CSH3L3-MACHINE LEARNING
FINAL PROJECT

BY

NI PUTU WINDA ARDIYANTI

1301174460

IF-41-INT

TELKOM UNIVERSITY

SCHOOL OF COMPUTING

2020

I. INTRODUCTION

Machine learning is the study that gives the computer the ability to learn based on the experience. Machine Learning is divided into Supervised Learning, Unsupervised Learning, and Reinforcement. Unsupervised Learning is the machine learning model that executes/learns any unlabeled data and makes predictions about it. Supervised learning is the machine learning model that learns any labeled data and makes predictions about it and Reinforcement is an area of machine learning concerned with how software agents ought to take actions in an environment in order to maximize the notion of cumulative reward.

In this report, I will develop some examples of supervised and unsupervised learning models. For supervised learning I will use 2 models of a classifier to execute the data that is given. And for unsupervised learning I will do the clustering of the data given. The aim of this project is to understand how machine learning works, how to do the features engineering of the data that can make our model optimal and also this project aims to compare the model that will be used for classification and also clustering.

II. DATASETS

A. Definition

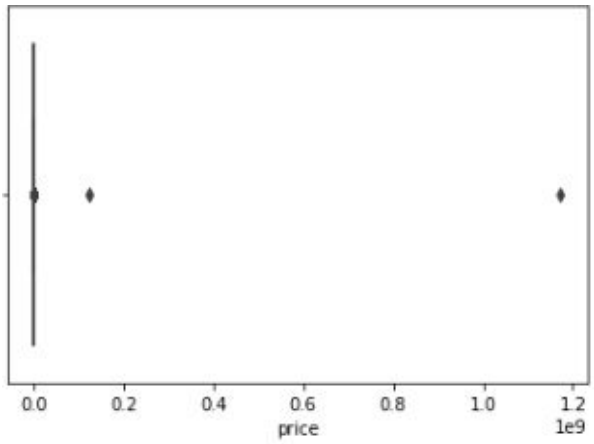
Datasets that will be used in this project is the dataset called 'used_cars'. This dataset has 26 columns and 20001 of data entries. The column that includes in the dataset such as 'id', ''url', 'region', 'region_url', 'price', 'year', 'manufacturer', 'model', 'condition', 'cylinders', 'fuel', 'odometer', 'title_status', 'transmission', 'vin', 'drive', 'size', 'type', 'paint_color', 'image_url', 'description', 'county', 'state', 'lat', and 'long'.

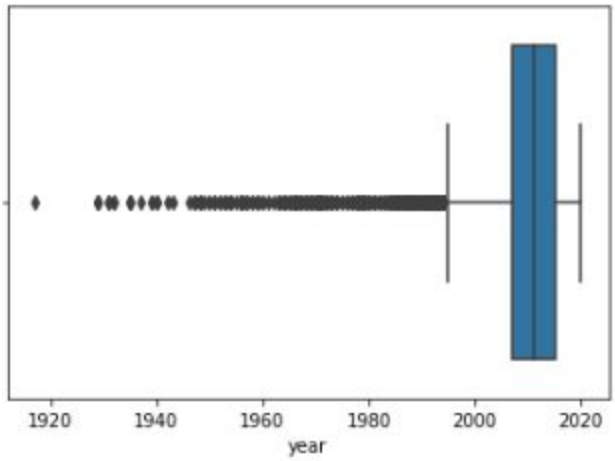
B. Data Exploration

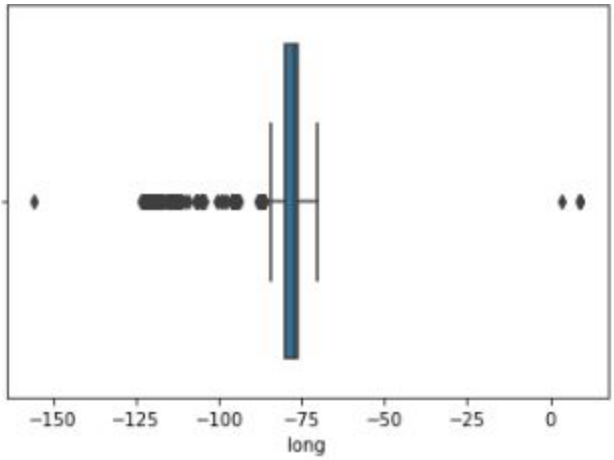
When I do the exploration on this dataset, I found some findings such as:

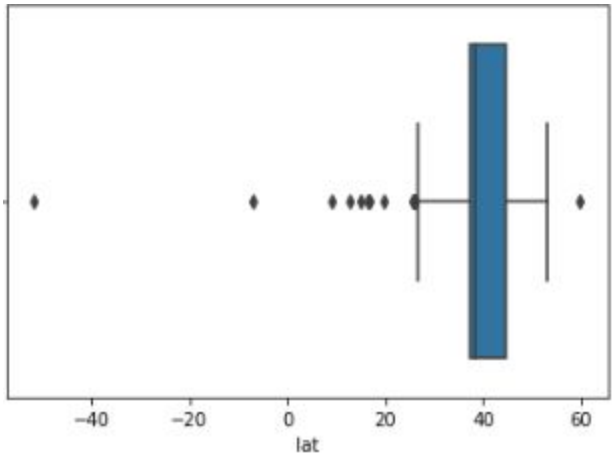
1. Outliers

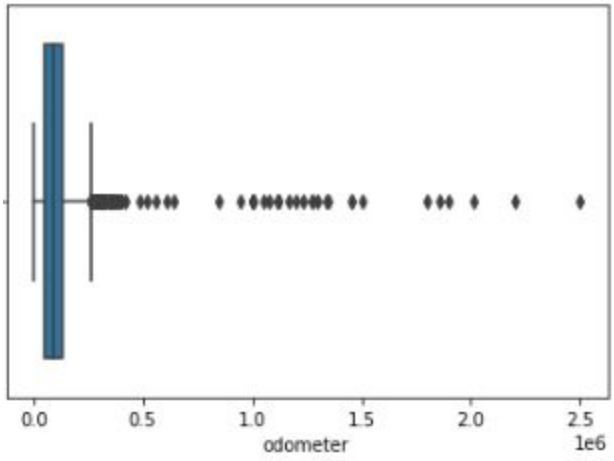
   When I tried to boxplot the features to find the outliers, I found that some of the features contain the outliers. The pictures below show that there are 5 features that have outliers.


