USED CARS PRICE PREDICTION

PAULAMI SANYAL

JUNE 2022

OBJECTIVE

- THE MAIN OBJECTIVE OF THIS ANALYSIS FOCUSES ON THE PREDICTION OF PRICE OF USED CARS.
- RUNNING TO USED CAR DEALERS JUST TO GET THE OPTIMAL PRICE OF THAT OLD CAR SELLERS NEED TO GATHER MIGHT BE AN UNNECESSARY TROUBLE.
- OUR OBJECTIVE IS TO BUILD A MODEL THAT COULD PREDICT THE OPTIMAL PRICE THAT THEY MIGHT RECEIVE WHILE SELLING THEIR OLD CARS.

DESCRIPTION OF THE DATA

THE DATASET CONSISTS OF 13 COLUMNS: 12 FEATURES AND ONE TARGET 'PRICE'.

```
In [8]:  1  df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6019 entries, 0 to 6018
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Name               6019 non-null   object 
 1   Location           6019 non-null   object 
 2   Year               6019 non-null   int64  
 3   Kilometers_Driven  6019 non-null   int64  
 4   Fuel_Type          6019 non-null   object 
 5   Transmission       6019 non-null   object 
 6   Owner_Type         6019 non-null   object 
 7   Mileage            6017 non-null   object 
 8   Engine             5983 non-null   object 
 9   Power              5983 non-null   object 
 10  Seats              5977 non-null   float64
 11  New_Price          824 non-null    object 
 12  Price              6019 non-null   float64
dtypes: float64(2), int64(2), object(9)
memory usage: 611.4+ KB
```

SUMMARY OF DATASET

THE FEATURES OF THE DATASET ARE:

1. **NAME** – MODEL OF THE CAR
2. **LOCATION** – CITY WHERE THE CAR IS SOLD
3. **YEAR** – YEAR OF MANUFACTURE
4. **KILOMETERS_DRIVEN** – NUMBER OF KILOMETRES THE CAR HAS BEEN DRIVEN
5. **FUEL_TYPE** – THE TYPE OF FUEL THE CAR RUNS ON
6. **TRANSMISSION**
7. **OWNER_TYPE** – WHETHER THE CAR HAS BEEN SOLD FIRST, SECOND, THIRD, FOURTH OR MORE NUMBER OF TIMES BEFORE
8. **MILEAGE** – HOW MANY KILOMETRES DO THE CAR RUN PER LITRE OF FUEL
9. **ENGINE** – THE CAPACITY OF THE ENGINE IN CC (CUBIC CENTIMETRES)
10. **POWER** – THE POWER OF THE ENGINE
11. **SEATS** – NUMBER OF SEATS IN THE CAR
12. **NEW_PRICE** – PRICE OF A NEW CAR
13. **PRICE** – CURRENT PRICE OF THE CAR. THIS IS OUR TARGET VARIABLE.

DATA EXPLORATION

There are 6019 data points and 12 features with 4 numeric and 9 object data types.

In [6]:
```
1 df.nunique()
```

Out[6]:
```
Name                 1876
Location               11
Year                   22
Kilometers_Driven    3093
Fuel_Type               5
Transmission            2
Owner_Type              4
Mileage               442
Engine                146
Power                 372
Seats                   9
New_Price             540
Price                1373
dtype: int64
```

In [7]:
```
1 df.describe()
```

Out[7]:

	Year	Kilometers_Driven	Seats	Price
count	6019.000000	6.019000e+03	5977.000000	6019.000000
mean	2013.358199	5.873838e+04	5.278735	9.479468
std	3.269742	9.126884e+04	0.808840	11.187917
min	1998.000000	1.710000e+02	0.000000	0.440000
25%	2011.000000	3.400000e+04	5.000000	3.500000
50%	2014.000000	5.300000e+04	5.000000	5.640000
75%	2016.000000	7.300000e+04	5.000000	9.950000
max	2019.000000	6.500000e+06	10.000000	160.000000

