I

Hardianto Tandi Seno

hardiantotandiseno@gmail.com

Predict Car Price with Multiple Linear Regression

https://github.com/hardiantots/linear_reg_predictprice

ABOUT PRESENTATION

Contains data science projects where the case is to **know the prediction of car prices** using **multiple linear regression**. This **algorithm is used** because the **predictor variable** that will be used to determine car price predictions is **more than one.**

ABOUT DATASET

The dataset used is the **Toyota Corolla Dataset**, with a total of **38 columns & 1436 rows.**

UNDERSTANDING DATA

Knowing Shape of Dataframe

Rows : 1436 ; Columns : 38

Knowing a few columns that are important

["Price", "Age_08_04", "KM", "HP", "cc", "Doors", "Gears", "Quarterly_Tax", "Weight"]

- **Price :** Offer Price in EUROs
- **Age_08_04 :** Age in months as in August 2004
- **KM :** Accumulated Kilometers on odometer
- **HP :** Horse Power
- **cc :** Cylinder Volume in cubic centimeters
- **Doors :** Number of doors
- **Gears** : Number of gear positions
- **Quarterly_Tax :** Quarterly road tax in EUROs
- **Weight :** Weight in Kilograms

See descriptive summary of the dataframe

	Price	Age_08_04	KM	HP	cc	Doors	Gears	Quarterly_Tax	Weight
count	1436.000000	1436.000000	1436.000000	1436.000000	1436.00000	1436.000000	1436.000000	1436.000000	1436.00000
mean	10730.824513	55.947075	68533.259749	101.502089	1576.85585	4.033426	5.026462	87.122563	1072.45961
std	3626.964585	18.599988	37506.448872	14.981080	424.38677	0.952677	0.188510	41.128611	52.64112
min	4350.000000	1.000000	1.000000	69.000000	1300.00000	2.000000	3.000000	19.000000	1000.00000
25%	8450.000000	44.000000	43000.000000	90.000000	1400.00000	3.000000	5.000000	69.000000	1040.00000
50%	9900.000000	61.000000	63389.500000	110.000000	1600.00000	4.000000	5.000000	85.000000	1070.00000
75%	11950.000000	70.000000	87020.750000	110.000000	1600.00000	5.000000	5.000000	85.000000	1085.00000
max	32500.000000	80.000000	243000.000000	192.000000	16000.00000	5.000000	6.000000	283.000000	1615.00000

Check Missing Value

```
Price      0
Age        0
KM         0
HP         0
CC         0
Doors      0
Gears      0
QT         0
Weight     0
dtype: int64
```

Check & Remove Duplicated Data

Duplicated data :

	Price	Age	KM	HP	CC	Doors	Gears	QT	Weight
113	24950	8	13253	116	2000	5	5	234	1320

After remove duplicated data :

Price	Age	KM	HP	CC	Doors	Gears	QT	Weight

Define Discrete & Continuous feature from Dataset

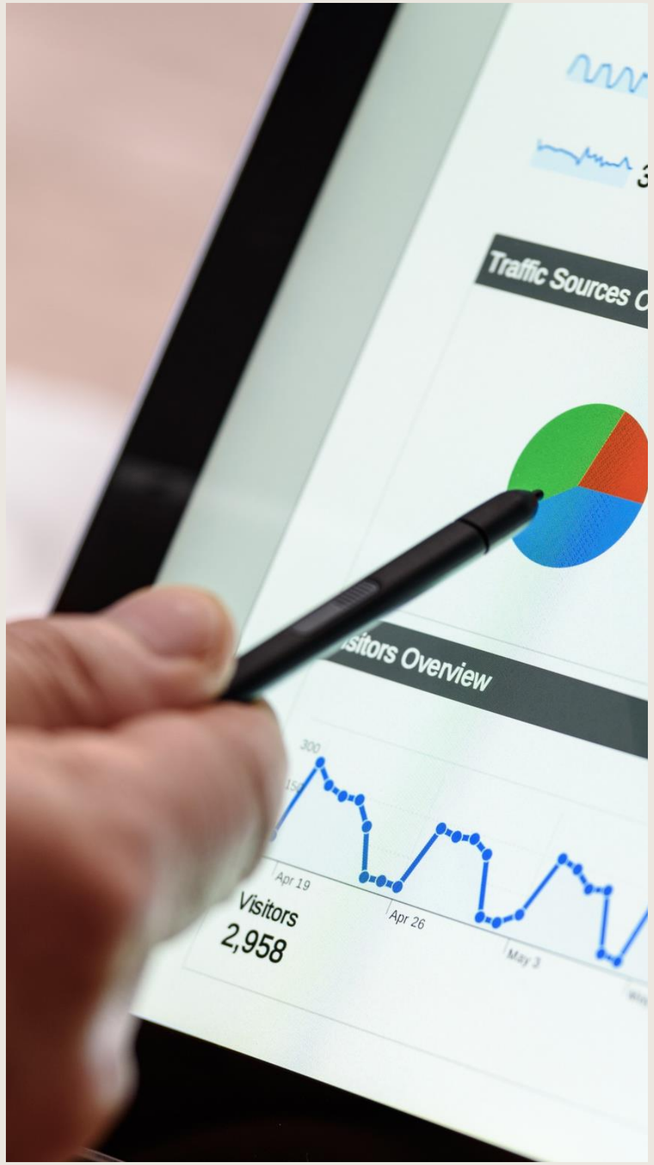
Discrete data is a type of data that **has clear spaces and points between values** and **can be calculated.**
Continuous data is data whose **value has unlimited possible values within a certain range** and **can be measured.**

Discrete data from new dataset :

```
['HP', 'CC', 'Doors', 'Gears', 'QT']
```

Continuous data from new dataset :

