Teknik Clustering dan Classification Menggunakan Pendekatan Algoritma K-Means, ID3 dan Naïve Bayes Pada Data Air_bnb

Irawansyah
Jurusan Teknik Informatika
Fakultas Informatika
Universitas Telkom, Bandung
Email : irawansyah@student.telkomuniversity.ac.id

1. PENDAHULUAN

1.1 Latar Belakang Masalah

Ketersediaan data yang berlimpah yang dihasilkan dari penggunaan teknologi informasi di hampir semua bidang kehidupan, menimbulkan kebutuhan untuk dapat memanfaatkan informasi dan pengetahuan yang terkandung di dalam limpahan data tersebut, yang kemudian melahirkan *data mining*. *Data mining* merupakan proses untuk menemukan pengetahuan (*knowledge discovery*) yang ditambang dari sekumpulan data yang volumenya sangat besar. Aplikasi *data mining* pada pengelolaan bisnis, pengendalian produksi, dan analisa pasar misalnya, memungkinkan diperolehnya pola dan hubungan yang dapat dimanfaatkan untuk pengambilan keputusan. Teknik-teknik yang digunakan untuk pengekstrakan pengetahuan dalam data mining adalah Clustering dan Klasifikasi.

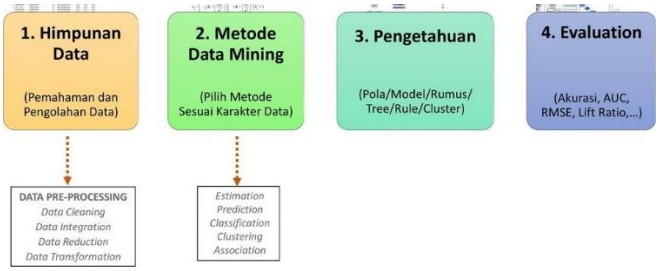
Clustering digunakan untuk pengelompokkan data berdasarkan kemiripan pada objek data dan sebaliknya meminimalkan kemiripan terhadap kluster yang lain. Klasifikasi adalah pemprosesan untuk menemukan sebuah model atau fungsi yang menjelaskan dan mencirikan konsep atau kelas data, untuk kepentingan tertentu. Pada tugas kali ini diberikan data air_bnb yang memiliki data sebanyak `(22552, 16)`. Berdasarkan latar belakang tersebut, maka akan dilakukan penelitian dengan Teknik clustering dengan metode klasifikasi dengan memanfaatkan data yang bervolume besar. Algoritma yang digunakan untuk clustering yang akan dipakai adalah K-means sedangkan algoritma klasifikasi yang akan dipakai adalah algoritma ID3 dan naïve bayes. Pemilihan algoritma kmeans dikarenakan merupakan sebuah metode clustering yang paling sederhana dan umum dan juga karena k-means mempunyai kemampuan mengelompok kan data dalam jumlah yang cukup besar dengan waktu komputasi yang cepat dan efisien. Dan metode Naïve Bayes adalah klasifikasi statistic yang dapat digunakan untuk memprediksi suatu kelas

1. 2 Tujuan

Tujuan dari penelitian ini adalah untuk mengelompokkan data air_bnb dengan menggunakan Teknik clustering dan memprediksi suatu kelas dengan menggunakan Teknik Naïve Bayes

2. METODE

Metode yang digunakan adalah sebagi berikut



1. Data Pre-processing

Pada tahap ini dilakukan membersihkan data dari hal-hal yang tidak dinginkan. Sebagai berikut :

- Data Cleaning

untuk menghapus row atau kolom yang memiliki missing values

```
datasets.dropna(inplace=True)
datasets.shape
```

```
(18587, 16)
```

– Label Encoding

Ketika kita melakukan klasifikasi, biasanya kita berurusan dengan banyak label. Label-label ini bisa dalam bentuk kata-kata, angka, atau sesuatu yang lain. Fungsi pembelajaran mesin dalam sklearn mengharapkan mereka menjadi angka

```
[15] categorical = datasets.dtypes==object
     categorical_cols = datasets.columns[categorical].tolist()
     datasets[categorical_cols] = datasets[categorical_cols].apply(lambda col: LabelEncoder().fit_transform(col))
     datasets[categorical_cols].head()
     print(datasets)
```

```
             id   name  ...  calculated_host_listings_count  availability_365
0          2015   2860  ...                               4               141
1          2695  11798  ...                               1                 0
2          3176   7410  ...                               1               220
3          3309   2908  ...                               1               297
4          7071   4020  ...                               1                26
...         ...    ...  ...                             ...               ...
22449  29772631   5450  ...                               1                32
22463  29784461   8917  ...                               2               339
22475  29792124  15770  ...                               1                 8
22508  29817383  11815  ...                               1               350
22536  29849565  15046  ...                               1                69

[18422 rows x 16 columns]
```