DATA CLEANING

- There are 5195 null values for the feature New_Price our of 6019 datapoints. So, we remove this feature entirely.
- We also remove all the null valued rows from the features Mileage, Engine, Power, & Seats

```
1  df.isna().sum()
```

```
Name                    0
Location                0
Year                    0
Kilometers_Driven       0
Fuel_Type               0
Transmission            0
Owner_Type              0
Mileage                 2
Engine                 36
Power                  36
Seats                  42
New_Price            5195
Price                   0
dtype: int64
```

FEATURE ENGINEERING

- WE DERIVED THE NAME OF THE COMPANY & THE MODEL OF THE CAR, WHILE REMOVING THE NAME COLUMN
- AS OBJECT DATATYPES CANNOT BE USED IN THE MODELLING WE CONVERTED THEM TO NUMERIC VALUES USING ONE-HOT-ENCODING – FOR COLUMNS LOCATION, FUEL_TYPE, TRANSMISSION, OWNER_TYPE, COMPANY & MODEL

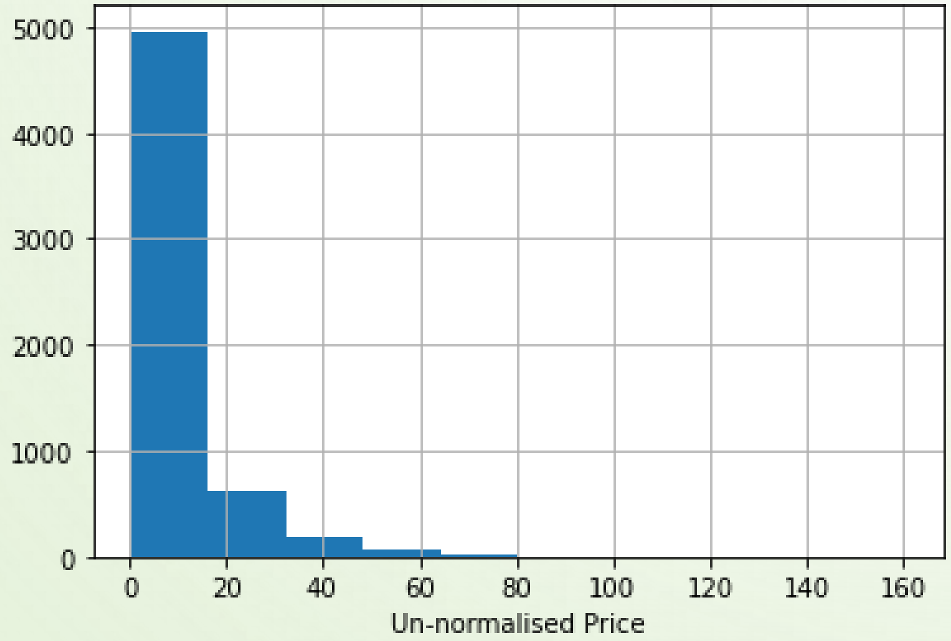
ONE HOT ENCODING - for the columns Location, Fuel_Type, Transmission, Owner_Type, and newly created features Company & Model

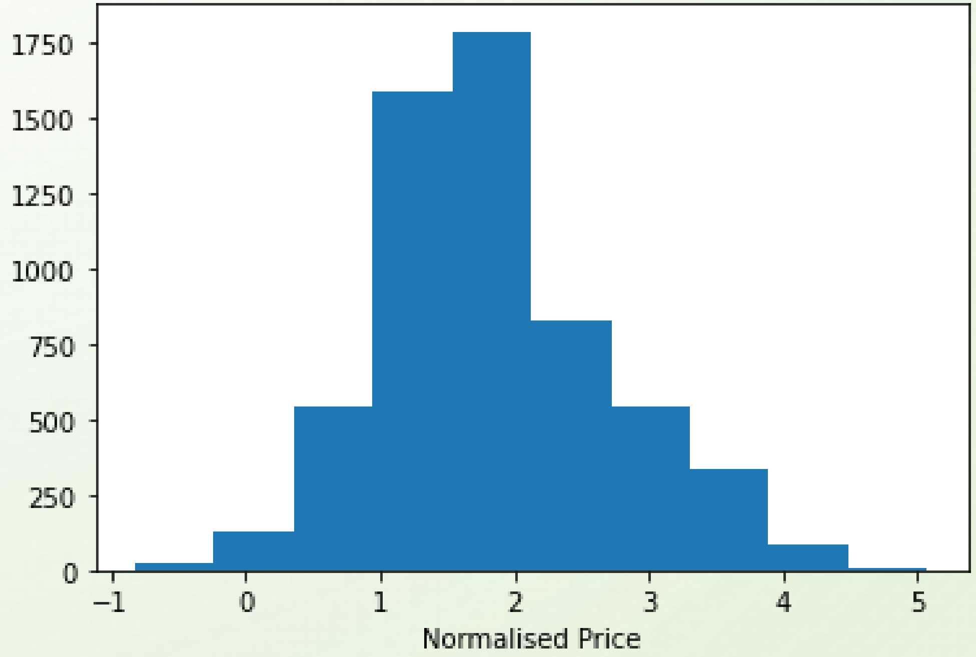
```
dummies = pd.get_dummies(df[['Location','Fuel_Type','Transmission','Owner_Type','Company','Model']])
dummies.head()
```