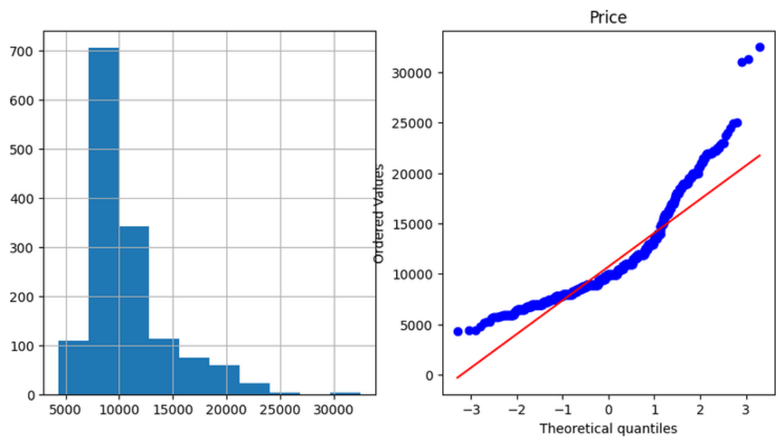
```
['Price', 'Age', 'KM', 'Weight']
```

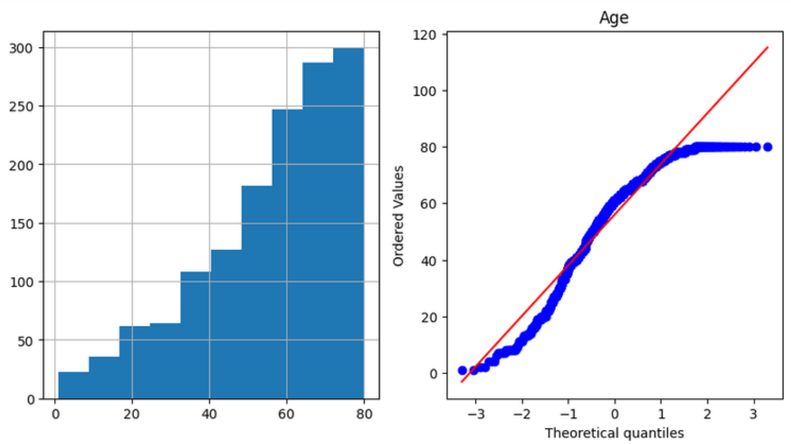


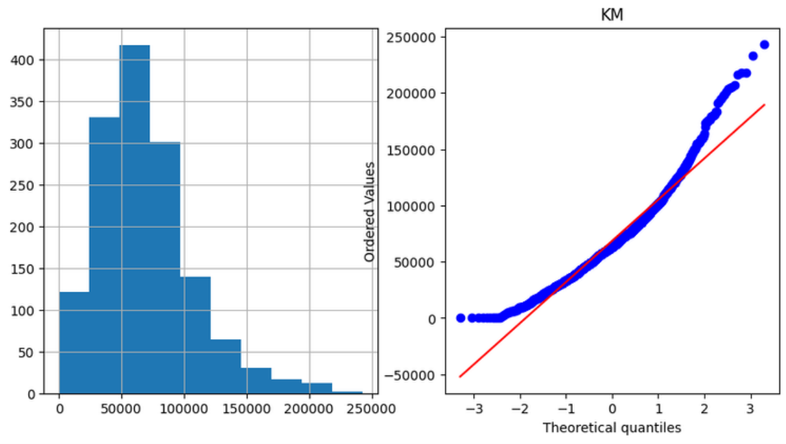
EXPLORATORY DATA ANALYSIS

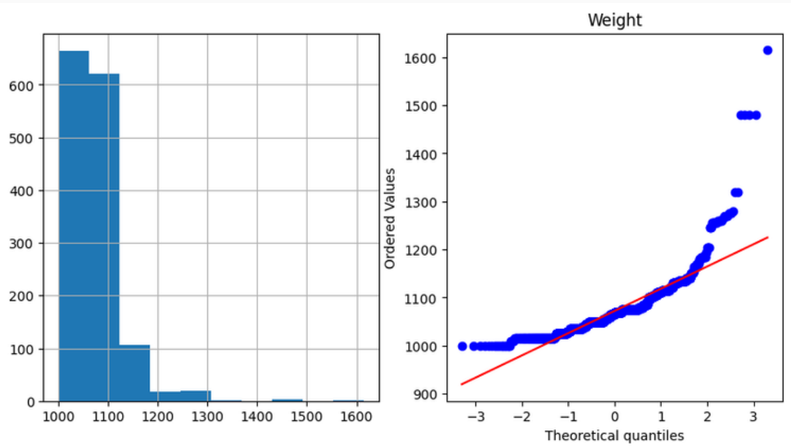
VISUALIZE DISTRIBUTION CONTINUOUS FEATURES