0.0
0.2
0.4
0.6
0.8
1.0
1.2
price
1e9


1920
1940
1960
1980
2000
2020
year


−150
−125
−100
−75
−50
−25
0
long


−40
−20
0
20
40
60
lat


0.0
0.5
1.0
1.5
2.0
2.5
odometer
1e6

2. Missing Values

   To find the missing values, I am using the pandas library to count the missing values on the dataset, from what I found there exist 17 features that contain missing values and as we know that there are 20001 data entries in every feature, and feature ‘county’ has the most missing value.

```
vin             6645
drive           4642
size           13115
type            3659
paint_color     5514
image_url          0
description        0
county         20001
state              0
lat             1031
long            1031
dtype: int64
```

```
Unnamed: 0          0
id                  0
url                 0
region              0
region_url          0
price               0
year               12
manufacturer      705
model             265
condition        9152
cylinders        7085
fuel               73
odometer         2389
title_status      110
transmission      190
```

3. Numerical & Categorical Data

   In this project, I divide the data into 2 categories which are numerical category and categorical category. The numerical category features that exist in this dataset are ‘id’,’price’,’year’,’ odometer’,’ lat’,’ long’. The categorical category features that exist in this data is ‘region’, ‘region_url’, ‘manufacturer’, ‘model’, ‘condition’, ‘cylinders’, ‘fuel’, ‘title_status’, ‘transmission’, ‘ state’, ‘descriptions’, ‘image_url’, ‘paint_color’, ‘type’, ‘size’, ‘drive’, and ‘vin’, and also there is one feature which is ‘county’ that cannot be assigned on the category since all the value of the features is null. The goal of categorizing the data is to make it easier to execute the data, for example to do the data imputation in missing value and to encode the data.

4. Data that has a different scale between others

From the observation, when I finish encoding the data because we have to do it for the classification, I found that several of the values in the features have a different range of values. For example, the pictures above show that every feature has a different range value one to others and the difference is quite large. So to avoid the unstable data and help to speed up the calculation between data, we should do feature scaling or normalization.

	id	year	manufacturer	model	fuel	odometer	transmission
0	7034441763	2012.0	38	2197	2	63500.0	1
1	7034440610	2016.0	12	1764	2	10.0	0
2	7034440588	2015.0	13	3285	2	7554.0	0
3	7034440546	2016.0	12	1764	2	10.0	0
4	7034406932	2018.0	12	1843	0	70150.0	0

III. METHODOLOGY

A.Data Preprocessing

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model, and it's also important for us because data preparation can help to increase our accuracy and efficiency of our model.

In this project, I do several data preprocessing which is

1. Finding missing value in the data and do the data imputation

   Missing values are a common occurrence that happens in the data, missing values perhaps caused by the not availability of the data(data is missing) or the event that can give the value for the data does not happen. In this phase, I do the data imputation for the missing value that occurs in the features by adding the value (mean, modus) to the missing value. I am using this method for all

the data preprocessing in classification and clustering. The steps to do this data preprocessing is

a. I sum the missing value that occurs in the features by using this code :

```
#we sum the missing value that occurs
df.isnull().sum()
```

b. After the data of the missing value is shown, I categorize the data that has missing value into two types of, which is categorical and numerical using this code:

```
numerical = ['id','year','odometer']
categorical = ['manufacturer','model',
               'fuel','transmission']
```

c. And the last step is I add the value of mean or modus to the missing value in every feature by using this code :

```
for num in numerical:
    df[num] = df[num].fillna(df[num].mean())

for cat in categorical:
    df[cat] = df[cat].fillna(df[cat].mode().values[0])
```

2. Drop the unnecessary columns that unrelate to our model

   In this phase, I am doing the dropping for the unnecessary features that unrelate to the model that I am going to build. To do this phase, I am using this code :

```
df = df.drop(columns=['Unnamed: 0','region_url', 'image_url', 'lat', 'long','drive',
                      'county','url','price','cylinders','title_status','vin','paint_color',
                      'description','state','region','size','condition','type'])
```

3. Feature scaling

   In this phase, I am doing the feature scaling for the clustering process and classification, the feature scaling that I do is using min-max normalization and standardization.

a. Min-Max Normalization is the technique to rescale the value of the features in the range between 0 and 1 or we can adjust the range

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

b. Standardization is a technique to re-scales a feature value so it has distribution with 0 mean value and variance(standard deviation) equals to 1

$$x' = \frac{x - \text{average}(x)}{stddev(x)}$$

4. Encoding categorical data

   In this phase I do the encoding data for all the classification and clustering models. Encoding is the important phase of data preprocessing because we need to transform the categorical data into numerical data since our model either classification or clustering needs the numerical form of data to execute their model. The categorical encoding that I am using in this project is **Label Encoding**. Label Encoding is the kind of data encoding where we change the label of data become the numerical value that is readable and can be understood by people, for example, we have the data of height that has a value height = {tall, medium, small} and then we want to encode it to the numeric form using label encoder, the value will be height = {0, 1, 2 }

5. Splitting dataset into test and training dataset

   In this phase, we need to split the dataset into 2 forms which are test and training datasets for our classification process. The picture above are the data that I am going to use as the training and testing dataset for the classification.

	id	year	manufacturer	model	fuel	odometer	transmission
0	7034441763	2012.0	38	2197	2	63500.0	1
1	7034440610	2016.0	12	1764	2	10.0	0
2	7034440588	2015.0	13	3285	2	7554.0	0
3	7034440546	2016.0	12	1764	2	10.0	0
4	7034406932	2018.0	12	1843	0	70150.0	0

a. Testing Dataset is a dataset that used to evaluate a model after it is completely trained. Usually, the official performance of the model is reported using this dataset. The testing dataset that is used for the model classification is the '**Transmission**' columns. Transmission column contains the data of a type of the power transmission system in the car. This column contains 3 values which are {automatic, manual, and other} and in numerical form, the value is written as {0, 1, 2}.

b. Training Dataset is the actual dataset used to train the model for performing various actions or making predictions. The training dataset that will be used in this classification contains 6 columns which are 'id', 'year', 'manufacturer', 'model', 'fuel', and 'odometer'.