2. Analisis Pemilihan Algoritma

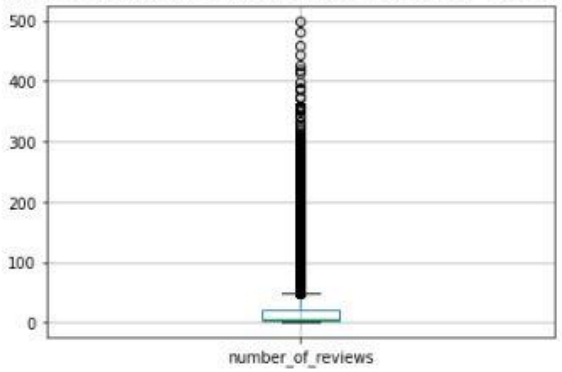
Penelitian ini menggunakan algoritma *K-means* untuk menyelesaikan masalah yang sudah saya jelaskan sebelumnya. Alasannya menggunakan algoritma ini karena

- **Normalization**

Menghapus outlier data dengan menggunakan box-plot untuk menunjukan ada tidaknya outlier

```
[18] databaru.boxplot(column=["number_of_reviews"])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3a44a8e668>
```

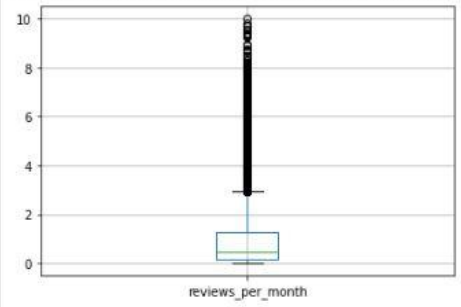



```
[20] databaru.drop(databaru[databaru.number_of_reviews >= 100].index ,inplace =True)
```

Bisa kita lihat dari boxplot diatas terdapat banyak sekali outlier data diatas 100 maka akan dilakukan drop apabali datanya lebih dari 100 pada atribut number_of_reviews

```
[22] databaru.boxplot(column=["reviews_per_month"])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3a4a5f7828>
```




```
[23] databaru.drop(databaru[databaru.reviews_per_month > 3].index ,inplace =True)
```

Bisa kita lihat dari boxplot diatas terdapat banyak sekali outlier data diatas 3 maka akan dilakukan drop apabali datanya lebih dari 3 pada atribut reviews per month

-Scalling

Untuk membuat numerical data pada dataset memiliki rentang nilai (scale) yang sama.

```
[26] mms = MinMaxScaler()
     scaler = mms.fit_transform(databaru)
     col_new = ["id","number_of_reviews","reviews_per_month"]
     datascalling = pd.DataFrame(scaler,columns=col_new)
     datascalling
```