Histogram & Probability Plot





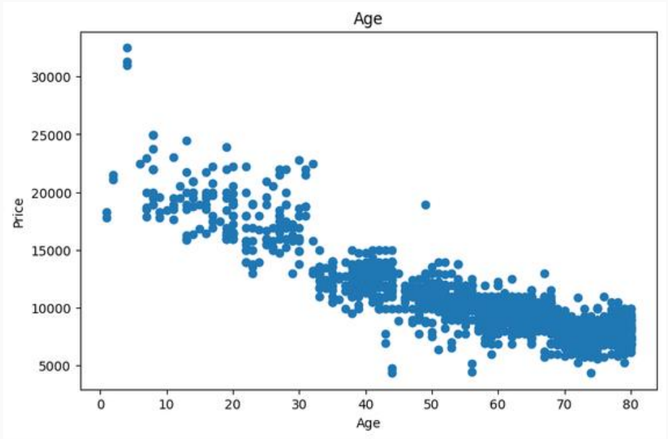


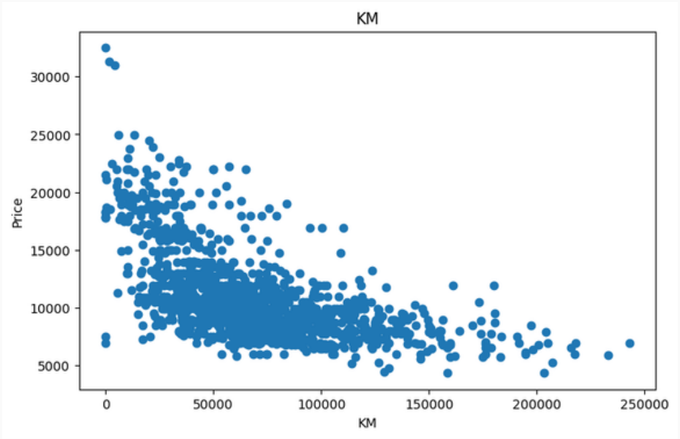


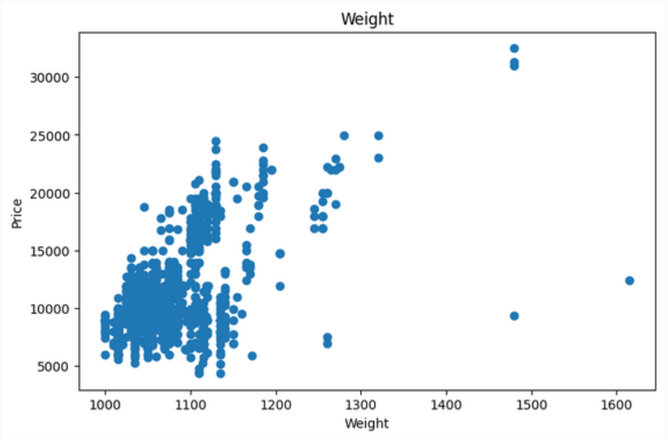
- **Summary from Visualization**

  Sample data from continuous features **don't come from a normal distribution.**

VISUALIZE RELATIONSHIP BETWEEN INDEPENDENT FEATURE & DEPENDENT FEATURE (PRICE)



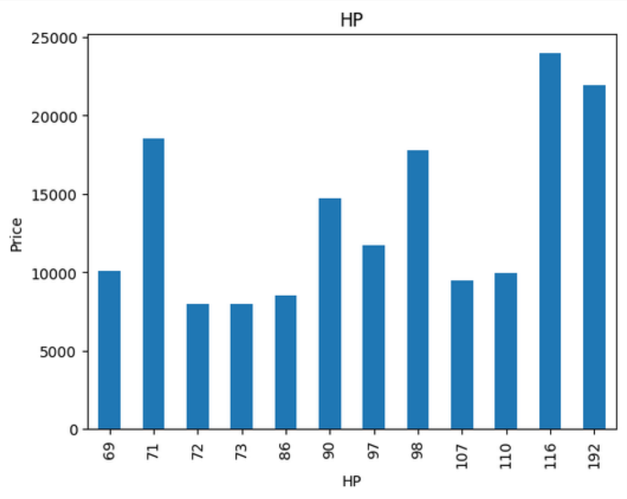


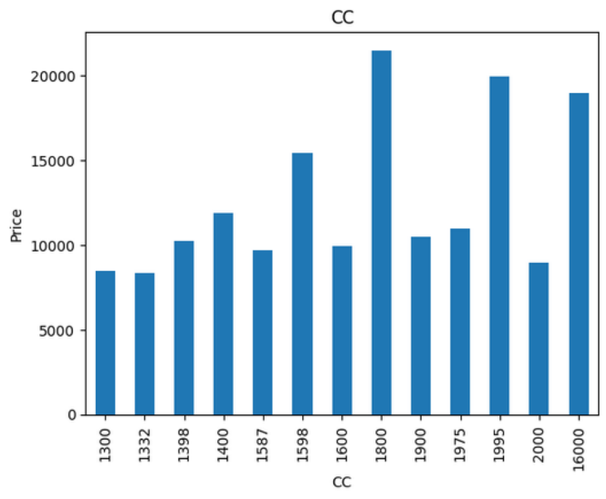


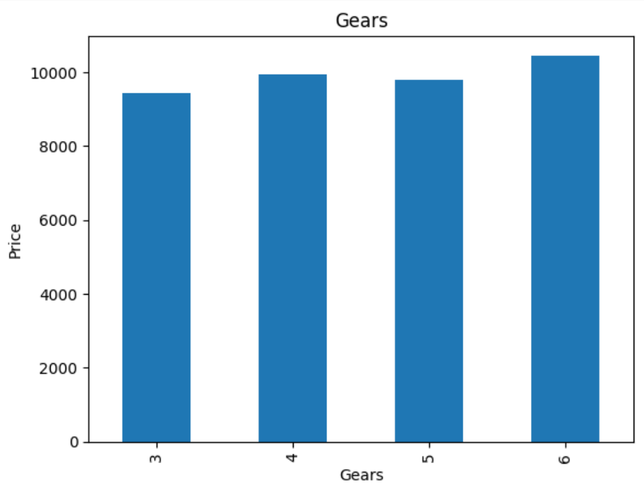
- **Summary from Visualization**

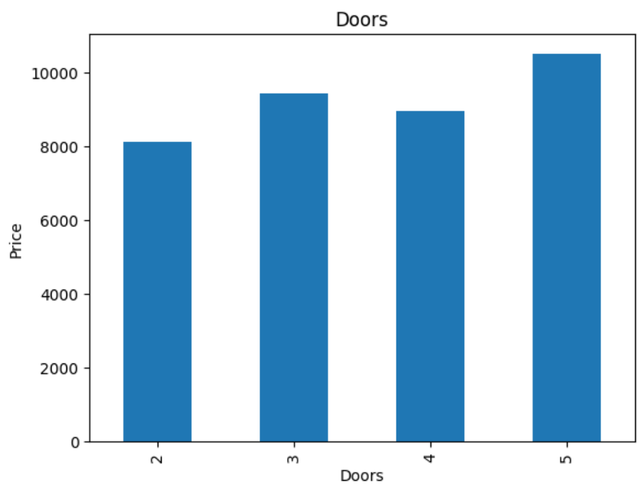
  Age have **a good linear relation with Price (Negative Correlation)** between other Independent Feature

