Fake News Predictions



Oleh :

Dhoni Hanif Supriyadi

1. Business Understanding

Business Understanding adalah salah satu tahap data science yang pertama kali dilakukan oleh seorang Data Scientist. Pada tahap ini, seorang Data Scientist harus mengerti tentang bisnis tersebut. Di masa sekarang ini, tentu kita tidak bisa lepas dari berita palsu atau berita bohong. Berita – berita ini tentunya sangat mengganggu kita karena kita tidak tahu apakah berita itu palsu atau tidak bahkan sebagian dari kita turut mempercayai berita yang tergolong palsu. Oleh karena itu, saya akan membuat sebuah sistem, dimana sistem ini akan sangat berguna untuk memprediksi apakah suatu berita itu palsu atau tidak.

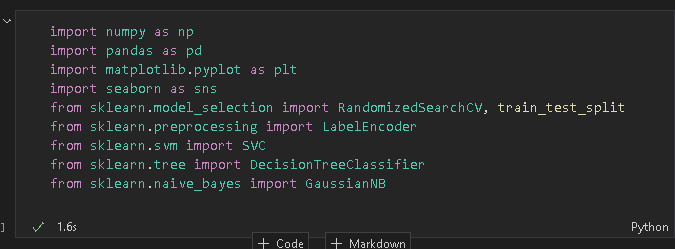
1. Data Collection

Pada tahapan ini, seorang Data Scientist akan mengumpulkan datanya terlebih dahulu. Disini, saya telah mengumpulkan data berita palsu dan data berita real. Saya mendapatkan data ini berasal dari kaggle.com. Data ini berjumlah 23 ribu data dengan 2 kategori yaitu berita real dan berita palsu.

1. Data Preparation

Pada tahapan ini, seorang Data Scientist akan membaca, mengecek, dan membersihkan datanya terlebih dahulu. Apabila terdapat nilai null, maka seorang Data Scientist akan mengisinya dengan nilai mean, median, dan modus bahkan bisa juga menghapus data yang berisi nilai null tersebut. Disini, Saya menggunakan python dan jupyter notebook.

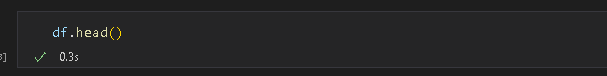
Pertama yang saya lakukan adalah mengimport library – library yang dibutuhkan seperti gambar berikut.



Selanjutnya, saya melakukan pembacaan pada data csv. Melihat data ini berbentuk csv, maka saya menggunakan metode pandas.read\_csv seperti gambar berikut.



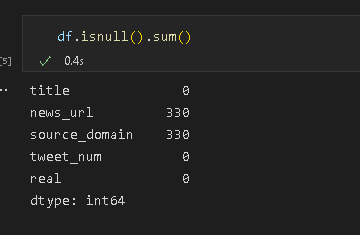
Lalu, saya mengecek 5 data teratas dengan metode head() dari library pandas.



Outputnya adalah sebagai berikut.

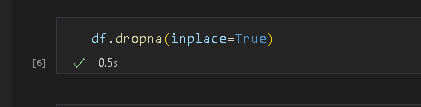


Selanjutnya, saya mengecek apakah ada missing value atau tidak dengan metode isnull() dari library pandas seperti berikut.

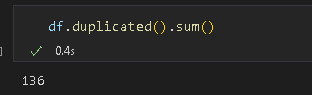


Seperti yang kita lihat, ternyata terdapat missing value pada data news\_url dan data source\_domain. Data news\_url adalah data yang berisi berbagai macam url atau situs – situs berita yang ada sedangkan data source domain adalah data yang berisi berbagai sumber situs atau home page dari situs news\_url.

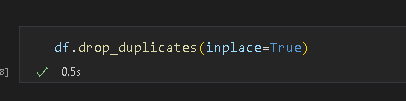
Melihat adanya missing value, alangkah baiknya kita menghilangkan missing value tersebut karena seperti kita ketahui, data tersebut adalah data bertipe objek dan tidak bisa asal mengisinya. Kita hapus dengan metode dropna() dari library pandas seperti berikut.



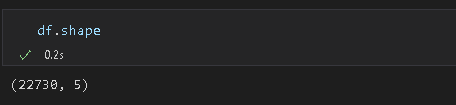
Data yang berisi missing value berhasil kita hapus. Selanjutnya, yang saya lakukan adalah mengecek data duplikat karena data duplikat akan sangat mempengaruhi performance dari model yang telah dibuat. Saya mengeceknya dengan metode duplicated() dari library pandas seperti berikut.