3. Pengetahuan

3.1 Clustering

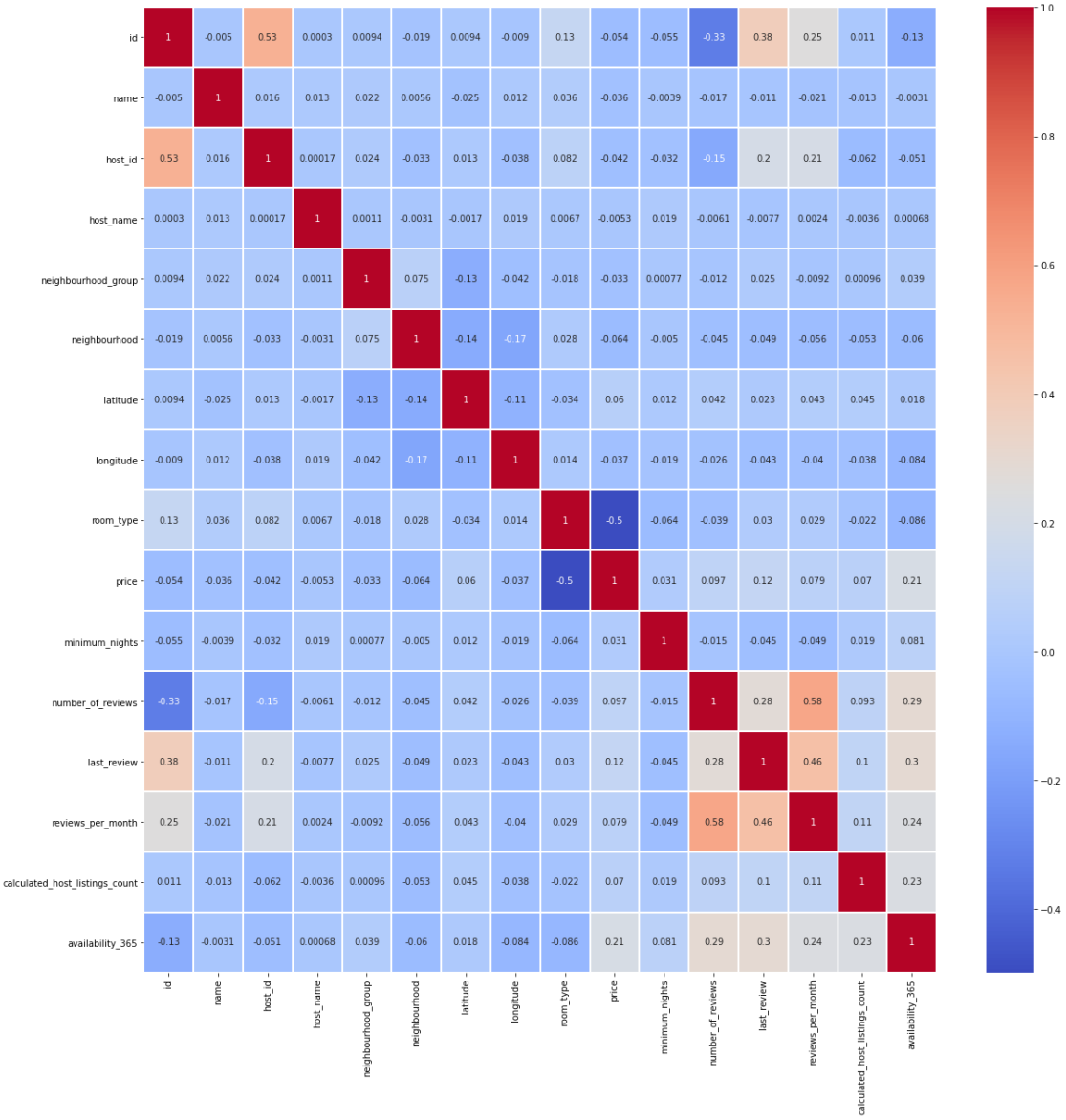
Merupakan sebuah metode clustering yang paling sederhana dan umum dan juga karena k-means mempunyai kemampuan mengelompok kan data dalam jumlah yang cukup besar dengan waktu komputasi yang cepat dan efisien. *K*-Means merupakan salah satu algoritma klastering dengan metode partisi (partitioning method) yang berbasis titik pusat (centroid) selain algoritma *k*-Medoids yang berbasis obyek. Algoritma *k*-Means dalam penerapannya memerlukan tiga parameter yang seluruhnya ditentukan pengguna yaitu jumlah *cluster k*, inisialisasi klaster, dan jarak system. Algoritma *K-mean* mendefinisikan *centroid* atau *pusat cluster* dari *cluster* menjadi rata-rata point dari *cluster* tersebut.Dalam penerapan algoritma *k*-Means, jika diberikan sekumpulan data $X = \{x_1, x_2, \ldots, x_n\}$ dimana xi = $(x_{i1}, x_{i2}, \ldots, x_{in})$ adalah ystem dalam ruang real *Rn*, maka algoritma *k*-Means akan menyusun partisi *X* dalam sejumlah *k cluster* (a priori). Setiap *cluster* memiliki titik tengah (*centroid)* yang merupakan nilai rata rata (*mean)* dari data-data dalam *cluster* tersebut. Tahapan awal, algoritma *k*-Means adalah memilih secara acak *k* buah obyek sebagai *centroid* dalam data. Kemudian, jarak antara obyek dan *centroid* dihitung menggunakan *Euclidian distance*. Algoritma *k*-Means secara *iterative* meningkatkan variasi nilai dalam dalam tiap tiap *cluster* dimana obyek selanjutnya ditempatkan dalam kelompok yang terdekat, dihitung dari titik tengah klaster. Titik tengah baru ditentukan bila semua data telah ditempatkan dalam *cluster* terdekat

Langkah- langkah Clustering sebagai berikut :

