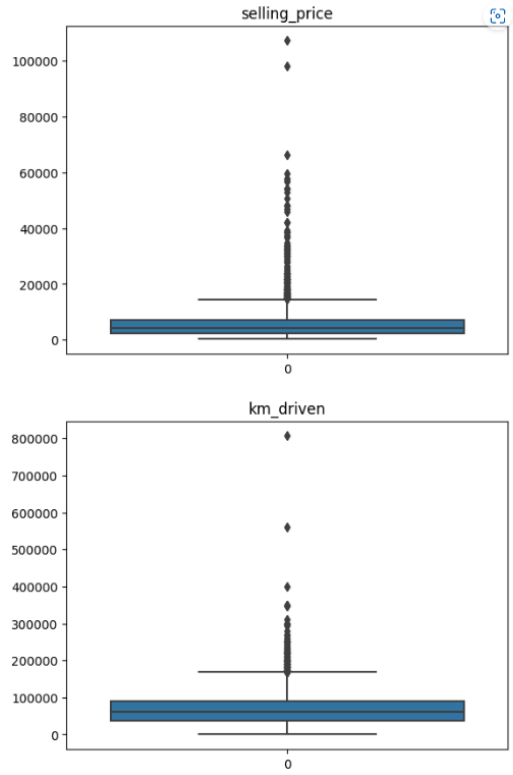


Using distplot function Univariate Analysis has been made which gives a similar kind of distribution, some features are showing nearby normal distribution while some are skewed.

Boxplot

boxplot

```
In [25]: for i in cont_col:
             plt.title(i)
             sns.boxplot(car[i])
             plt.show()
```



Bivariate Analysis:

Bivariate analysis refers to the analysis of two variables to determine relationships between them. Bivariate analyses are often reported in quality-of-life research. For an excellent example of research that utilizes bivariate analyses and demonstrates how the results of bivariate analyses can be used to inform furthermore complex analyses.

We find the relation between Selling Price and Car age which is bi with scatter plotting

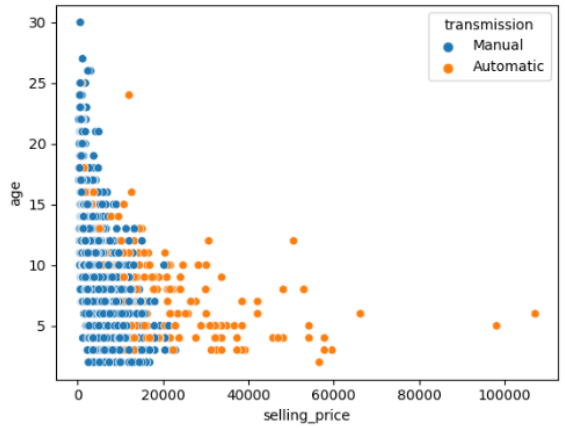
Scatter Plot:

Bivaraite Analysis

```
In [26]: # variables for plotting
         cont_col =[ 'selling_price','km_driven', 'fuel']
```

scatter plot

```
In [27]: # plotting graph b/w count and continuous columns taking transmission as hue
         for i in cont_col:
             sns.scatterplot(data = car[i], x = car[i], y = car['age'], hue=car['transmission'])
             plt.show()
```



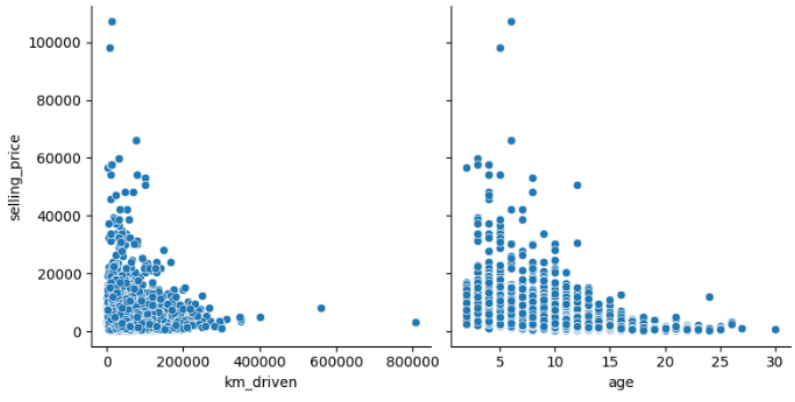
PairPlot:

Here we took selling price and compare it with km driven and age of car.

pair plot

```
In [28]: def pp(x,y):
             sns.pairplot(car, x_vars=[x,y], y_vars='selling_price',size=4, aspect=1, kind='scatter')
             plt.show()

         pp('km_driven', 'age')
```



5 GitHub Repository

https://github.com/samba-chennamsetty/used-car-selling-price-linear-regression

6 References

[1] https://www.kaggle.com/code/gauravduttakiit/old-car-selling-price-with-linear-regression

[2] https://www.kaggle.com/code/gauravduttakiit/old-car-selling-price-with-linear-regression/data?select=car+data.csv

[3] www.cardekho.com