B. Classification

In this classification, I created 3 different models for the classification. I Am using the same type of data preparation for every model and different classifiers for every model. The explanation of every model will be explained above.

1. **Model A**
    - Classifier: Naive Bayes

      Naive Bayes is one of the commonly-used algorithms for classification, this classifier is a collection of classification algorithms based on

Bayes' Theorem and also this classifier is a kind of probabilistic classifier.

- Datasets: Used Cars
- Data Preprocessing type :
  - Imputation of missing value using mean and modus
  - Categorical encoding using laber encoder (label encoder)
  - Data splitting
- Testing Dataset: Transmission columns
- Training Dataset: 'id', 'year', 'manufacturer', 'model', 'fuel', and 'odometer' columns.
- Result of the model
  - Define the classifier and calculate the predictions

```
#using Gaussian Naive Bayes to predict the class
gnb = GaussianNB()
gnb.fit(X_train, y_train)
y_pred_NB = gnb.predict(X_test)
#return the prediction
y_pred_NB
```

```
array([0, 0, 0, ..., 0, 0, 0])
```

  - Confusion Matrix

		Actual Class		
		0	1	2
Predicted Class	0	5219	101	0
	1	443	62	0
	2	171	5	0

  - Accuracy Score: **0.8800199966672221**
  - Report :

	Precision	Recall	F1-score	support

0	0.89	0.98	0.94	5320
1	0.37	0.12	0.18	505
2	0.00	0.00	0.00	176
Accuracy			**0.88**	6001
Macro avg	0.42	0.37	0.37	6001
Weighted avg	0.82	0.88	0.85	6001

- Result Class Predictions

Class Label	Sum of Result
0 (Automatic)	5833
1 (Manual)	168
2 (Others)	0

2. Model B

- Classifier: Decision Tree

  Decision Tree is one of the predictive modeling approaches used in machine learning for use in terms of decision analysis.

- Datasets: Used Car
- Data Preprocessing type :
  - Imputation of missing value using mean and modus
  - Categorical encoding using laber encoder
  - MinMax Normalization
  - Data splitting
- Testing Dataset: Transmission columns

- Training Dataset: ‘id’, ‘year’, ‘manufacturer’, ‘model’, ‘fuel’, and ‘odometer’ columns.
- Result of the model
  - Define the classifier and calculate the predictions

```
#using the decision tree to predict the class
model_dt = DecisionTreeClassifier(criterion='entropy')
model_dt.fit(X_train,y_train)
y_pred_ID3 = model_dt.predict(X_test)
#return the prediction
y_pred_ID3
```

```
array([0, 0, 0, ..., 0, 0, 0])
```

  - Confusion Matrix

		Actual Class		
		0	1	2
Predicted Class	0	4926	285	109
	1	269	226	10
	2	105	12	59

  - Accuracy Score: **0.8683552741209798**
  - Report :

	Precision	Recall	F1-score	support
0	0,93	0,93	0,93	5320
1	0,43	0,45	0,44	505
2	0,33	0,34	0,33	176
Accuracy			**0,87**	6001
Macro avg	0,56	0,57	0,57	6001
Weighted avg	0,87	0,87	0,87	6001

- Result Class Predictions

Class Label	Sum of Result
0 (Automatic)	5300
1 (Manual)	523
2 (Others)	178

3. Model C

- Classifier: KNN
- Datasets: Used Car
- Data Preprocessing type :
  - Imputation of missing value using mean and modus
  - Categorical encoding using laber encoder
  - MinMax Normalization
  - Data splitting
- Testing Dataset: Transmission columns
- Training Dataset: ‘id’, ‘year’, ‘manufacturer’, ‘model’, ‘fuel’, and ‘odometer’ columns.
- Result of the model
  - Define the classifier and calculate the predictions
  - Set the neighbors as 4

```
#using the knn to predict the class
model_knn = KNeighborsClassifier(n_neighbors=4)
model_knn.fit(X_train, y_train)
y_pred_knn = model_knn.predict(X_test)
#return the prediction
y_pred_knn
```

```
array([0, 0, 0, ..., 0, 0, 0])
```

- Confusion Matrix

		Actual Class		
		0	1	2
Predicted Class	0	5162	133	25
	1	402	101	2
	2	158	9	9

- Accuracy Score : **0.8836860523246126**
- Report :

	Precision	Recall	F1-score	support
0	0,90	0,97	0,93	5320
1	0,42	0,20	0,27	505
2	0,25	0,05	0,08	176
Accuracy			**0,88**	6001
Macro avg	0,52	0,41	0,43	6001
Weighted avg	0,84	0,88	0,85	6001

- Result Class Predictions

Class Label	Sum of Result
0 (Automatic)	5722
1 (Manual)	243
2 (Others)	36

4. Result Analysis of Classification

From the classification that I've done, in this phase, I will deliver the result analysis of the classification.

- Result Class

	Actual Predictions	Model A (Naive Bayes)	Model B (Decision Tree)	Model C (KNN)
0	5320	5833	5311	5722
1	505	168	522	243
2	176	0	168	36

From the result class above we know that all the models have a different sum value of label 0,1,2. And the difference is not really significant from the actual predictions unless the prediction of model A and Model C has a huge gap with the actual predictions. The resulting class of model B (Decision Tree) is almost the same as the Actual predictions.

- Accuracy

Model A (Naive Bayes)	Model B (Decision Tree)	Model C (KNN)
0.8800199966672221	0.8691884685885686	0.8785202466255624

The accuracy betweens models is quite similar. The average accuracy betweens models (Model A to C, Model B to C) only has a 0.01 range of differences or in percent, we can say the accuracy of the model betweens is 1 % differences. But for model A and B the difference is

0.02 or 2%. I also can conclude that the biggest accuracy of the model is in the model A (Naive Bayes) even though when we look at the result classes, the result of model B is almost similar with the result of true predictions

C.Clustering

In this clustering process, I created 2 different models for the clustering. I am using a different type of data preparation for every model and also using a different feature for the model. I am using 2 different features for every clustering, which means the clustering visualization will be formed as 2D (x, y). The algorithm that is used in this clustering is the K-Means algorithm. The explanation of every model will be explained above.