	Location_Ahmedabad	Location_Bangalore	Location_Chennai	Location_Coimbatore	Location_Delhi	Location_Hyderabad	Location_Jaipur	Locat
0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	
2	0	0	1	0	0	0	0	
3	0	0	1	0	0	0	0	
4	0	0	0	1	0	0	0	

FEATURE ENGINEERING

- OUR TARGET VARIABLE PRICE WAS RIGHT-SKEWED. WE USED LOG-TRANSFORMATION TO NORMALIZE IT. THIS IMPROVED BOTH THE ERROR VALUES AND R-SQUARE VALUES AND GAVE US AN IMPROVED MODEL.





SUMMARY OF LINEAR REGRESSION MODELS

- SIMPLE LINEAR REGRESSION – SIMPLE LINEAR REGRESSION MODEL HAD A R-SQUARE VALUE OF ABOUT 81%
- LINEAR REGRESSION WITH POLYNOMIAL EFFECTS – MY SYSTEM HAD MEMORY RESTRICTIONS HENCE I HAD TO BUILD THIS ONE WITH POLYNOMIAL OF DEGREE ONE, WHICH IS SAME AS SIMPLE LINEAR REGRESSION AND HAD A R-SQUARE VALUE OF 81% AS WELL
- REGULARIZATION REGRESSION – WE APPLIED RIDGE, LASSO & ELASTIC NET REGULARIZATION TECHNIQUES FOR MODEL BUILDING AND FOUND THAT ELASTIC NET REGRESSION HAD THE LEAST ROOT MEAN SQUARE ERROR AND THE HIGHEST R-SQUARE VALUE OF ALL THREE.

MODEL RECOMMENDATION

- IN TERMS OF BOTH ROOT MEAN SQUARE ERROR AND R-SQUARE VALUE, WE FIND THAT REGRESSION WITH ELASTIC NET REGULARIZATION TO BE THE BEST ALTERNATIVE AMONGST SIMPLE LINEAR REGRESSION, RIDGE REGRESSION, LASSO REGRESSION AND ELASTIC NET REGRESSION.
- EVEN IN STOCHASTIC GRADIENT DESCENT, ELASTIC NET HAD BETTER VALUES THAN THE REST OF THE MODELS.

	RMSE	R2	RMSE-SGD	R2-SGD
Linear	0.216189	0.936538	9.585133e+16	-1.247499e+34
Lasso	0.215193	0.937122	2.031819e+19	-5.605498e+38
Ridge	0.212020	0.938962	8.748683e+17	-1.039272e+36
ElasticNet	0.212006	0.938971	1.856228e+18	-4.678501e+36