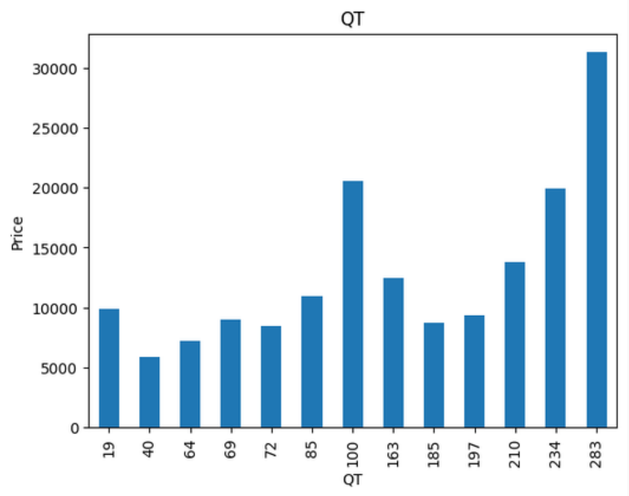
ANALYZE RELATIONSHIP BETWEEN DISCRETE FEATURE & DEPENDENT FEATURE (PRICE)











ANALYZE RELATIONSHIP BETWEEN DISCRETE FEATURE & DEPENDENT FEATURE (PRICE)

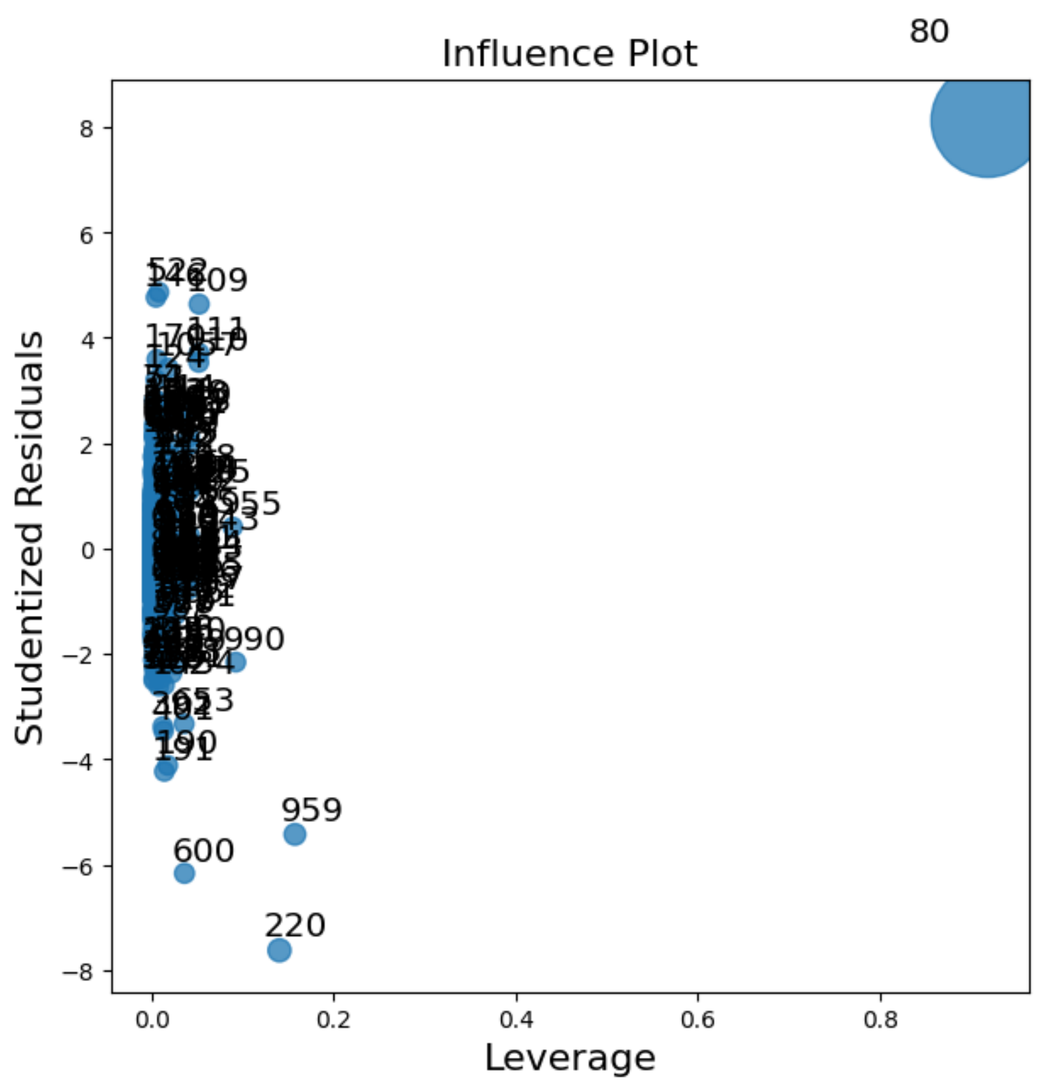
- **Summary from Visualization**

  From the results of the existing visualization, it can be seen that **based on the Gears & Doors category**, the **price range of the existing cars is not too significant**, so the conclusion is that **Gears & Doors** does **not have too much effect on changes in car prices.**

CHECKING OUTLIERS / INFLUENCERS WITH COOK'S DISTANCE & LEVERAGE VALUE

- **Why Using Cook's Distance & Leverage Value?**
  - The main purpose of using **Cook's distance** is to **identify outliers or influential points** that may have a **disproportionate impact on the regression model.**
  - **Leverage values** help in **understanding the influence of individual data points** on the regression analysis, allowing for the **identification of outliers or influential observations** that may need further investigation.

