LAPORAN TUGAS BESAR *MACHINE LEARNING*

1301174038

Ekky Yulianti Prastika S.

Program Studi Sarjana Informatika

Fakultas Informatika

Universitas Telkom

Bandung

2020

A. Latar Belakang Masalah

Seiring dengan perkembangan jaman dan semakin meningkatnya kebutuhan alat transportasi membuat masyarakat berbondong-bondong untuk membeli kendaraan, salah satunya yaitu mobil. Mobil sangat dibutuhkan oleh banyak khalayak publik sebagai sarana transportasi sehari–hari yang lebih efisien dan dinamis. Oleh sebab itu, bukan hal asing lagi kalau sekarang banyak terdapat penjual mobil bekas.

Beberapa pertimbangan yang sering dijumpai dalam melihat mobil bekas yaitu mesin, tahun mobil, harga mobil, kondisi mobil, silinder dari mobil, tipe mobil, jenis persneling yang digunakan, bahan bakar, odometer atau kilometer dari mobil, dan daerah dimana mobil itu.

Menurut data yang dikumpulkan, eksistensi mobil bekas tidak ada matinya ditandai dengan masih banyaknya mobil yang tetap digunakan dimana mesinnya dari tahun 1917 sampai dengan 2020.

Setiap mobil mempunyai bermacam-macam indikator yang berbeda untuk mengetahui jenis persneling (*transmission*) yang digunakan, terutama pada mobil bekas. Hal ini digunakan untuk mempermudah montir dalam menindaklanjuti atau mengambil keputusan untuk tindakan lanjutan apabila terjadi kerusakan atau kendala pada mobil. Sedangkan untuk toko *sparepart* adalah membantu menjaga persediaan sparepart untuk jenis persneling (*transmission*) yang paling sering digunakan oleh masyarakat.

B. Masalah

Klasifikasi dilakukan untuk menentukan jenis persneling atau *transmission* pada suatu mobil bekas dengan menggunakan fitur ‘price’, ‘year’, ‘manufacturer’, ‘condition’, ‘cylinders’, ‘fuel’, ‘odometer’, dan ‘type’ untuk menentukan 3 label yaitu 0, 1, dan 2. Sedangkan untuk *clustering*, akan mengelompokkan fitur ‘transmission’ dan ‘condition’.

C. Tujuan

Berikut tujuan dari penulisan yaitu:

1. Untuk membantu mengambil keputusan berdasarkan data yang diberikan.
2. Memprediksi jenis persneling (*transmission*) yang digunakan pada mobil bekas dan mengelompokkan jenis persneling yang paling sering digunakan pada mobil bekas.
3. Mendapatkan informasi tentang data di mobil bekas.