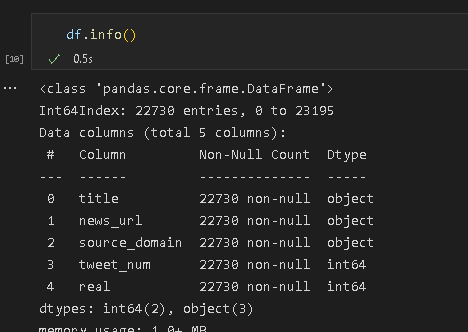
Seperti yang kita lihat, ternyata terdapat 136 data duplikat. Alangkah baiknya kita segera menghapusnya dengan metode drop\_duplicates() dari library pandas.



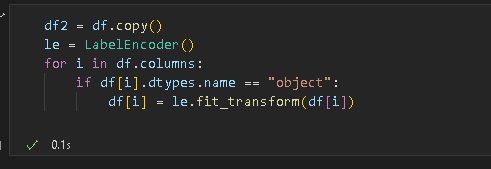
Seperti yang terlihat, data duplikat berhasil kita hapus. Selanjutnya mengecek jumlah data kembali karna pastinya data sudah berkurang sangat banyak. Cek data dengan motode shape dari library pandas seperti berikut.



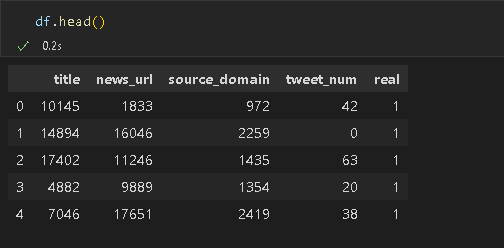
Seperti yang terlihat, data ini sekarang berjumlah 22.730 data dengan 4 variable indenpenden dan 1 variable dependen. Selanjutnya kita mengecek informasi dari data mengenai nilai null, columns, dan tipe masing – masing dari data dengan metode info() dari library pandas.



Seperti yang kita lihat, dari 5 variabel ini, terdapat 3 variabel dengan tipe data objek dan 2 variabel dengan tipe data integer. Selanjutnya, Saya melakukan encoding terhadap data bertipe objek agar data dapat diproses dengan machine learning. Saya memakai Label Encoder untuk mengencode data yang bertipe objek seperti berikut.



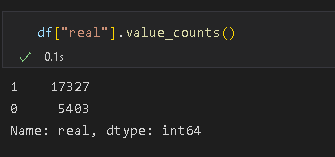
Lalu, saya mengecek kembali 5 data dari atas seperti apa datanya sekarang setelah dibersihkan dan diencode.



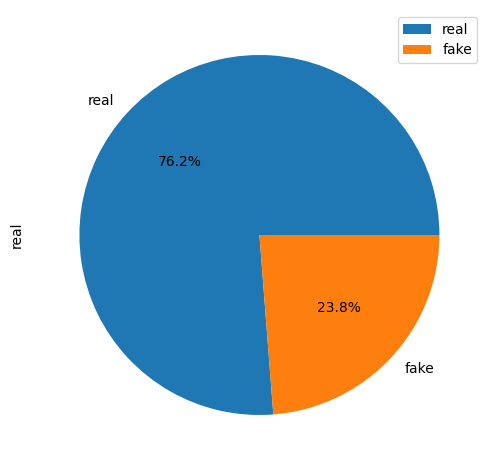
Seperti yang kita lihat, data sudah siap kita pakai.

1. Exploratory Data Analysis (EDA)

Pada tahapan ini, seorang Data Scientist melakukan analisis terhadap data yang ada dengan memvisualisasikan datanya guna mendapatkan insight dari data. Hal pertama yang saya lakukan adalah mengecek jumlah data real yang berisi palsu dan real seperti berikut.



Lalu, saya visualisasikan dengan pie plot



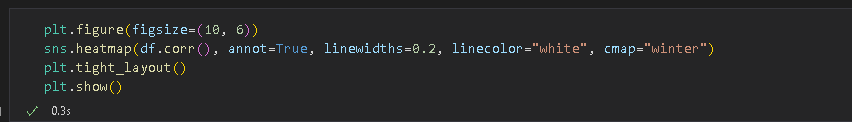
Seperti yang kita lihat, data yang kita dapatkan ternyata tidak balance. Data real berjumlah 76.2% dari keseluruhan data atau sekitar 17 ribu data sedangkan data fake berjumlah 23.8% dari keseluruhan data atau 5 ribu data.