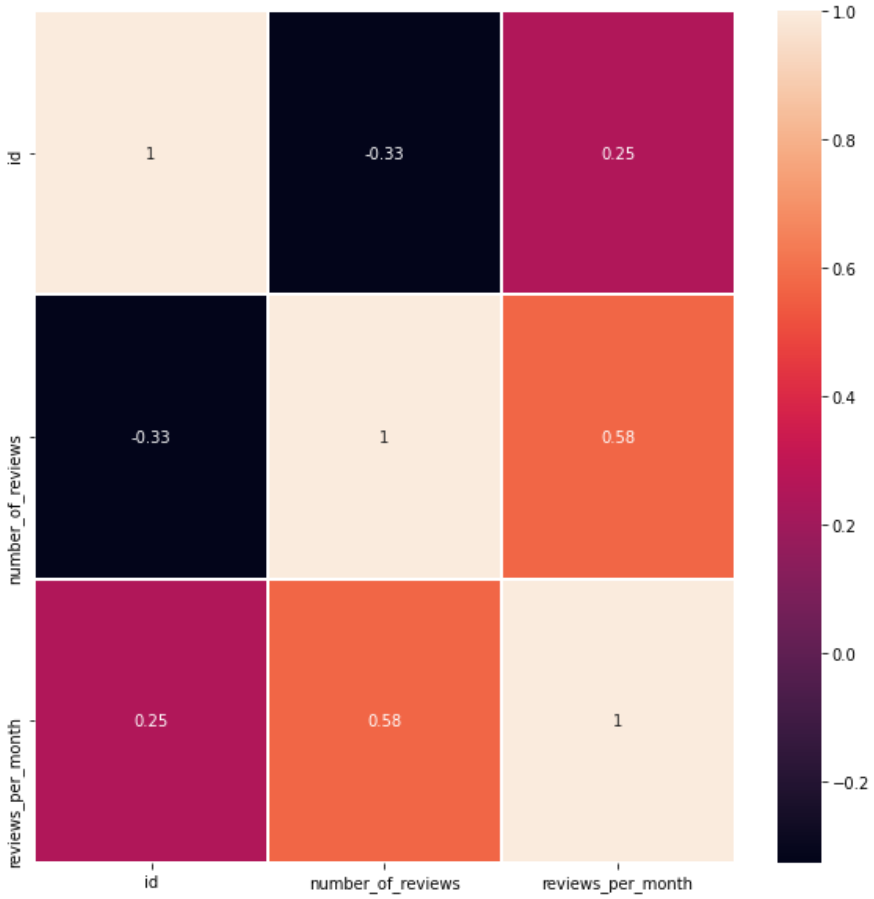
Sebelum memulai clustering saya memilih 2 atribut dari dataset yaitu Airbnb yaitu "number of review" dan "reviews_per_month" dengan menggunakan Correlation Matrix untuk memilih atribut yang akan digunakan.

```
plt.figure(figsize=(20,20))
sns.heatmap(datasets.corr(),
            cmap='coolwarm',
            annot=True,linewidths=1);
```

dengan hasil plot sebagai berikut :