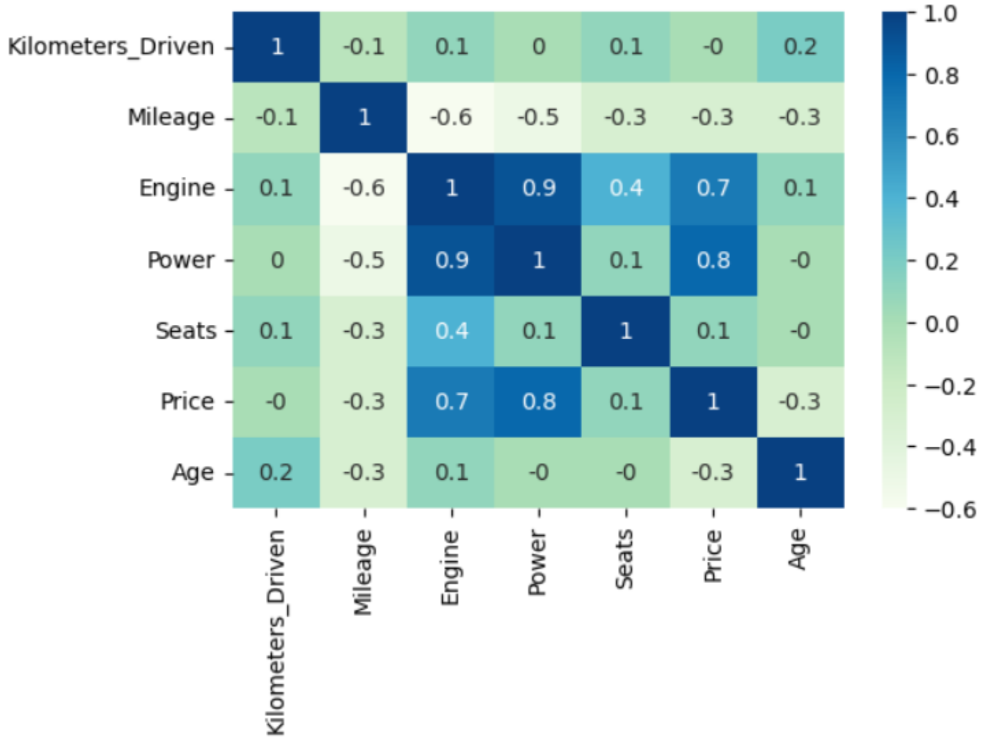
KEY FINDINGS & INSIGHTS

- From the heatmap created we find that the features Power & Engine most positively correlated, followed by Power & Price, and Engine & Price.

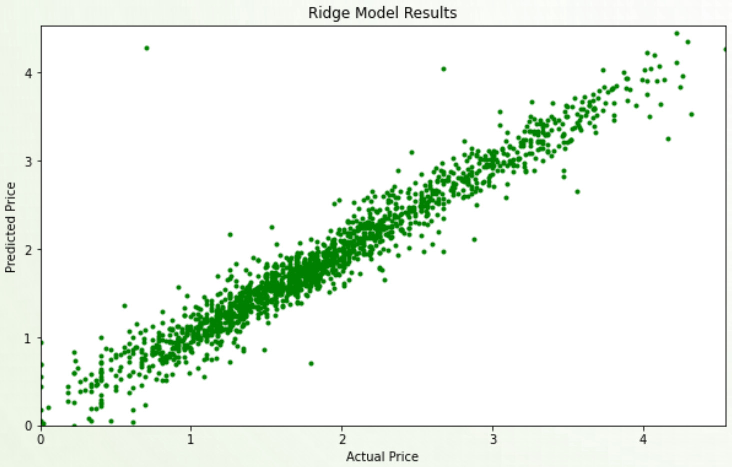
- So, we can say that Power & Engine are the most important features that decides the valuation of the car.