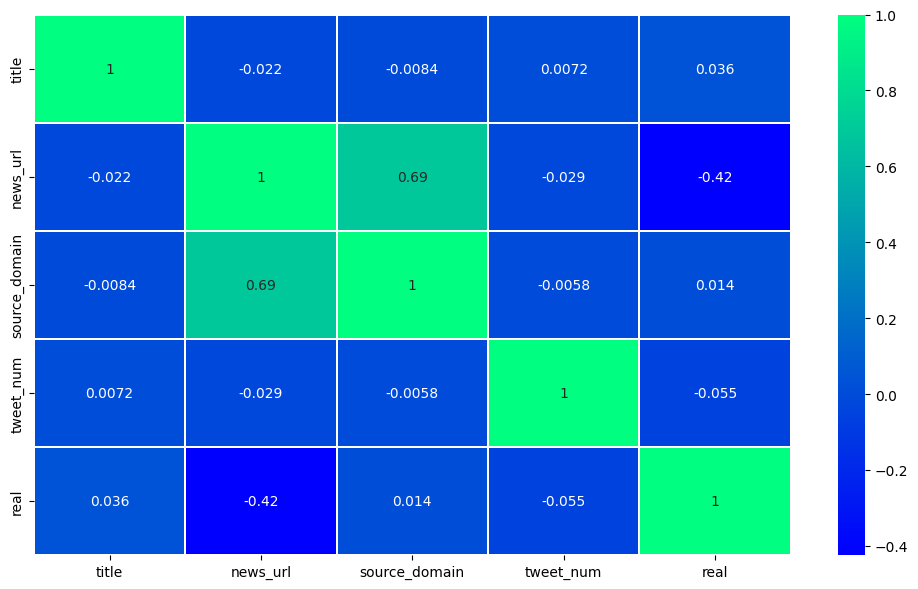
Selanjutnya, kita mengecek deskripsi dari data kategorik seperti berikut.



Seperti yang kita lihat, data title memiliki nilai unik sekitar 21 ribu dengan top atau modusnya adalah Connecting People Through News dan frekuensinya adalah 20, data news\_url memiliki nilai unik 21 ribu data dengan top atau modusnya adalah www.thewrap.com / dan frekuensinya adalah 11, dan data source\_domain memiliki nilai unik 2 ribu data dengan top atau modusnya adalah people.com dan frekuensinya 1.779.

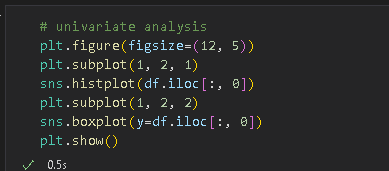
Selanjutnya, saya mengecek korelasi antar data guna mengecek keterhubungan antar data dengan kode berikut.

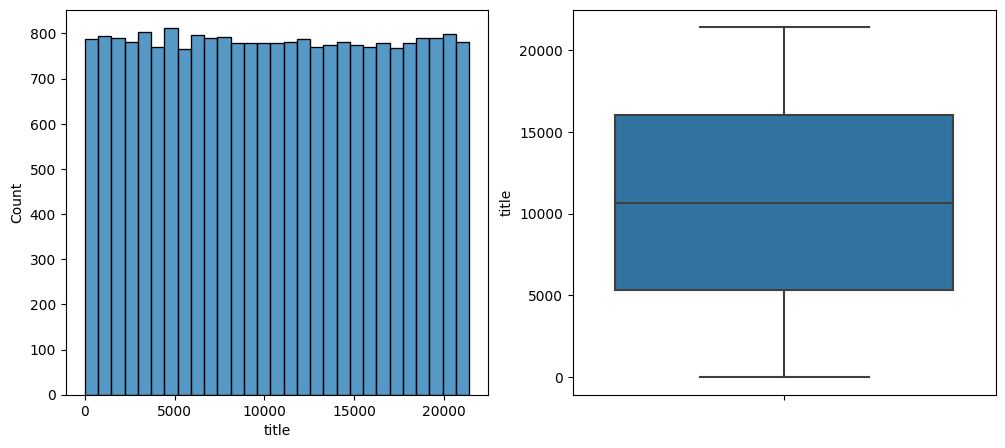




Seperti yang kita lihat, terdapat data yang berkorelasi rendah dengan data lainnya dan ada juga data yang berkorelasi tinggi dengan data lainnya seperti news\_url berkorelasi sangat baik dengan data source\_domain. Lalu, ada juga data yang berkorelasi sangat rendah seperti data title dengan data source\_domain.