D. Eksplorasi Data

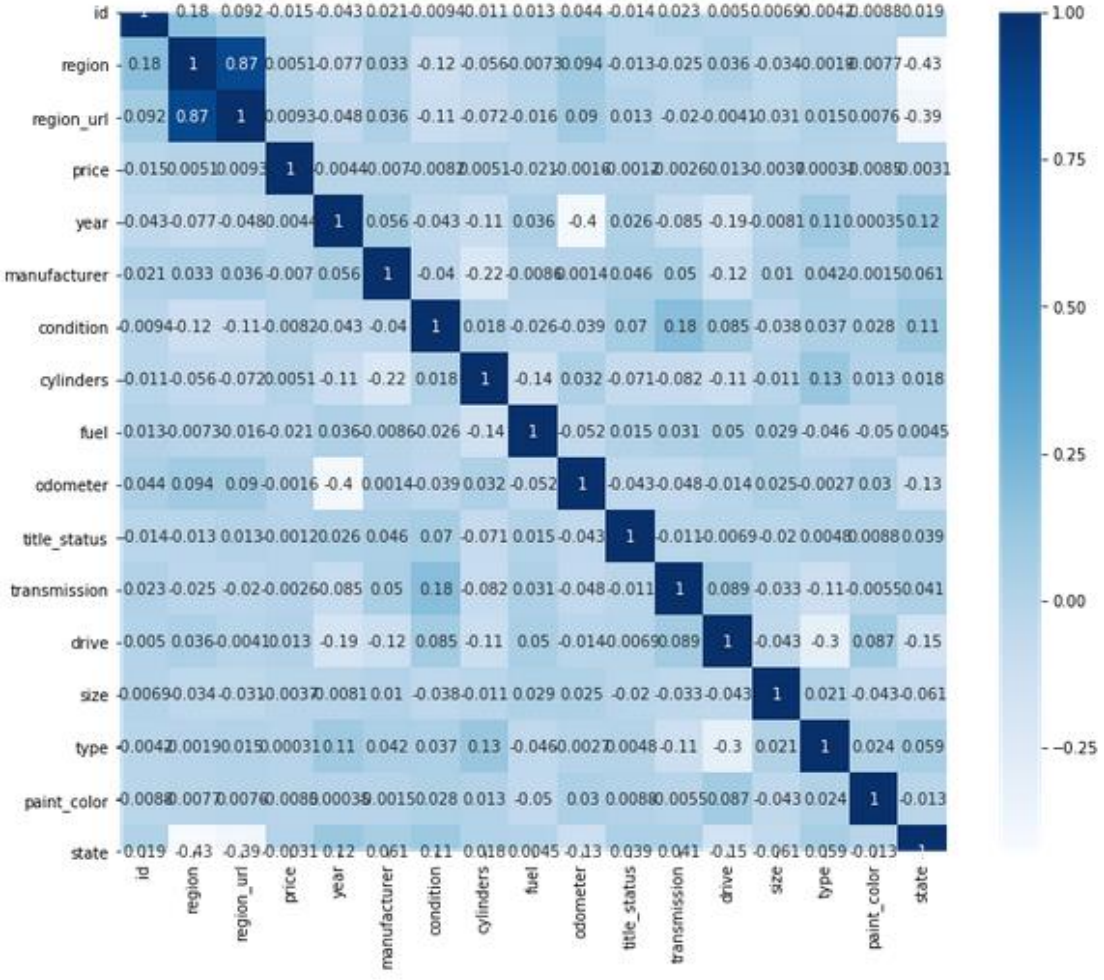
a. Menghapus atribut yang berisi url atau alamat dari *website* karena data dari atribut yang berisi url memiliki *session id* yang berbeda-beda tiap barisnya. Sehingga itu tidak diperlukan dan tidak memiliki informasi yang cukup untuk mencapai tujuan.

b. Memilih atribut ‘price’, ‘year’, ‘manufacturer’, ‘condition’, ‘cylinders’, ‘fuel’, ‘odometer’, dan ‘type’ untuk klasifikasi. Untuk *Clustering* menggunakan atribut ‘transmission’ dan ‘condition’.

Pemilihan atribut untuk klasifikasi dilakukan sebagai indicator untuk mengetahui jenis persneling yang digunakan, terutama pada mobil bekas. Hal ini digunakan untuk mempermudah montir dalam menindaklanjuti atau mengambil keputusan untuk tindakan selanjutnya apabila terjadi kerusakan atau kendala pada mobil. Sedangkan untuk *clustering*, hal ini digunakan untuk mengelompokkan jenis persneling yang paling sering digunakan pada mobil bekas. Hal ini sangat berguna bagi toko *sparepart* untuk menjaga persediaan barang yang digunakan oleh jenis persneling tertentu.

c. Metode *Pearson Correlation*

Memilih atribut yang digunakan berdasarkan nilai kolerasi yang besar antar atribut. Data yang telah dilakukan proses filter akan diproses ke dalam perhitungan kolerasi untuk dapat mengetahui seberapa dekat hubungan antara satu atribut dengan atribut lainnya. Gambar merepresentasikan hasil nilai kolerasi.



Dapat dilihat dari tabel bahwa atribut region terhadap region_url sangat berkolerasi dengan nilai 0,87 hampir mendekati 1 yang berarti korelasi linier positif.