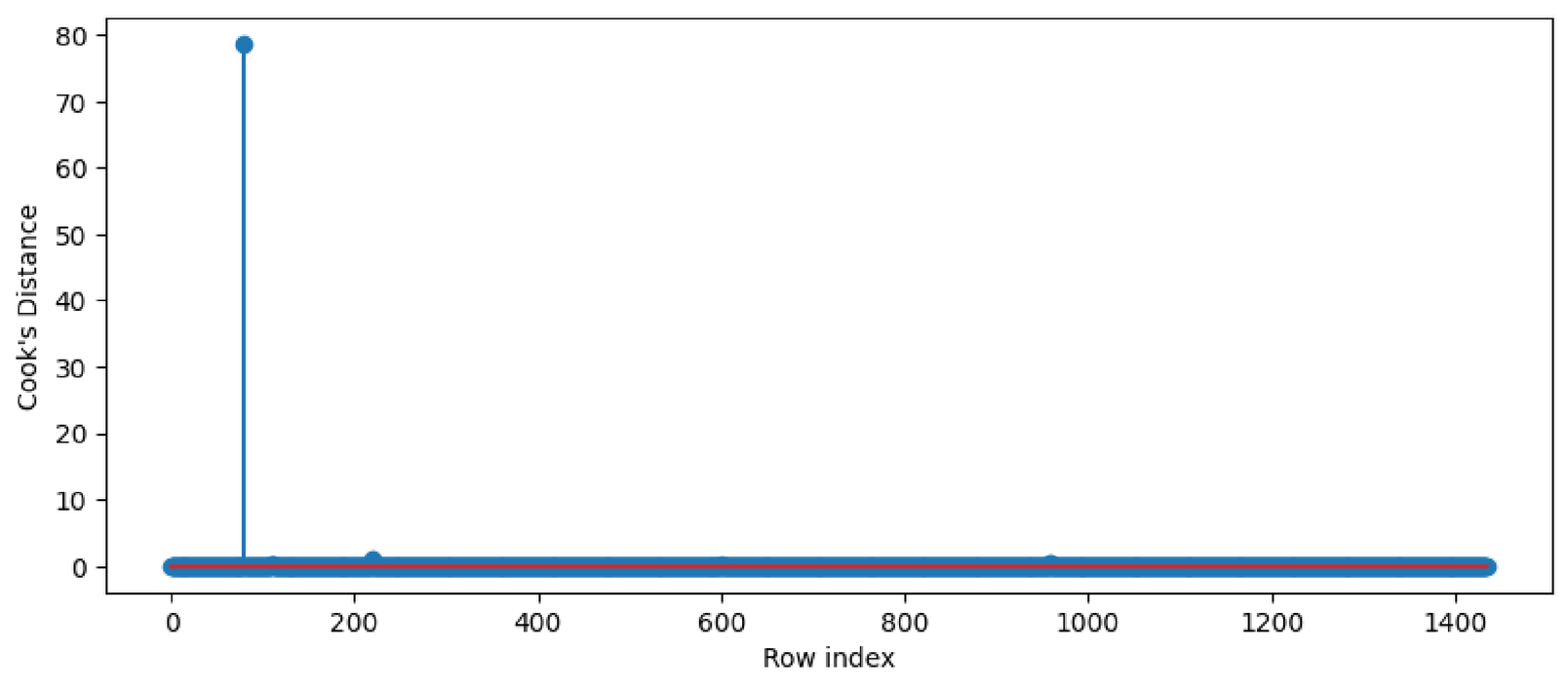
CHECKING OUTLIERS / INFLUENCERS WITH LEVERAGE VALUE



- **Summary from Visualization**

From the results of the existing visualization, it can be seen that **there is one point** that has **high leverage and studentized residuals**, namely the point **number 80 (indicating the index on the dataframe)**. In addition, the **circle** at this point also has the largest diameter among the others, **indicating that this point has a strong ability** to **influence the regression line** and **may represent outliers or data points** that have a **disproportionate impact on the model results.**

CHECKING OUTLIERS / INFLUENCERS WITH COOK'S DISTANCE USING STEM PLOT



- **Summary from Visualization**

  From the results of the existing visualization it can be seen that there is **one point that has a Cook's Distance** which is **quite far from the average existing data**, namely the **point with number 80 (indicating the index in the dataframe)**. The **position of the data point on the x axis shows the Cook's Distance value.**

DATA PRE-PROCESSING

REMOVING INFLUENCER VALUE FROM DATASET

- **Make New Datafame without Influencer Value**

Index from dataframe that will be removed **(Influencer Value) :**

	Price	Age	KM	HP	CC	Doors	Gears	QT	Weight
80	18950	25	20019	110	16000	5	5	100	1180

Comparison shape (rows, columns) previous dataframe with new dataframe :

```
(1435, 9)
(1434, 9)
```

REMOVING INFLUENCER VALUE FROM DATASET

- **Do iterations when modeling with OLS (Ordinary Least Squares) Regression by applying Leverage Cutoff for Improve Accuracy Model**