Cek lagi Correlaction antar atribut dengan menggunakan heatmap

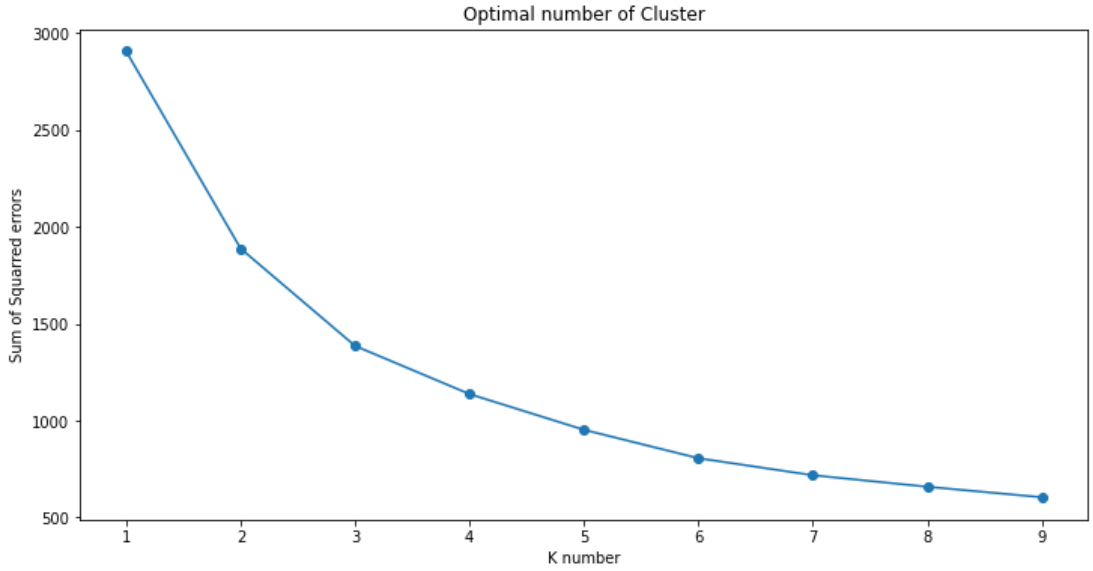


Bisa dilihat, korelasi tertinggi didapat dari atribut dengan **0.58** yaitu "number of review" dan "reviews_per_month" karena apabila jumlah ulasan tinggi pada penginapan rumahan (Airbnb) maka akan tinggi juga ulasannya perbulan. Dan untuk kasus ini,batas nilai korelasi tertinggi adalah 1 karena 1 pasti atribut itu sendiri

Langkah 1: Tentukan berapa banyak *cluster* k dari dataset yang akan dibagi.

```
kmeans = KMeans(n_clusters = 3)
```

Langkah 2: Tetapkan secara acak data k menjadi pusat awal lokasi klaster.

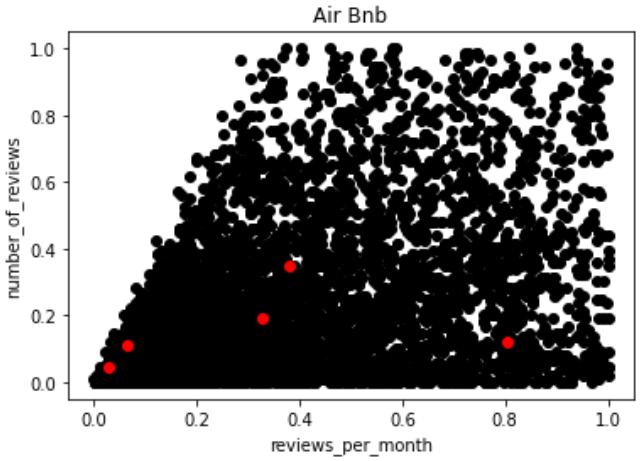


Berdasarkan grafik diatas, dapat dilihat bahwa Method *elbow* terlihat saat jumlah klaster adalah 5. Oleh karena itu, nilai *k* yang baik adalah 5.

Langkah 3: Untuk masing-masing data, temukan pusat *cluster* terdekat. Dengan demikian berarti masing-masing pusat *cluster* memiliki sebuah subset dari dataset, sehingga mewakili bagian dari dataset. Oleh karena itu, telah terbentuk *cluster* k: C1, C2, C3, …, Ck .

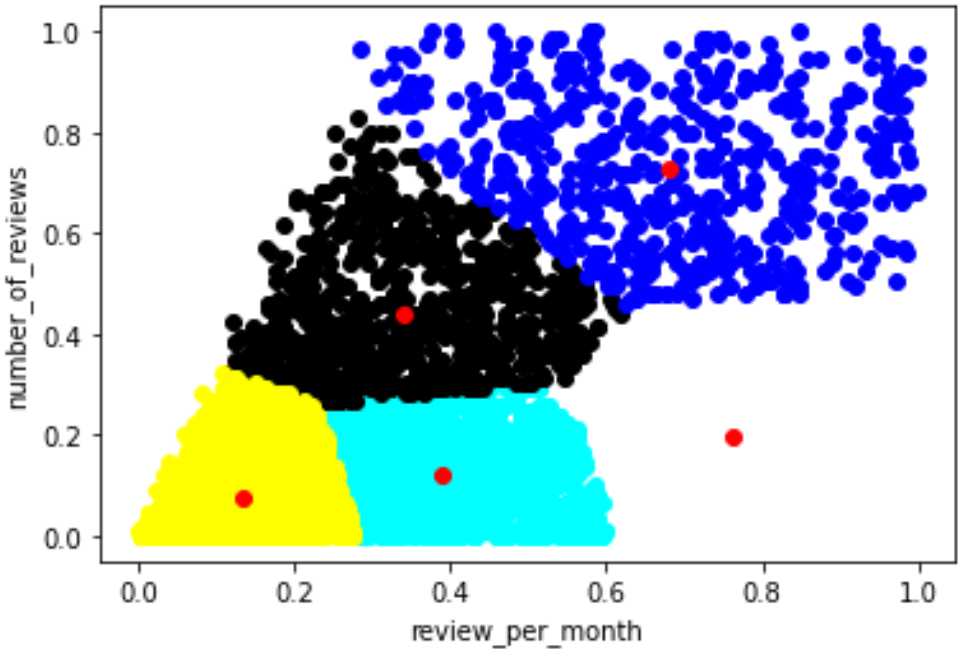
```
newscale = datascalling[["reviews_per_month","price"]]
K=5
Centroids = (newscale.sample(K))
```

Langkah 4: Untuk masing-masing *cluster* k, temukan pusat luasan klaster, dan perbarui lokasi dari masing-masing pusat *cluster* ke nilai baru dari pusat luasan.