Atribut yang akan dipakai menggunakan batasan nilai korelasi atau $\alpha > 0.05$ untuk dapat meminimalisir penggunaan atribut terlalu banyak yang dipakai. Dengan nilai α sebesar 0.05 yang menyatakan bahwa kolerasi atribut seolah sudah separuhnya serupa.

Atribut yang mempunyai nilai lebih dari *threshold* yang ditetapkan adalah 'region', 'region_url', 'year', ‘manufacturer’, 'condition', 'cylinders', 'fuel', 'odometer', 'title_status', ’transmission’, 'drive', 'type', 'paint_color', dan 'state'. Metode ini sebagai bahan pertimbangan dalam memilih beberapa fitur yang akan digunakan dalam klasifikasi maupun *clustering*. Hasil dari *Pearson Correlation* akan berpengaruh pada hubungan linier antar atribut.

d. *Split Data*

Pada setiap metode yang digunakan untuk klasifikasi maupun *Clustering* harus melakukan *learning. Learning* disini membutuhkan data latih(*data train*). Fungsi ini didapatkan dari *split data,* yaitu membagi menjadi *data train* dan *data test.* Akan tetapi, dari *Clustering* merupakan *unsupervised learning* dimana data belum ada label. Jadi, *split data* hanya dilakukan untuk *supervised learning* yaitu pada klasifikasi saja.

E. Praproses

Praproses adalah proses untuk membersihkan atau memperbaiki data apabila ditemukan missing value, atau NaN di dalam dataset. Proses ini membantu untuk membuat data agar konsisten dan menghindari ketidakseimbanagan sebuah data. Pada tugas ini akan dilakukan praproses dengan tahapan yakni mengganti nilai missing value dan NaN pada dataset, mengubah bentuk data yang string menjadi numerik pada data, pengambilan data yang akan diproses pada tahap selanjutnya.

A) Mengubah Bentuk Data yang *Float* dan *Object* menjadi *Integer* pada Data

Perlu adanya mengubah bentuk data *float* dan *object* menjadi data *integer* agar mempermudah proses dan mempengaruhi proses performansi pada suatu metode yang akan digunakan. Pada proses ini dilakukan perubahan bentuk data dari *object* menjadi *integer.* Berikut kode dari *encode* bentuk data dari *object*.

```
# encode to float for searching correlation
dataset['transmission'] = pd.to_numeric(dataset['transmission'], errors='coerce')
```

Selanjutnya, kolom ‘price’, ‘year’, ‘manufacturer’, ‘condition’, ‘cylinders’, ‘fuel’, ‘odometer’, ‘transmission’, dan ‘type’ akan diubah bentuk datanya dari *float* menjadi *integer.*

```
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()

dataset['transmission'] = encoder.fit_transform(dataset['transmission'])
```

```
data['year'] = data['year'].astype(int)
data['manufacturer'] = data['manufacturer'].astype(int)
data['condition'] = data['condition'].astype(int)
data['cylinders'] = data['cylinders'].astype(int)
data['fuel'] = data['fuel'].astype(int)
data['odometer'] = data['odometer'].astype(int)
data['type'] = data['type'].astype(int)
```

[317]:
```
data.head()
# data.head()
```

[317]:

	price	year	manufacturer	condition	cylinders	fuel	odometer	type
0	3	2	3	1	0	3	1	0
1	0	3	1	1	0	3	0	1
2	3	3	1	1	0	3	0	1
3	0	3	1	1	0	3	0	1
4	3	3	1	1	0	0	1	0

Proses perubahan bentuk data menggunakan cara yang berbeda dikarenakan nilai dari tiap kolom yang berbeda.