Leverage Cutoff Formula :

```
leverage_cutoff = (3*(k+1))/n
```

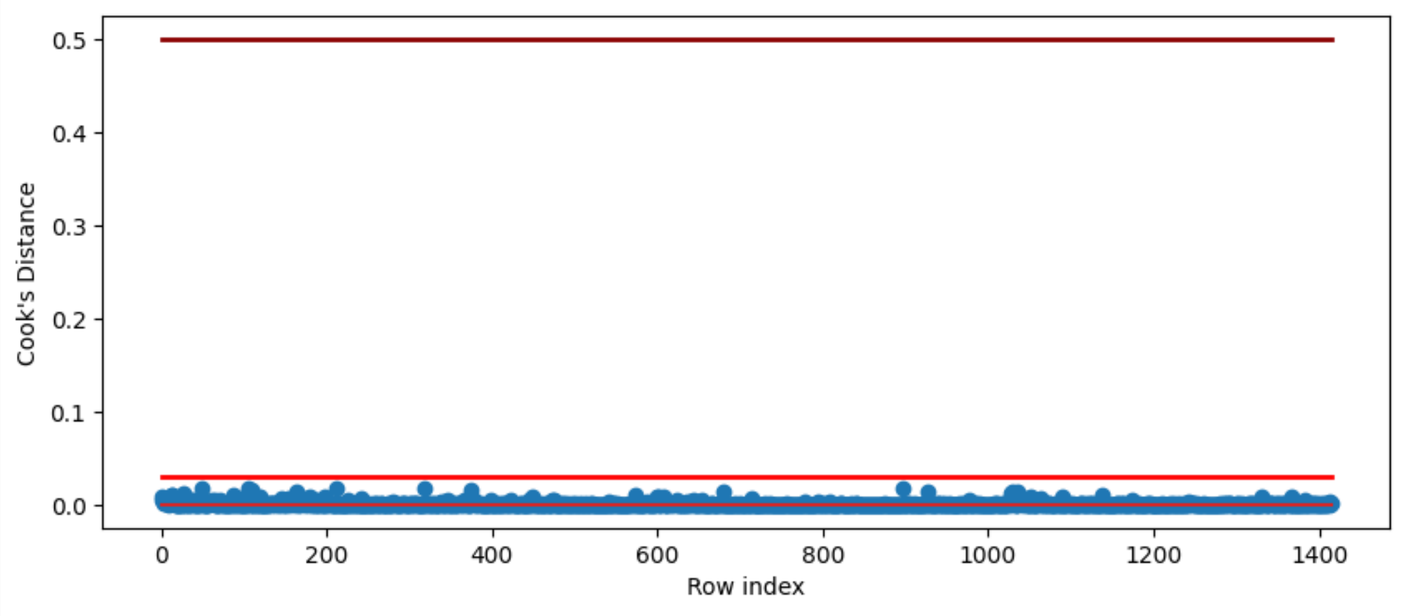
k = Number of variables
n = Total dataset from dataframe

Reason using Leverage Cutoff for iteration :
Limit iterations by looking at the thresholds used to identify influential data points or observations in a regression analysis.

Final Result :

```
Final improved Model accuracy :  0.8960864004304145
```

- # REMOVING INFLUENCER VALUE FROM DATASET



After the iteration process, there was **a lot of data changes in the dataframe which left 1415 (from 1434)**, because in the **previous iteration**, a process was carried out to **drop the influencer value.**

In this stem plot it can also be seen that **there are no more influencers** in the data compared to the previous figure.

- # TRANSFORMATION DATA

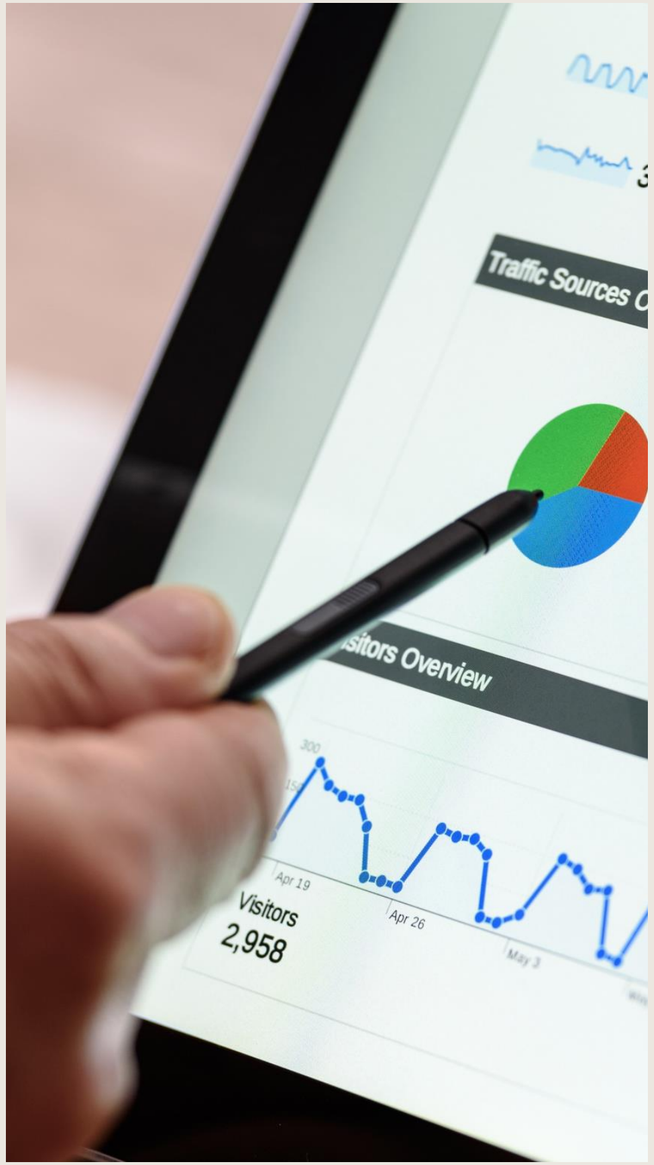
Using **StandardScaler (Standardization) Method** to transform the data for improve performance model.