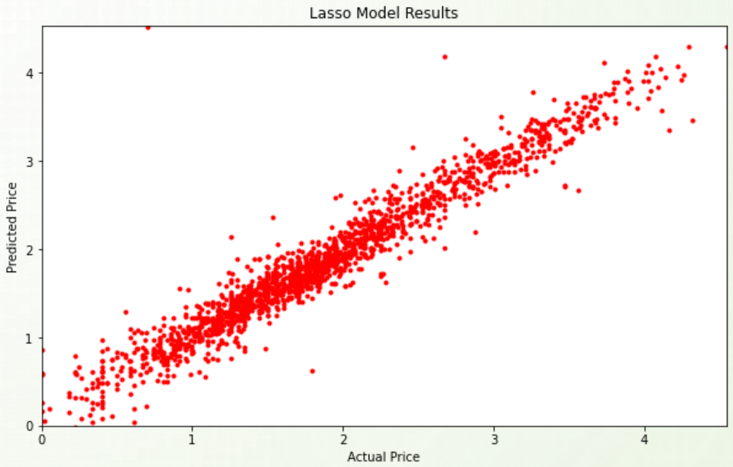
Correlations between the features

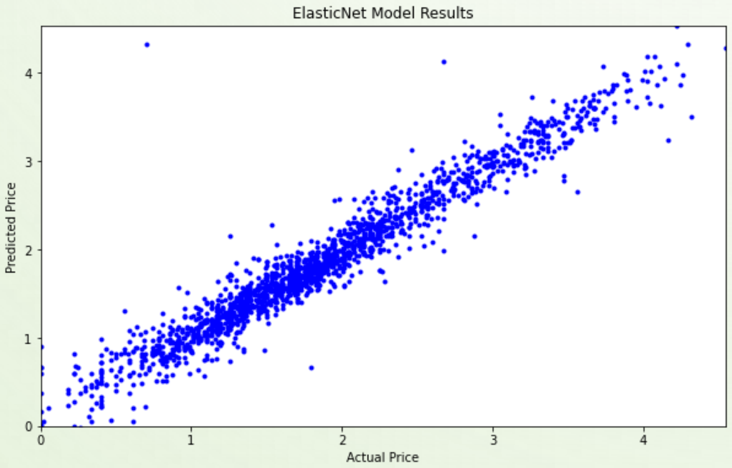
```
# Studying the corellations between features using Heat Map!
plt.figure(dpi=100)
sns.heatmap(np.round(df.corr(),1),annot=True, cmap="GnBu")
plt.show()
```

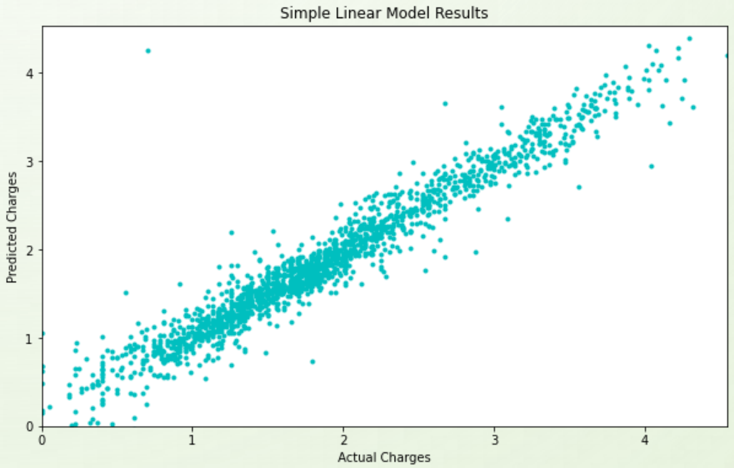


KEY FINDINGS & INSIGHTS









KEY FINDINGS & INSIGHTS

- NORMALIZING THE TARGET VARIABLE PRICE MADE THE EVALUATION LINEARLY DISTRIBUTED THROUGH ALL THE MODELS – LINEAR REGRESSION, RIDGE REGRESSION, LASSO REGRESSION AND ELASTIC NET REGRESSION.
- HOWEVER, ELASTIC NET HAD THE BEST VALUES OF ALL OF THEM, AND HENCE IS RECOMMENDED.

SUGGESTIONS FOR NEXT STEPS

- LINEAR REGRESSION RAN VERY SMOOTHLY AND VERY FAST. HOWEVER, THE REGULARIZATION TOOK MUCH MORE TIME FOR TRAINING THE MODELS, BUT PROVIDED BETTER OUTCOMES IN TERMS OF ERROR AND R-SQUARE VALUES.
- **PERSONALLY, I WOULD'VE ADDED A FEATURE OF HOW MANY TIMES THE CAR WAS TAKEN FOR SERVICING SINCE BEING SOLD THE FIRST TIME.** THIS FEATURE MIGHT IMPROVE THE MODEL BUILDING AS WELL AS EVALUATION TO SOME EXTENT. THIS FEATURE MIGHT BE ADDED TO THE MODEL AND EVALUATED FURTHER.

REFERENCES

- KAGGLE.COM
- GITHUB.COM

THANK YOU!

IBM Machine Learning Professional Certificate

Supervised Machine Learning: Regression

By PAULAMI SANYAL