B) Mengubah *Missing Value*

Data yang diperoleh masih memiliki missing value, yaitu ‘price’, ‘year’, ‘manufacturer’, ‘condition’, ‘cylinders’, ‘fuel’, ‘odometer’, ‘transmission’, dan ‘type’. Berikut merupakan penjelasan data yang memiliki *missing value* pada data used_cars.

find missing value

[278]:
```
dataset.isnull().sum()
# print((data == 0).sum())
```

[278]:
```
price              0
year              12
manufacturer     705
condition       9152
cylinders       7085
fuel              73
odometer        2389
transmission       0
type            3659
dtype: int64
```

	price	year	manufacturer	condition	cylinders	fuel	odometer	transmission	type
0	17899	2012.0	38.0	0.0	3.0	2.0	63500.0	3	3.0
1	0	2016.0	12.0	0.0	NaN	2.0	10.0	0	NaN
2	46463	2015.0	13.0	0.0	NaN	2.0	7554.0	0	NaN
3	0	2016.0	12.0	0.0	NaN	2.0	10.0	0	NaN
4	49999	2018.0	12.0	NaN	NaN	0.0	70150.0	0	6.0

Kemudian, untuk menghindari adanya NaN pada data maka akan dilakukan proses untuk mengganti NaN dengan nilai rata-rata (*mean*)

dari nilai pada masing-masing kolom yang memiliki NaN *value*. Berikut kode untuk mengisi *missing value* dengan rata-rata:

```
fill missing value with the attribute mean (Data Cleaning Methods)

: dataset.fillna(dataset.mean(), inplace=True)
```

Proses ini dilakukan sampai tidak ditemukan *missing value* pada masing-masing atribut.

```
[281]: print(dataset.isnull().sum())

price           0
year            0
manufacturer    0
condition       0
cylinders       0
fuel            0
odometer        0
transmission    0
type            0
dtype: int64
```

C) Metode Quartil

Merupakan salah satu metode dalam *data preprocessing* yang akan digunakan untuk klasifikasi. Metode ini yaitu merepresentasikan data menjadi tiga *range* data berdasarkan nilai minimum atau kuartil bawah (Q1), nilai median atau kuartil tengah (Q2), dan nilai maksimum atau kuartil atas (Q3). Metode ini digunakan agar data yang mempunyai mempunyai nilai selain dari Q1, Q2, dan Q3 akan diubah sesuai dengan *range* yang telah ditentukan menurut hasil Q1, Q2, dan Q3. Berikut merupakan hasil dari atribut yang telah diubah dengan metode kuartil:

```
[327]: data_train.head()
```

[327]:

	price	year	manufacturer	condition	cylinders	fuel	odometer	tipe
0	3	2	3	1	0	3	1	0
1	0	3	1	1	0	3	0	1
2	3	3	1	1	0	3	0	1
3	0	3	1	1	0	3	0	1
4	3	3	1	1	0	0	1	0

Pada *clustering*, metode ini tidak diterapkan karena atribut data yang dipilih tidak mempunyai nilai yang variatif atau dapat dikatakan nilai datanya cenderung memiliki *range* yang serupa.

F. Pemodelan

A) Classification

1. Metode Gaussian Naïve Bayes

Merupakan salah satu metode *supervised learning* dimana data yang sudah ada label sudah diberikan sebelumnya jadi data akan masuk ke bagian tertentu. Naïve bayes sangat tepat digunakan pada data kontinu maupun diskrit. Alasan penggunaan metode ini adalah atribut yang digunakan sebagai label adalah kategorikal. Sehingga, performansi dari metode ini akan menjadi lebih baik. Selain itu, Gaussian Naïve Bayes dipilih karena label data yang digunakan berbentuk distribusi univariate atau merupakan varial dependen yang berarti bahwa atribut tersebut dipengaruhi oleh atribut lainnya. Dalam menghitung probabilitas dapat menggunakan persamaan di bawah ini:

$$P(X_i = x_i | Y = y_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}}} e^{\frac{(x_i - \mu_{ij})^2}{2\sigma^2 ij}}$$

Keterangan:

- P: Peluang
- Xi: Atribut ke-i
- xi: Nilai atribut ke-i
- Y: Kelas yang dicari
- yi: Sub kelas Y yang dicari
- μ: mean, menyatakan rata-rata atribut dari setiap kelas
- σ: Standar deviasi, nilai yang digunakan untuk menunjukkan ukuran varian