Before Transformation

	Price	Age	KM	HP	CC	Doors	Gears	QT	Weight
0	13500	23	46986	90	2000	3	5	210	1165
1	13750	23	72937	90	2000	3	5	210	1165
2	13950	24	41711	90	2000	3	5	210	1165
3	14950	26	48000	90	2000	3	5	210	1165
4	13750	30	38500	90	2000	3	5	210	1170

After Transformation

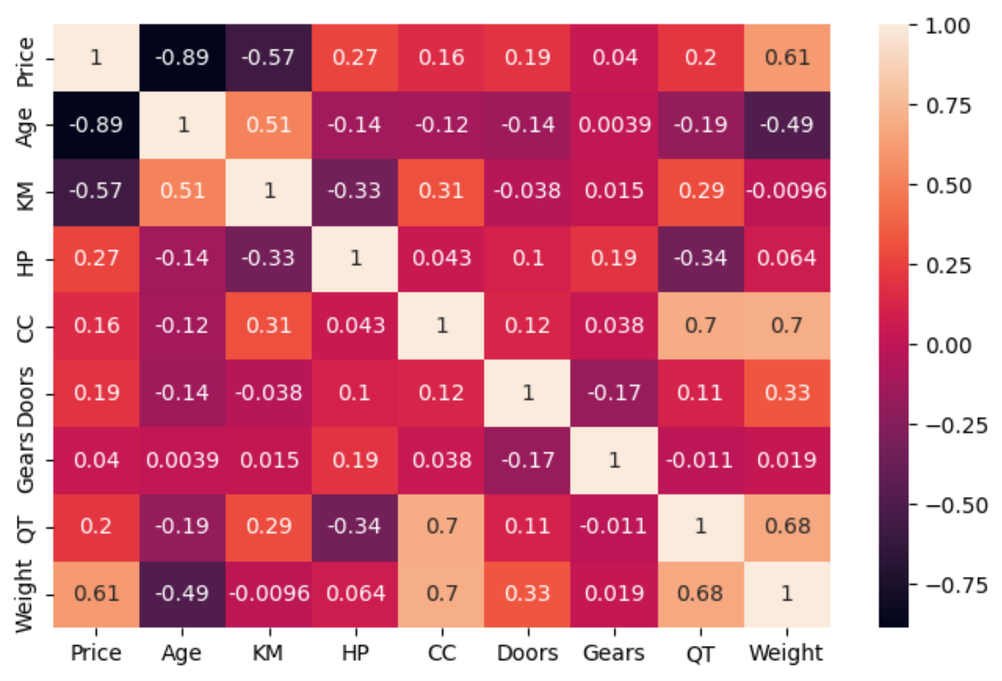
	Price	Age	KM	HP	CC	Doors	Gears	QT	Weight
0	0.812024	-1.792982	-0.581019	-0.800676	2.351384	-1.081458	-0.150625	3.032691	2.013025
1	0.883978	-1.792982	0.118411	-0.800676	2.351384	-1.081458	-0.150625	3.032691	2.013025
2	0.941541	-1.738892	-0.723191	-0.800676	2.351384	-1.081458	-0.150625	3.032691	2.013025
3	1.229357	-1.630713	-0.553690	-0.800676	2.351384	-1.081458	-0.150625	3.032691	2.013025
4	0.883978	-1.414355	-0.809733	-0.800676	2.351384	-1.081458	-0.150625	3.032691	2.119377