1. What is K-Means Algorithm?

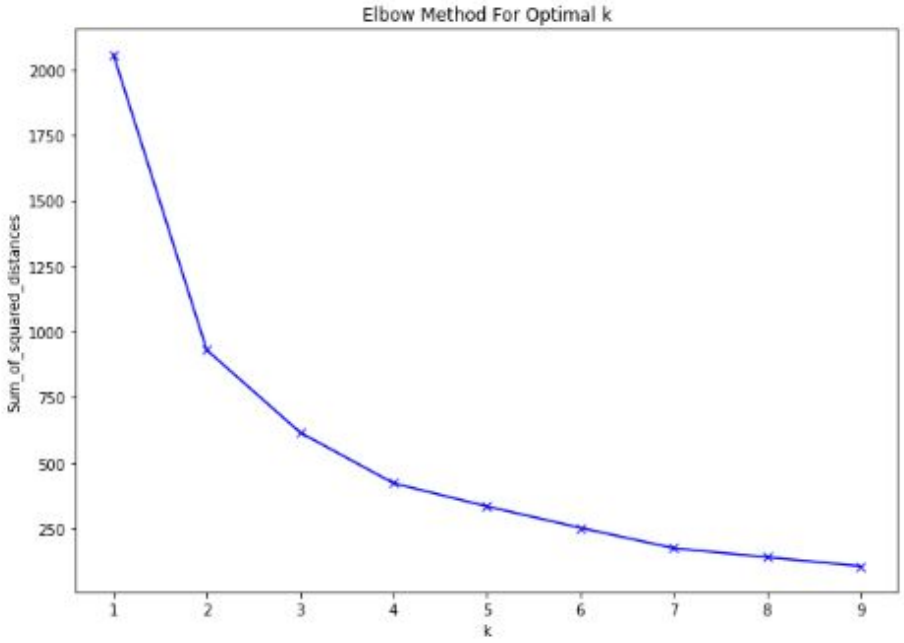
K-Means algorithm is an iterative algorithm that tries to partition the dataset into K clusters where each data point belongs to only one group. This algorithm will try to make every data point in one cluster is as similar as possible and every data point in other clusters will be as different as possible.

2. Clustering Model A

- Clustering Algorithm: K-Means Algorithm
- Datasets: Used cars
- Data Preprocessing Steps :
    - Drop the unnecessary columns since the model only uses 2 columns which is ‘fuel’ and ‘model’ columns
    - Data Imputation, in this phase the model does the data imputation by filling the missing value of columns ‘fuel’ and ‘model’ with the mode/modus.

- Categorical Encode, in this phase the model does the categorical encode to column ‘fuel’ and ‘model’. The categorical encode that is used in this phase is label encode.
- Data Normalization, in this phase the model does the data normalization using Min-Max normalization

- Cluster’s Label (label x, label y)
  - Label x: model
  - Label y: fuel
- Clustering Process
  - Visualize the elbow method result to define the optimal K for clustering



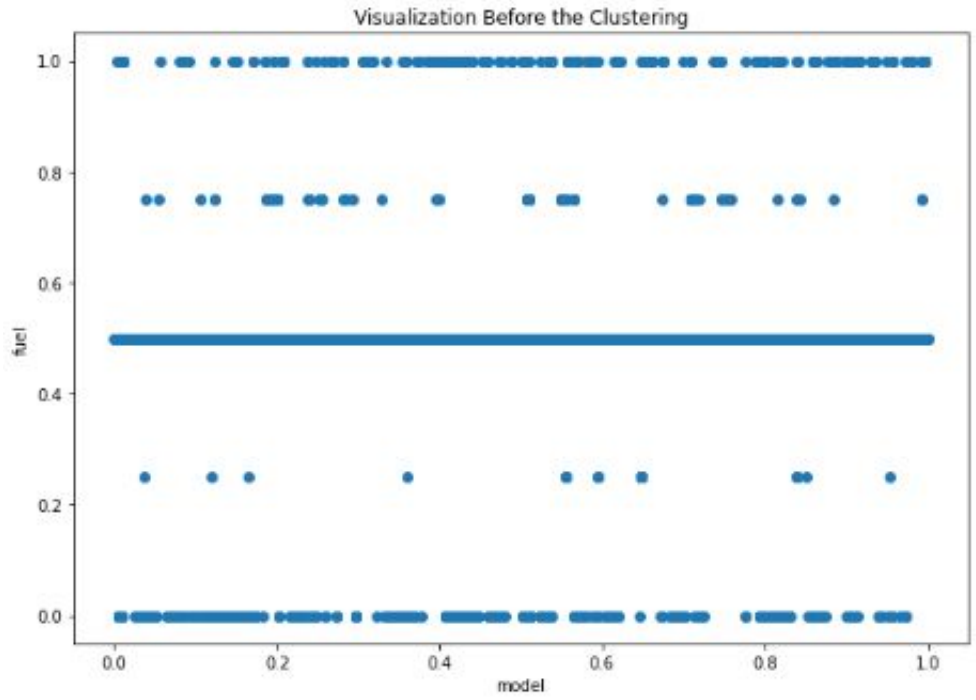
- Assign the K value, the K value that we want to assign is optional. For example, in this model, I assign the **K value is 7,** according to the elbow’s result.
- After we assign the K value, we initialize our first centroid randomly.