Langkah 5: Ulangi langkah ke-3 dan ke-5 hingga data-data pada tiap *cluster* menjadi terpusat atau selesai.

Setelah melakukan 5 langkah tersebut Lakukan plot lagi bisa dilihat terjadi perubahan titik centroid dan data sudah dicluster berdasarkan rata-rata(mean) dari sekumpulan data tersebut



Setelah itu saya memberi label dari scalar menjadi label kelas berdasarkan hasil cluster tersebut, jika hasil clusternya 0 maka diberi label ‘ Tidak Bagus’ , 1 diberi label ‘Kurang Bagus’ dan yang terakhir 2 maka akan diberi label ‘Bagus’

3.2 Classification

Klasifikasi adalah pemprosesan untuk menemukan sebuah model atau fungsi yang menjelaskan dan mencirikan konsep atau kelas data, untuk kepentingan tertentu. Tujuan ‘classification’ adalah untuk menganalisa data historis yang disimpan dalam database dan secara otomatis menghasilkan suatu model yang bisa memprediksi perilaku di masa mendatang. Model induksi ini terdiri dari generalisasi pada baris-baris data yang digunakan untuk pelatihan, yang akan membantu membedakan class-class standar. Harapannya adalah bahwa model tersebut kemudian bisa digunakan untuk memprediksi class-class dari baris-baris lain yang belum diklasifikasikan, dan lebih penting lagi, bisa secara akurat memprediksi peristiwa-peristiwa aktual mendatang. Disini saya menggunakan 2 algoritma classification yaitu ID3 dan Naïve Bayes. Alasannya menggunakan 2 algoritma tersebut adalah ingin membandingkan accuracy dari kedua algoritma tersebut. Kenapa memilih Naïve Bayes karena Naïve Bayes bekerja lebih baik disbanding dengan model classifier lainnya dan juga membutuhkan jumlah data training yang kecil untuk menentukan estimasi parameter yang diperlukan dalam proses pengklasifikasian

Adapun tahapnya yaitu meng-split dataset menjadi 80% untuk Data Train dan 20% Data Test

```
[ ]  x = datascalling[['number_of_reviews','reviews_per_month']]
     y = datascalling[['cluster']]
```

```
[ ]  X_train, X_test, y_train, y_test = train_test_split(x,y, test_size = 0.2)
```

Setelah itu untuk buat modelnya dan latih model yang sudah dibuat dengan data latih Dan lakukan prediksi untuk data uji
Model Decision Tree :

```
from sklearn.tree import DecisionTreeClassifier
model_dt = DecisionTreeClassifier(criterion='entropy')
model_dt.fit(X_train,y_train)
y_pred_ID3 = model_dt.predict(X_test)
y_pred_ID3
```

```
array(['Bagus', 'Tidak Bagus', 'Bagus', ..., 'Tidak Bagus',
       'Kurang Bagus', 'Bagus'], dtype=object)
```

Model Naïve Bayes :

```
# Import Gaussian Naive Bayes model
from sklearn.naive_bayes import GaussianNB

#Buat model Gaussian Classifier
model = GaussianNB()

#Latih model menggunakan data latih
model.fit(X_train, y_train)

#Prediksi respons untuk dataset uji
y_pred_NB = model.predict(X_test)
y_pred_NB
```

```
array(['Bagus', 'Tidak Bagus', 'Bagus', ..., 'Tidak Bagus',
       'Kurang Bagus', 'Bagus'], dtype='<U12')
```

4 Evaluasi

Setetelah tahapan pemodelan dilakukan maka selanjutnya adalah tahapan evaluasi dengan melihat hasil dari tahapan pemodelan yang telah dilakukan. Evaluasi yang dilakukan menggunakan akurasi score, precision,recall,f1-scroe dan juga support. *Accuracy* menggambarkan seberapa akurat model dapat mengklasifikasikan dengan benar, *Precision* menggambarkan tingkat keakuratan antara data yang diminta dengan hasil prediksi yang diberikan oleh model, *Recall* menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi

```
Confusion Matrix :
[[179   8   0]
 [ 29 329 120]
 [ 15  33 439]]
Accuracy Score : 0.8220486111111112
Report :
              precision    recall  f1-score   support

       Bagus       0.80      0.96      0.87       187
Kurang Bagus       0.89      0.69      0.78       478
 Tidak Bagus       0.79      0.90      0.84       487

    accuracy                           0.82      1152
   macro avg       0.83      0.85      0.83      1152
weighted avg       0.83      0.82      0.82      1152
```