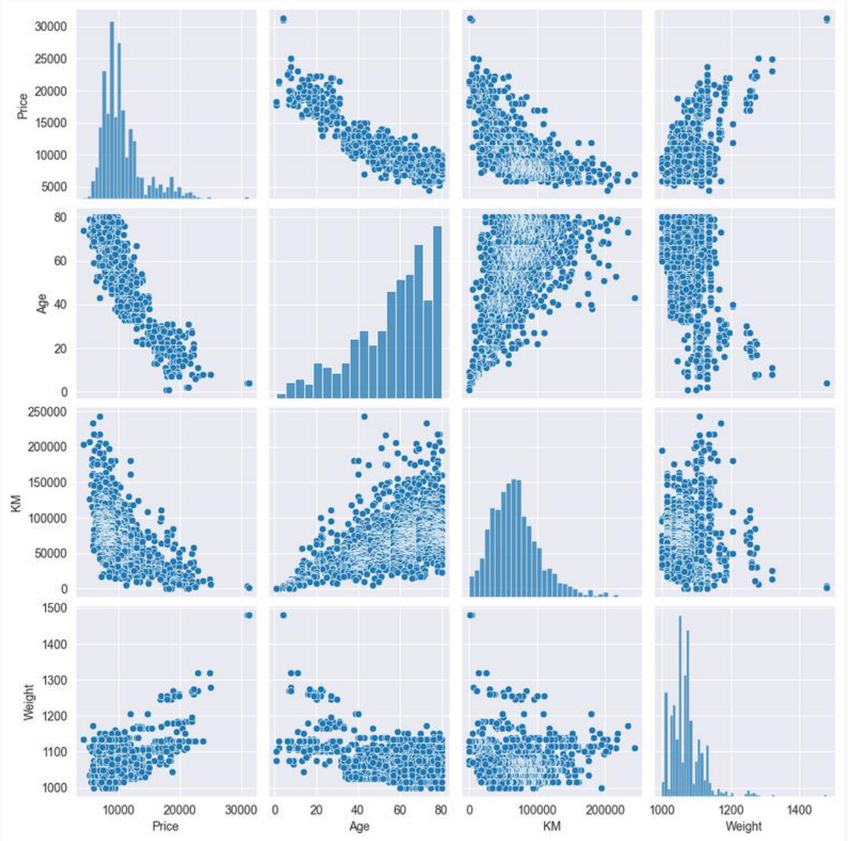


EXPLORATORY DATA ANALYSIS 2

SEARCH CORRELATION BETWEEN VARIABLES

Heatmap





Pairplot

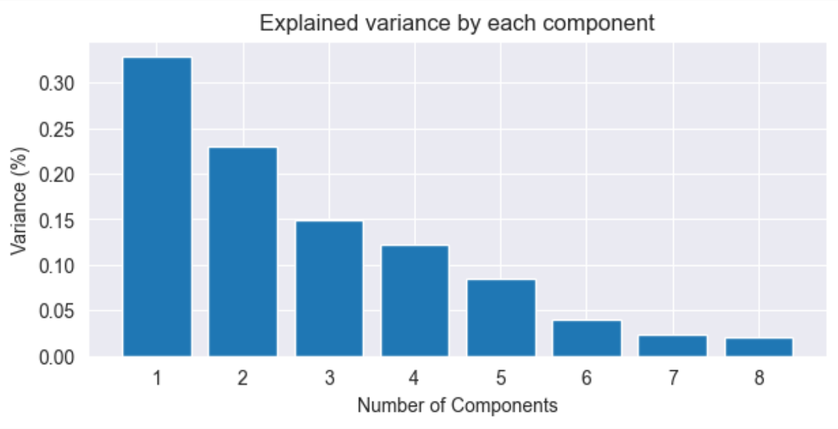
- **Summary from Visualization**
  - **Age & KM** have the **highest correlation with Price**, **but in negative Correlation**
  - **Weight** have the **highest correlation with Price in Positive Correlation**
  - **Weight & QT, QT & CC, Weight & CC, and KM & Age** have good correlation

TRAINING MODEL

PERFORM PCA ON DATASET

- **See the variance ratio in dataframe using PCA**

PCA is used to **transform a high-dimensional dataset** into **a lower-dimensional space** while retaining the **most important patterns or variations** in the data & can be employed to **address multicollinearity issues**, where the predictor variables are highly correlated with each other.




```
array([32.89370451, 22.9851236 , 14.92104303, 12.26370331,  8.42129667,
        4.05649305,  2.37941708,  2.07921875])
```

In the PCA process, we must look at the **variance ratio of each existing data column.** Variance ratio can **guide decisions about how many principal components to retain**, since components with **higher variance ratios** are usually **prioritized for inclusion in a reduced-dimensional data representation.**

In the existing chart, **Age (1), KM (2), & HP (3) columns** have a **large variance ratio** compared to the others. So this column will be prioritized for inclusion in a reduced-dimensional data representation.

CONVERT PCA RESULTS INTO DATAFRAME

Next, do the **PCA result conversion process into a dataframe** because **this dataframe** will be **used in the model training process using linear regression.**

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	Price
0	4.188870	0.241424	-0.340592	-2.298491	0.821049	-0.498966	0.073663	-0.481326	13500
1	4.276369	-0.192878	-0.412960	-2.107000	0.809163	-0.009463	0.178998	-0.468204	13750
2	4.157875	0.300939	-0.331409	-2.315291	0.829087	-0.632062	0.055468	-0.468727	13950
3	4.152659	0.138160	-0.360001	-2.224636	0.837452	-0.580628	0.087427	-0.435015	14950
4	4.127483	0.201474	-0.358977	-2.205313	0.865429	-0.893211	0.072980	-0.293721	13750

The following is the **result of the PCA process that has been included in the dataframe.** There is an additional column, **namely "Price" from the results of the Transformation Data** which is **changed back** to its **normal form.**

SET & SCALLING TRAIN AND TEST DATASET

Feature Variable : All columns, except the Price column
Target Variable : Only the Price column

Shape from Train & Test Data

```
Shape of X_train :  (990, 8)
Shape of X_test  :  (425, 8)
Shape of y_train :  (990,)
Shape of y_test  :  (425,)
```

After setting the training and test data, the next step is to **perform scaling/transformation** of **these two data using StandardScaler**

DO THE MODELLING USE LINEAR REGRESSION

This modeling process is called **"Multiple Linear Regression"**, because it models the **relationship between the dependent variable and two or more independent variables.**

```
Cross Validation Score   :  0.886584406164688
R2 Score (Train)         :  0.892373719425433
R2 Score (Test)          :  0.902872080035517 8
RMSE (Test)              :  1112.2249773038372
```

After do the modelling, next is see the **evaluation metrics** for see the performance of the model.

- **Cross Validation Score** (**Considered good**, because it shows that the model has a relatively high predictive ability)
- **R2 Score** (**Considered good**, because this evaluation suggests that the model is able to capture a significant portion of the underlying patterns and relationships in the data.)
- **RMSE** (**Considered good**, because the **target variable (Price) has a range ranging from 4400 to 31275**, so the **RMSE value tends to be low (the lower the better)**. Besides that, the **smaller RMSE value indicates** that the **model's predictions are relatively close to the actual values on average.**)

COMPARED PREDICTED PRICE WITH ACTUAL PRICE

SHOW COMPARISON

After make new dataframe with containing predicted price and actual price, there are the results from this :

	Predict_Price	Price
275	11314.084604	9950
320	10019.545929	11950
100	5620.496755	19450
27	9306.773372	17950
221	10783.086338	12900
348	9965.396554	12950
169	18515.363641	19950
170	8006.752155	21950
49	10895.899566	20500
326	13091.733228	12750
55	9652.825588	18950
164	7565.151327	19500
396	9975.135778	7900
228	11349.730294	10950
267	12317.866760	14990
11	10194.576401	19950
370	11044.232586	8900
138	7387.224832	16450
210	7853.760169	11950
317	7486.975130	11695

In 20 samples taken randomly to see the results of price predictions on the actual price, it can be seen that there are some predictions that are not too far from the actual price (around 10), but there are also predictions that are too far from the actual price.

CONCLUSION

Model can still be improved

Model can still be improved to produce better predictions by doing more pre-processing of the data or using more advanced techniques when modelling **(Polynomial Regression, Decision Trees, & Random Forests)**

Evaluation Metrics is Good

From several evaluation metrics used **(RMSE, R2 Score, Cross Validation Score)**, all showed good results

Possible that quite a lot of predictions are missed.

I conclude this because based on the results of 20 samples, there are quite a lot of predictions are missed.

END

THANK YOU

Hardianto Tandi Seno

hardiantotandiseno@gmail.com

HTS