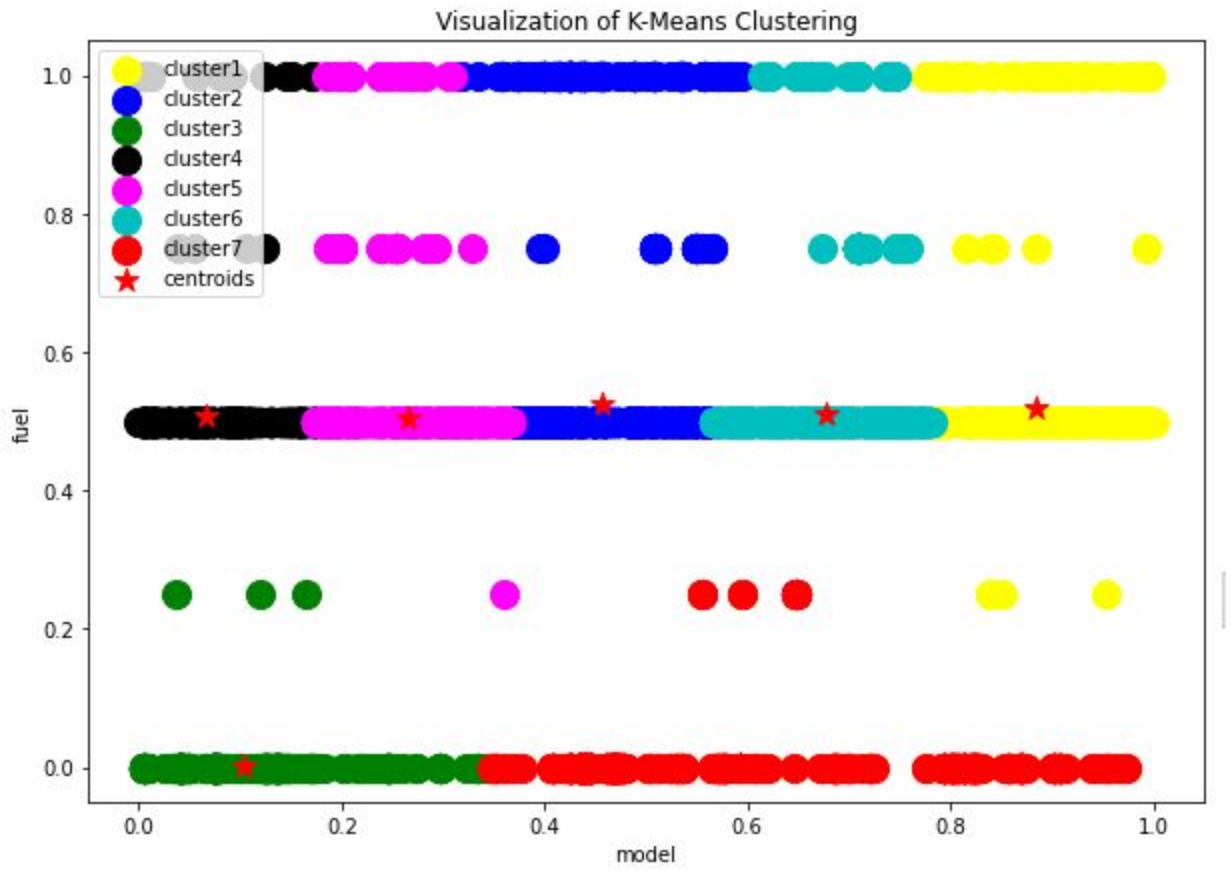
- Calculate the Euclidean distance from each point to the centroids, euclidean distance is the distance between 2 points, the formula to calculate it is :

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

- Re-group the data points based on the clusters and the distance that we've been initialized and store it to the dictionary that we've been prepared
- We repeat the step where we calculate the euclidean and regroup the data points until its convergence(there's no data point that moving), in this method I initialized that the repeat step (iteration) is
- After it convergence, we visualize the clustering result

- Visualization





3. Result Analysis of Clustering Model A

Clusters Result

a. Cluster 1

As we can see from the picture below, cluster 1 (represented with yellow color) has 4465 data members. Column '0' means that the data points on label x (model0 and column '1' means that the data points on label y (fuel). From what I observe the there are a lots of data member of cluster 1 in label y = 0.5 and y = 1.0

	0	1
0	0.792904	0.5
1	0.792662	0.5
2	0.992035	1.0
3	0.931451	0.5
4	0.792662	0.5
...	...	...
4460	0.883659	0.5
4461	0.960898	0.5
4462	0.996138	0.5
4463	0.967656	0.5
4464	0.782525	0.5

4465 rows × 2 columns

b. Cluster 2

As we can see from the picture below, cluster 2 (represented with blue color) has 4561 data members. Column '0' means that the data points on label x (model0 and column '1' means that the data points on label y (fuel). From what i observe the there are a lots of data member of cluster 2 in label y = 0.5 and y = 1.0

	0	1
0	0.530292	0.5
1	0.425778	0.5
2	0.425778	0.5
3	0.425778	0.5
4	0.433985	0.5
...	...	...
4556	0.538016	0.5
4557	0.416848	0.5
4558	0.389573	0.5
4559	0.451364	0.5
4560	0.536809	0.5

4561 rows × 2 columns

c. Cluster 3

As we can see from the picture below, cluster 3 (represented with green color) has 552 data members. Column ‘0’ means that the data points on label x (model0 and column ‘1’ means that the data points on label y (fuel). From what i observe the there are a lots of data member of cluster 3 in label y = 0.0

	0	1
0	0.040068	0.0
1	0.077239	0.0
2	0.110789	0.0
3	0.074584	0.0
4	0.216751	0.0
...	...	...
547	0.043688	0.0
548	0.125513	0.0
549	0.047309	0.0
550	0.040068	0.0
551	0.332851	0.0

552 rows × 2 columns

d. Cluster 4

As we can see from the picture below, cluster 4 (represented with black color) has 1855 data members. Column ‘0’ means that the data points on label x (model0 and column ‘1’ means that the data points on label y (fuel). From what i observe the there are a lots of data member of cluster 4 in label y = 0.5

	0	1
0	0.004103	0.5
1	0.004103	0.5
2	0.051895	0.5
3	0.172098	0.5
4	0.050447	0.5
...	...	...
1850	0.010138	0.5
1851	0.041516	0.5
1852	0.029930	0.5
1853	0.070239	0.5
1854	0.089790	0.5