ID3 :

```
Confusion Matrix :
[[177   9   1]
 [ 13 444  21]
 [  2  38 447]]
Accuracy Score : 0.9270833333333334
Report :
              precision    recall  f1-score   support

       Bagus       0.92      0.95      0.93       187
Kurang Bagus       0.90      0.93      0.92       478
 Tidak Bagus       0.95      0.92      0.94       487

    accuracy                           0.93      1152
   macro avg       0.93      0.93      0.93      1152
weighted avg       0.93      0.93      0.93      1152
```

Dapat dilihat accuracy Score dengan menggunakan ID3 mencapai 92 % sedangkan Naïve Bayes mencapai 82 %

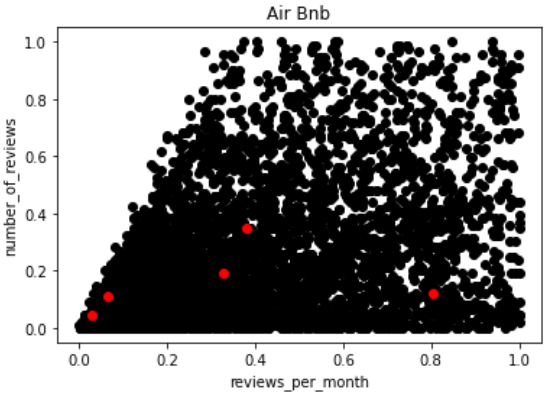
Dan bisa kita lihat perbandingan akurasinya seperti dibawah ini

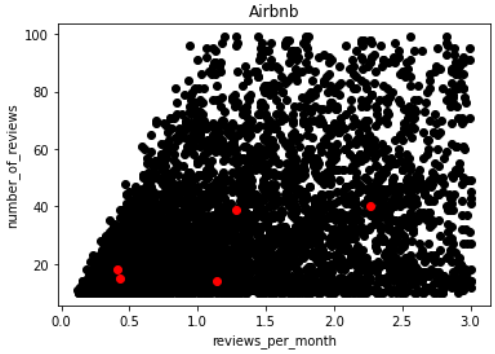
	Hasil Sebenarnya	Naive Bayes	ID3
2048	Bagus	Bagus	Bagus
4379	Kurang Bagus	Tidak Bagus	Tidak Bagus
2728	Bagus	Bagus	Bagus
5165	Kurang Bagus	Kurang Bagus	Kurang Bagus
178	Tidak Bagus	Bagus	Tidak Bagus
...	...	...	...
4144	Kurang Bagus	Kurang Bagus	Kurang Bagus
1192	Bagus	Bagus	Bagus
1707	Tidak Bagus	Tidak Bagus	Tidak Bagus
5297	Kurang Bagus	Kurang Bagus	Kurang Bagus

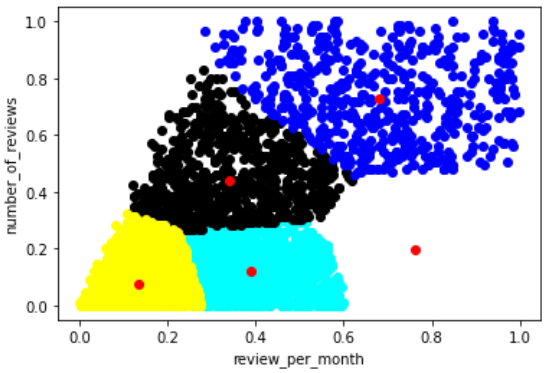
3. ANALISIS DAN HASIL

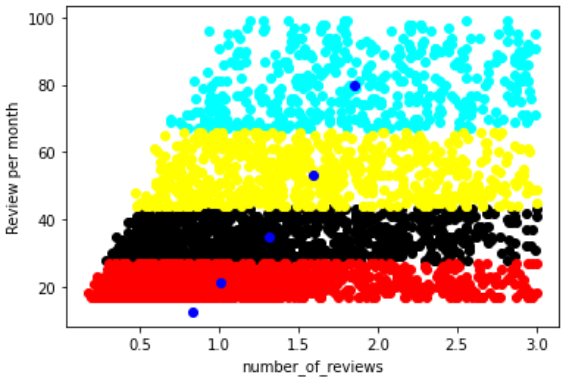
Pada tahap ini saya melakukan 2 eksperimen dengan “Teknik penyiapan data yang berbeda” yaitu dengan scalling dan tidak