2. Metode k-Nearest Neighbors

Metode ***k-Nearest Neighbors*** adalah metode *supervised learning* dimana hasil dari *instance* yang baru diklasifikasikan berdasarkan mayoritas dari kategori **k**-tetangga terdekat. Tujuan dari metode ini adalah untuk mengklasifikasikan obyek baru berdasarkan atribut dari data *training.* Metode *k-Nearest Neighbors* menggunakan *Neighborhood Classification* sebagai nilai prediksi dari nilai *instance* yang baru. Karena atribut yang digunakan banyak dan beragam, maka metode ini dirasa cocok untuk klasifikasi. Sehingga, tujuan dari penulisan dapat tercapai. Pada model dengan metode kNN, nilai k ditentukan sebesar 3 karena nilai k yang kecil akan berpengaruh pada hasil akurasi yang tinggi.

B) Clustering

1. Metode K-Means

Data Used Cars belum memiliki label atau kelas yang mendeskripsikan data setiap baris dimana itu yang mendefinisikan bahwa data ini termasuk ke dalam *unsupervised learning*. Clustering merupakan salah satu pendekatan metode dari *unsupervised learning* dimana metode ini akan membantu untuk mengelompokkan data mana saja yang memiliki jarak yang dekat. Clustering yang digunakan adalah K-Means. K-Means adalah salah satu metode yang sangat tepat karena bentuk data yang diperoleh yakni numerik (berdasarkan hasil prapemrosesan yang telah dilakukan) dengan atribut yang sangat banyak dan beragam.

G. Eksperimen

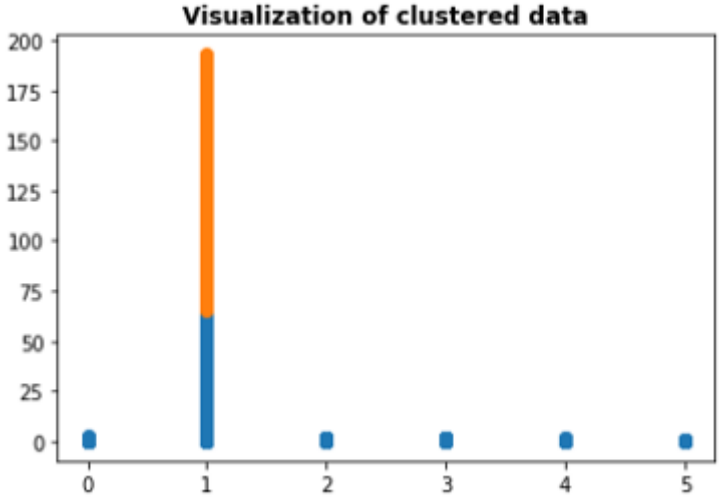
A) Classification

Eksprimen yang dilakukan untuk klasifikasi adalah menggunakan model yang berbeda, yaitu menggunakan model dengan metode Gaussian Naïve Bayes dan metode k-Nearest Neighbors. Hasil dari percobaan diukur dengan evaluasi perfomansi. Hasil evaluasi dari masing-masing model dibahas di poin Evaluasi.

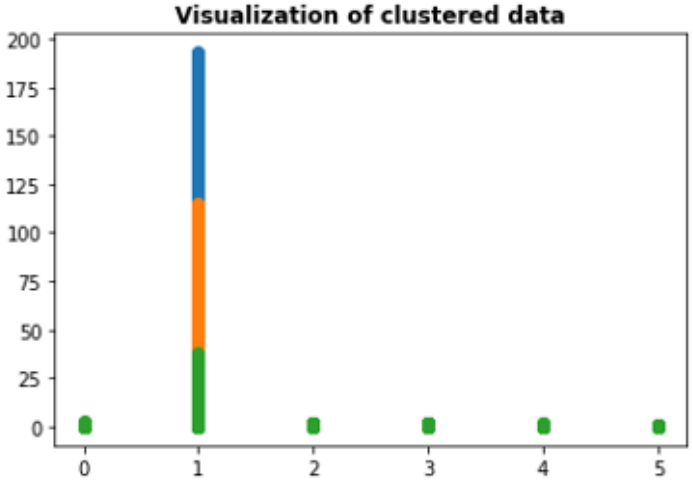
B) Clustering

Eksperimen yang dilakukan untuk *clustering* adalah membuat nilai k untuk pemodelan kluster berbeda, yaitu menggunakan nilai k = 2 dan iterasi maksimal sebesar 100, lalu k = 3 dan maksimal iterasi 150. Berikut gambar dari hasil kluster:

a. Hasil dari nilai k = 2 dan maksimal iterasi 100



b. Hasil dari nilai k = 3 dan maksimal iterasi 150



H. Evaluasi

A) Classification

Confussion Matrix merupakan salah satu metode dalam *machine learning* untuk mengukur kinerja dari suatu model khususnya pada klasifikasi (*supervised learning*). Pada dasarnya *confusion matrix* memberikan informasi perbandingan hasil klasifikasi yang dilakukan oleh sistem (model) dengan hasil klasifikasi sebenarnya. Terdapat 4 representasi yang dihasilkan oleh *confussion matrix*, yaitu:

- *True Positive* (TP): jumlah klasifikasi yang diprediksi positif dan benar.
- *True Negative* (TN): jumlah klasifikasi yang diprediksi negatif dan benar.
- *False Positif* (FP): jumlah klasifikasi yang diprediksi positif dan salah.
- *False Negative* (FN): jumlah klasifikasi yang diprediksi negatif dan salah.

Penggunaan c*onfusion matrix* untuk menghitung berbagai *performance metrics* untuk mengukur kinerja model yang telah dibuat. Beberapa *performance metrics* populer yang umum dan sering digunakan: *accuracy*, *precission*, *f1 score*, dan *recall*. Performansi *metric* yang digunakan dalam evaluasi untuk model klasifikasi yang diterapkan oleh data mobil bekas adalah *accuracy*. *Accuracy* atau Akurasi merupakan tingkat kedekatan antara nilai prediksi dengan nilai aktual. Berikut rumus dari *accuracy*:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} x100\%$$

a. Hasil Evaluasi dengan Metode Gaussian Naïve Bayes
Berikut hasil evaluasi dengan *confussion matrix* yang diperoleh dari model yang menggunakan metode Gaussian Naïve Bayes:

```
Confusion Matrix :
[[5027   10  232 ...    0    0    1]
 [ 494   12   32 ...    0    0    0]
 [  96    0  103 ...    0    0    0]
 ...
 [   1    0    0 ...    0    0    0]
 [   0    0    0 ...    0    0    0]
 [   0    0    0 ...    0    0    0]]
Accuracy Score : 0.7789728828965309
Report :
              precision    recall  f1-score   support

           0       0.89      0.87      0.88      5755
           1       0.55      0.02      0.04       566
           2       0.28      0.50      0.36       204
           3       0.00      0.00      0.00         0
           4       0.00      0.00      0.00         1
```

Didapatkan nilai *accuracy* sebesar 0,78 atau 78%. Ini berarti sebanyak 78% dari *data train* keseluruhan benar dengan membandingkan dengan *data test*. Selain itu, menggunakan *confussion matrix* juga mudah secara visualisasi untuk melihat kesalahan prediksi. Berdasarkan gambar di atas, dapat dijelaskan bahwa model dari metode Gaussian Naïve Bayes diperoleh nilai TP sebesar 5027 yang berarti prediksi yang dihasilkan dari metode ini benar, FP sebesar 10 berarti klasifikasi yang diprediksi negatif dan benar, FN sebesar 494 jumlah yang diprediksi positif dan salah, TN sebesar 12 berarti jumlah klasifikasi yang diprediksi negatif dan salah.

b. Hasil Evaluasi dengan Metode k-Nearest Neighbors
Berikut hasil evalusi yang diperoleh dengan *confussion matrix* yang diperoleh dari model dengan menggunakan metode k-Nearest Neighbors:

```
Confusion Matrix:
[[5437  240   78 ...    0    0    0]
 [ 440  120    6 ...    0    0    0]
 [ 118    9   77 ...    0    0    0]
 ...
 [   1    0    0 ...    0    0    0]
 [   1    0    0 ...    0    0    0]
 [   1    0    0 ...    0    0    0]]
Accuracy Score : 0.853507044387214
Report:
              precision    recall  f1-score   support

           0       0.90      0.94      0.92      5755
           1       0.33      0.21      0.26       566
           2       0.48      0.38      0.42       204
           4       0.00      0.00      0.00         1
           5       0.00      0.00      0.00         1
           9       0.00      0.00      0.00         1
```