1855 rows × 2 columns

e. Cluster 5

As we can see from the picture below, cluster 5 (represented with purple color) has 3674 data members. Column '0' means that the data points on label x (model0 and column '1' means that the data points on label y (fuel). From what i observe the there are a lots of data member of cluster 5 in label y = 0.5 and y = 1.0

	0	1
0	0.280714	0.5
1	0.280714	0.5
2	0.291576	0.5
3	0.203476	0.5
4	0.293748	0.5
...	...	...
3669	0.307989	0.5
3670	0.321989	0.5
3671	0.307989	0.5
3672	0.306541	0.5
3673	0.307989	0.5

3674 rows × 2 columns

f. Cluster 6

As we can see from the picture below, cluster 6 (represented with cyan color) has 3763 data members. Column ‘0’ means that the data points on label x (model0 and column ‘1’ means that the data points on label y (fuel). From what i observe the there are a lots of data member of cluster 6 in label y = 0.5 and y = 1.0

	0	1
0	0.759594	0.5
1	0.672701	0.5
2	0.672460	0.5
3	0.753077	0.5
4	0.674873	0.5
...	...	...
3758	0.606565	0.5
3759	0.610427	0.5
3760	0.685976	0.5
3761	0.750422	0.5
3762	0.651219	0.5

3763 rows × 2 columns

g. Cluster 7

As we can see from the picture below, cluster 7 (represented with red color) has 1131 data members. Column ‘0’ means that the data points on label x (model0 and column ‘1’ means that the data points on label y (fuel). From what I observe the there are a lots of data member of cluster 7 in label y = 0.0

	0	1
0	0.444847	0.0
1	0.439054	0.0
2	0.425778	0.0
3	0.425778	0.0
4	0.792662	0.0
...	...	...
1126	0.821627	0.0
1127	0.807386	0.0
1128	0.808593	0.0
1129	0.472604	0.0
1130	0.829592	0.0

1131 rows × 2 columns

Centroids

There are 7 centroids that exist in this clustering, and every centroids appear in every clusters

	0	1	2	3	4	5	6
0	0.885244	0.457366	0.103518	0.065714	0.264734	0.676353	0.592405
1	0.518309	0.525652	0.001359	0.507143	0.505444	0.510696	0.004421

The columns from 0-6 represent the centroids index from centroid 0 to centroid 6. The row ‘0’ and ‘1’ represents the label, where ‘0’ is coordinate/value the data points on label x (model)and ‘1’ is the coordinate/value of the data points on label y (fuel)

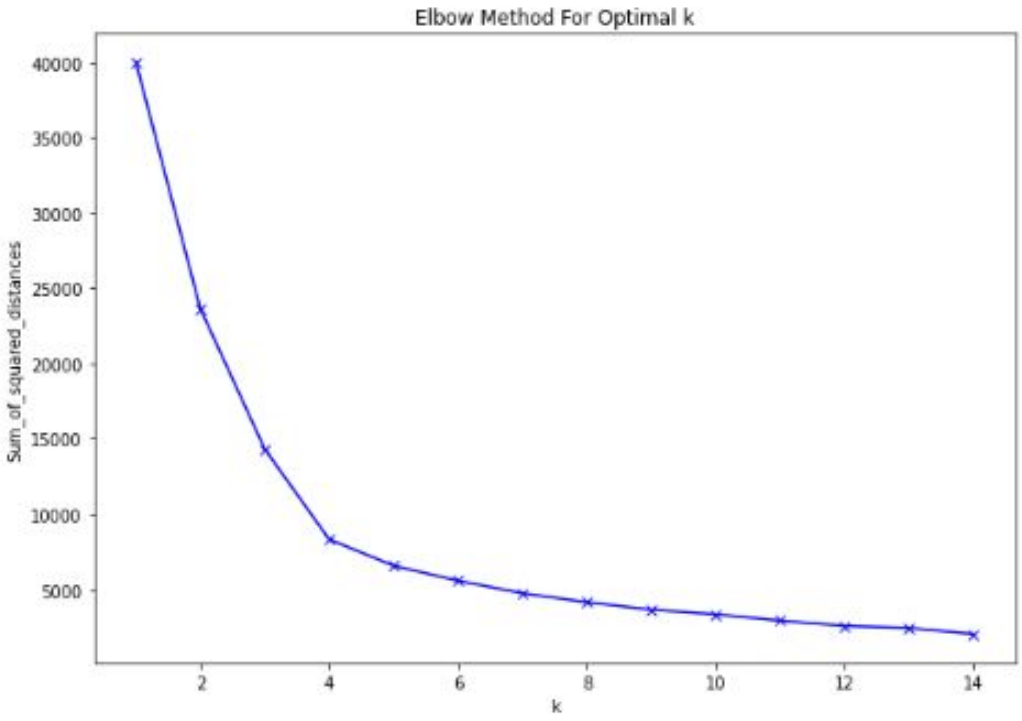
Result of the centroid can be read as:

1. The first centroid (centroid 0 at cluster 1) at label x = 0.885244 and y = 0.518309
2. The second Centroid (centroid 1 at cluster 2) at label x = 0.457366 and y = 0.525652
3. The third centroid (centroid 2 at cluster 3) at label x = 0.103518 and y = 0.001359
4. The fourth centroid (centroid 3 at cluster 4) at label x = 0.065714 and y = 0.507143
5. The fourth centroid (centroid 4 at cluster 5) at label x = 0.264734 and y = 0.505444

6. The fourth centroid (centroid 5 at cluster 6) at label x = 0.676353 and y = 0.510696
7. The fourth centroid (centroid 6 at cluster 7) at label x = 0.592405 and y = 0.004421

4. Clustering Model B

- Clustering Algorithm: K-Means Algorithm
- Datasets: Used cars
- Data Preprocessing Steps :
    - Drop the unnecessary columns since the model only uses 2 columns which is ‘manufacturer’ and ‘model’ columns
    - Data Imputation, in this phase the model does the data imputation by filling the missing value of columns ‘manufacturer’ and ‘model’ with the mode/modus.
    - Categorical Encode, in this phase the model does the categorical encode to column ‘manufacturer’ and ‘model’. The categorical encode that is used in this phase is label encode.
    - Data Normalization, in this phase the model does the data normalization using Standard normalization
- Cluster’s Label (label x, label y)
    - Label x: manufacturer
    - Label y: model
- Clustering Process
    - Visualize the elbow method result to define the optimal K for clustering