Perbandingan plot Cluster dengan Scalling dan tidak :










```
[ ] datascalling.groupby("cluster").size()
```

```
cluster
Bagus           986
Kurang Bagus    2420
Tidak Bagus     2350
dtype: int64
```

Menggunakan Scalling

```
[64] databaru.groupby("cluster").size()
```

```
cluster
Bagus           1924
Kurang Bagus    2171
Tidak Bagus     1661
dtype: int64
```

Tidak Menggunakan Scalling

Dari hasil diatas dapat kita analisis hasil Cluster berbeda karena data yang tidak mengalami Scalling cenderung lebih sedikit hasil yang bagus dikarenakan Min-Max Scaling bekerja dengan scaling data/menyesuaikan data dalam rentang/range tertentu.

Klasifikasi dengan Naïve Bayes(dengan Scalling)

```
Confusion Matrix :
[[179   8   0]
 [ 29 329 120]
 [ 15  33 439]]
Accuracy Score : 0.8220486111111112
Report :
              precision    recall  f1-score   support

       Bagus       0.80      0.96      0.87       187
Kurang Bagus       0.89      0.69      0.78       478
 Tidak Bagus       0.79      0.90      0.84       487

    accuracy                           0.82      1152
   macro avg       0.83      0.85      0.83      1152
weighted avg       0.83      0.82      0.82      1152
```

Klasifikasi dengan ID3 (dengan Scalling)

```
Confusion Matrix :
[[177   9   1]
 [ 13 444  21]
 [  2  38 447]]
Accuracy Score : 0.9270833333333334
Report :
              precision    recall  f1-score   support

       Bagus       0.92      0.95      0.93       187
Kurang Bagus       0.90      0.93      0.92       478
 Tidak Bagus       0.95      0.92      0.94       487

    accuracy                           0.93      1152
   macro avg       0.93      0.93      0.93      1152
weighted avg       0.93      0.93      0.93      1152
```

```
Confusion Matrix :
[[ 48 314  10]
 [ 84 275  80]
 [ 59 227  55]]
Accuracy Score : 0.328125
Report :
              precision    recall  f1-score   support

       Bagus       0.25      0.13      0.17       372
Kurang Bagus       0.34      0.63      0.44       439
 Tidak Bagus       0.38      0.16      0.23       341

    accuracy                           0.33      1152
   macro avg       0.32      0.31      0.28      1152
weighted avg       0.32      0.33      0.29      1152
```

```
Confusion Matrix :
[[  0   0 372]
 [  0   0 439]
 [  0   0 341]]
Accuracy Score : 0.2960069444444444
Report :
              precision    recall  f1-score   support

       Bagus       0.00      0.00      0.00       372
Kurang Bagus       0.00      0.00      0.00       439
 Tidak Bagus       0.30      1.00      0.46       341

    accuracy                           0.30      1152
   macro avg       0.10      0.33      0.15      1152
weighted avg       0.09      0.30      0.14      1152
```

Klasifikasi dengan Naïve Bayes(Tanpa Scalling)

Klasifikasi dengan ID3 (tanpa Scalling)

Dari hasil diatas, dapat disimpulkan bahwa Klasifikasi dengan melakukan Scalling terlebih dahulu untuk atribut reviews_per_month dan number of reviews berhasil mendapatkan akurasi sebesar **82,2%** dengan data training dan testing 80-20 dengan algoritma Naïve Bayes dan dengan menggunakan ID3 mendpat akurasi sebesar **92,7%** dengan data training dan testing 80-20

Dan Klasifikasi tanpa melakukan Scalling untuk atribut reviews_per_month dan number of reviews berhasil mendapatkan akurasi sebesar **32,8%** dengan data training dan testing 80-20 dengan algoritma Naïve Bayes dan dengan menggunakan ID3 mendpat akurasi sebesar 0**,29%** dengan data training dan testing 80-20. Dapat disimpulkan bahwa, mengunakan persiapan data yang berbeda dapat mempengaruhi hasil akurasi yang didapat. Selain, itu teknik scaling sangat mempengaruhi hasil akurasi.

4. Daftar Pustaka

Defiyanti, S., & Jajuli, M. (2015). Integrasi metode klasifikasi dan clustering dalam data mining. *Konferensi Nasional Informatika (KNIF)*, *10*(15), 39-44.