Didapatkan nilai *accuracy* sebesar 0,85 atau 85%. Ini berarti sebanyak 85% dari *data train* keseluruhan benar dengan membandingkan dengan *data test*. Selain itu, menggunakan *confussion matrix* juga mudah secara visualisasi untuk melihat kesalahan prediksi. Berdasarkan gambar di atas, dapat dijelaskan bahwa model dari metode k-Nearest Neighbors diperoleh nilai TP sebesar 5437 yang berarti prediksi yang dihasilkan dari metode ini benar, FP sebesar 240 berarti klasifikasi yang diprediksi negatif dan benar, FN sebesar 440 jumlah yang diprediksi positif dan salah, TN sebesar 120 berarti jumlah klasifikasi yang diprediksi negatif dan salah.

B) Clustering

I. Kesimpulan

a. Classification

Dari keseluruhan proses yang dijalankan dalam klasifikasi, bagian yang paling penting terletak pada data preprosesing yang mana membuat data menjadi bersih yang berarti secara bentuk data adalah sama, konsisten, dan lengkap. Sehingga informasi yang diperoleh jelas dan lengkap, tidak ada data yang dihapus ataupun dikurangi.

Dalam membuat model klasifikasi menggunakan dua metode yang berbeda, diperoleh bahwa model dengan metode k-Nearest Neighbour memiliki akurasi lebih tinggi yaitu sebesar 85%, dibandingkan dengan metode Naïve Bayes yang akurasinya sebesar 78%. Hal ini membuktikan bahwa penggunaan metode untuk pemodelan sangat mempengaruhi hasil dari akurasi.

Saran untuk *improvement* selanjutnya yaitu mencoba menggunakan seluruh atribut dari 'region', 'region_url', 'year', ‘manufacturer’, 'condition', 'cylinders', 'fuel', 'odometer', 'title_status', 'drive', 'type', 'paint_color', dan 'state'. Atribut

tersebut merupakan hasil dari metode Pearson Correlation. Selain itu, penerapan metode dalam membuat pemodelan juga sangat berpengaruh dalam menghasilkan akurasi. Jika ingin menggunakan metode yang sama, khususnya untuk metode k-Nearest Neighbors diharapkan mencari nilai k dengan menggunakan *cross validation* dari *data train*. Dengan menerapkan beberapa atribut yang mendekati nilai *perfect positive linear correlation* atau memiliki nilai korelasi lebih dari nilai α sebesar 0,05 dan penggunaan metode yang tepat untuk klasifikasi diharapkan dapat meningkatkan akurasi.

b. Clustering

Dari keseluruhan proses yang dijalankan dalam *clustering*, bagian yang paling penting terletak pada data preprosesing yang mana membuat data menjadi bersih yang berarti secara bentuk data adalah sama, konsisten, dan lengkap. Sehingga informasi yang diperoleh jelas dan lengkap, tidak ada data yang dihapus ataupun dikurangi. Selain itu, pada penerapan metode pemodelan hal yang terpenting terletak pada pemilihan nilai k dan juga *update centroid* saat menentukan bagaimana metode K-Means berhenti.

Saran untuk *improvement* selanjutnya yaitu mencoba menggunakan seluruh atribut untuk *clustering* dari 'region', 'region_url', 'year', ‘manufacturer’, 'condition', 'cylinders', 'fuel', 'odometer', 'title_status', 'drive', 'type', 'paint_color', dan 'state'. Atribut tersebut merupakan hasil dari metode Pearson Correlation. Selain itu, untuk mendapatkan hasil yang optimal dapat juga mencari atau menentukan nilai k yang optimal dengan metode *elbow*.