- Assign the K value, the K value that we want to assign is optional. For example, in this model, I assign the **K value is 4,** according to the elbow's result.
- After we assign the K value, we initialize our first centroid randomly.
- Calculate the Euclidean distance from each point to the centroids, euclidean distance is the distance between 2 points, the formula to calculate it is :

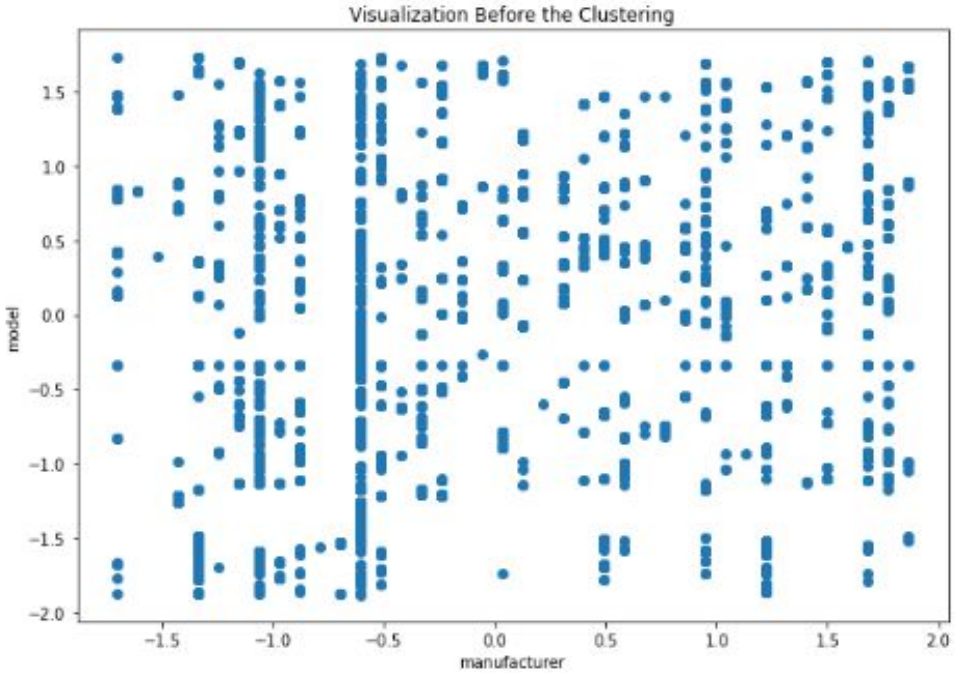
$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

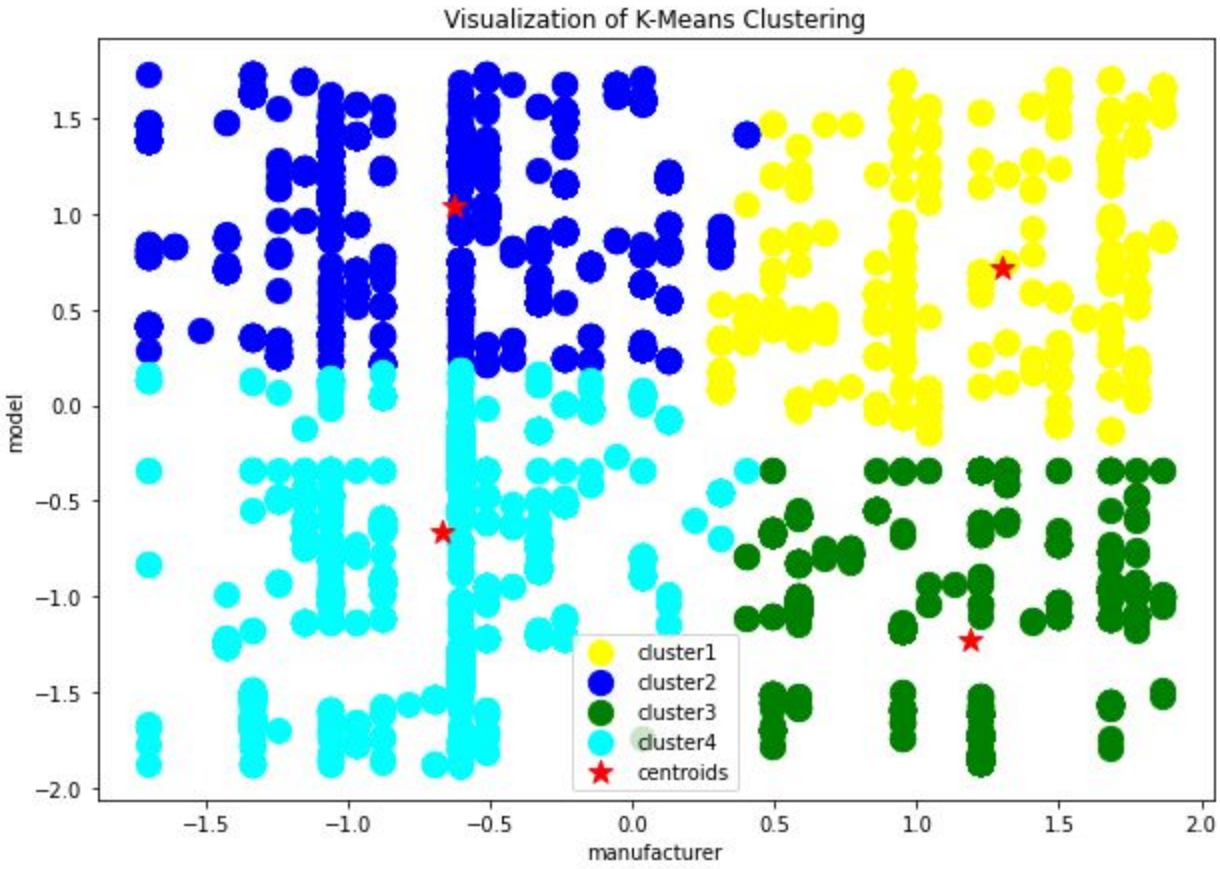
- Re-group the data points based on the clusters and the distance that we've been initialized and store it to the dictionary that we've been prepared
- We repeat the step where we calculate the euclidean and regroup the data points until its convergence(there's no data

point that moving), in this method I initialized that the repeat step (iteration) is

- After it convergence, we visualize the clustering result

- Visualization





5. Result Analysis of Clustering Model B

Cluster Result

a. Cluster 1

As we can see from the picture below, cluster 1 (represented with yellow color) has 4187 data members. Column ‘0’ means that the data points on label x (manufacturer) and column ‘1’ means that the data points on label y (model). From what i observe the data distribution of clustering 1 is on label x between coordinate/value 0.5-2.0 and on label y is between coordinate/value -0.2-more than 1.5

	0	1
0	1.772412	0.036298
1	1.680947	1.485904
2	0.583373	0.864893
3	1.498018	0.558749
4	1.680947	1.485904
...	...	...
4182	1.680947	0.102586
4183	1.680947	1.313207
4184	0.949231	0.473273
4185	1.498018	1.616735
4186	0.949231	0.947753

4187 rows × 2 columns

b. Cluster 2

As we can see from the picture below, cluster 2 (represented with blue color) has 5273 data members. Column '0' means that the data points on label x (manufacturer) and column '1' means that the data points on label y (model). From what i observe the data distribution of clustering 2 is on label x between coordinate/value >-1.5-0.5 and on label y is between coordinate/value 0.3-more than 1.5

	0	1
0	-0.514202	0.985258
1	-0.514202	0.984386
2	-0.514202	1.704828
3	-0.514202	0.984386
4	-0.514202	0.984386
...	...	...
5268	0.034586	1.606268
5269	-1.611776	0.831750
5270	-1.062989	1.117833
5271	0.034586	1.592313
5272	-0.514202	1.719655

5273 rows × 2 columns

c. Cluster 3

As we can see from the picture below, cluster 3 (represented with green color) has 2640 data members. Column ‘0’ means that the data points on label x (manufacturer) and column ‘1’ means that the data points on label y (model). From what I observe the data distribution of clustering 3 is on label x between coordinate/value >0.4 until less than 2.0 and on label y is between coordinate/value -1.9 until more than -0.7

	0	1
0	1.223624	-0.341367
1	1.223624	-0.341367
2	1.223624	-0.341367
3	1.223624	-1.735152
4	1.223624	-1.600832
...	...	...
2635	1.680947	-0.767004
2636	0.766302	-0.772237
2637	1.223624	-0.677166
2638	1.680947	-0.767004
2639	1.680947	-1.555478

2640 rows × 2 columns

d. Cluster 4

As we can see from the picture below, cluster 4 (represented with cyan (light blue) color) has 7901 data members. Column ‘0’ means that the data points on label x (manufacturer) and column ‘1’ means that the data points on label y (model). From what I observe the data distribution of clustering 4 is on label x between coordinate/value less than -1.5 until 0.5 and on label y is between coordinate/value -1.9 until 0.3

	0	1
0	-0.605666	-0.341367
1	-0.605666	-0.341367
2	-0.605666	-0.272463
3	-0.605666	-0.293396
4	-0.605666	-0.311712
...	...	...
7896	0.034586	0.064208
7897	-0.605666	-0.373639
7898	-1.062989	-0.472198
7899	-0.605666	-0.248913
7900	0.034586	0.059847

7901 rows × 2 columns

Centroids

There are 4 centroids that exist in this clustering, and every centroids appear in every clusters

	0	1	2	3
0	1.296870	-0.624799	1.190295	-0.667993
1	0.717348	1.039226	-1.226442	-0.663912

The columns from 0-6 represent the centroids index from centroid 0 to centroid 6. The row ‘0’ and ‘1’ represents the label, where ‘0’ is coordinate/value the data points on label x (manufacturer)and ‘1’ is the coordinate/value of the data points on label y (model)

Result of the centroid can be read as:

1. The first centroid (centroid 0 at cluster 1) at label x = 1.296870 and y = 0.717348
2. The second Centroid (centroid 1 at cluster 2) at label x = -0.624799 and y =1.039226

3. The third centroid (centroid 2 at cluster 3) at label x = 1.190295 and y = -1.226442
4. The fourth centroid (centroid 3 at cluster 4) at label x = -0.667993 and y = -0.663912

IV. CONCLUSION

1. **Data Preparation**

   In data preparation there are several important things that should be done before we do the classification. The first important thing is we should understand the data that is given to us, we can do such a thing called data exploration, where we define the types of data, the outliers in data and missing values that appear on the data . After we do the data exploration, we should decide what should be done to the data, for example if we found the missing value, what should be done with it, and this step we can call as data preparation. in data preparation we do such things as drop the columns, data imputation, normalization/scaling, data splitting, label encoder

2. **Classification**

   a. From the classification result I can conclude that when I am using 3 models to execute the classification, I found that the accuracy between the models is quite similar and also the gap of the result class of the model is not huge if we compare with the true predictions result.

   b. The biggest accuracy that has been done by the model is at 88% when I am using Naive Bayes model.

   c. For the accuration result, i can conclude that the accuracy result its already goods because its mode than 80%

3. Clustering

a. Model A

- For model A in clustering, when I visualize the elbow method to define the optimal K, the optimal number K is 7.
- There are 7 clusters that appear in the clustering process, since we assign the K is 7
- There are also 7 centroids that appears in every clusters
- For the clustering result, cluster 2 is the biggest cluster since its have a biggest data member which is 4561 data

b. Model B

- For model B in clustering, when I visualize the elbow method to define the optimal K, the optimal number K is 4.
- There are 4 clusters that appear in the clustering process, since we assign the K is 4
- There are also 4 centroids that appears in every clusters
- For the clustering result, cluster 4 is the biggest cluster since its have a biggest data member which is 7901 data