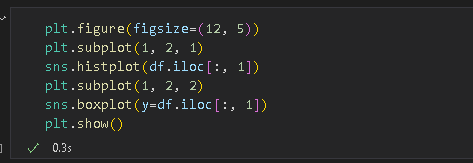
Selanjutnya, saya melakukan univariate analysis guna mencari informasi masing – masing dari data set ini. Pertama, yang saya analisis adalah data title.

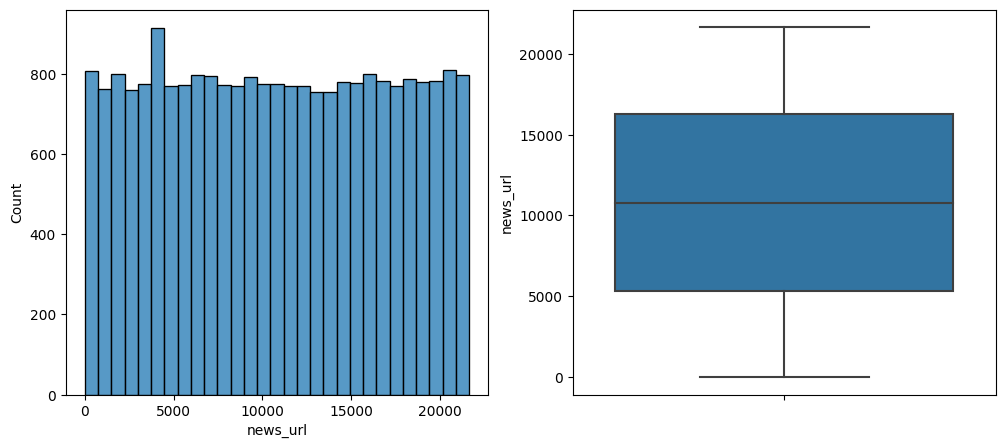




Seperti yang kita lihat, data title memiliki penyebaran yang cukup bagus dan tidak terdapat outlier disana. Ini adalah data yang aslinya objek yang telah saya encode menjadi data numerik.

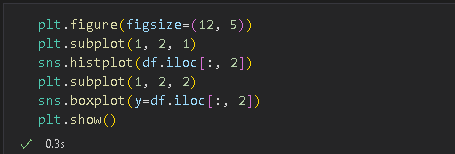
Selanjutnya, saya melakukan analisis terhadap data news\_url seperti berikut.

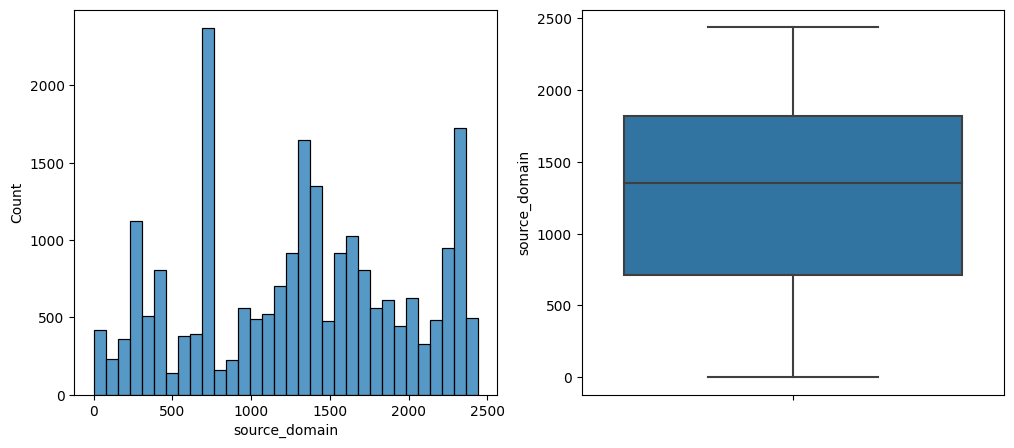




Seperti yang kita lihat, penyebaran data cukup bagus dan tidak terdapat nilai outlier. Sama seperti data sebelumnya, ini adalah data objek yang telah saya encode dengan numerik.

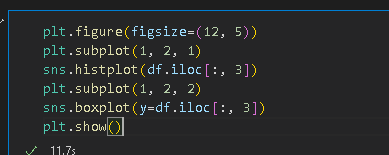
Selanjutnya, saya melakukan analisis terhadap data source domain seperti berikut.

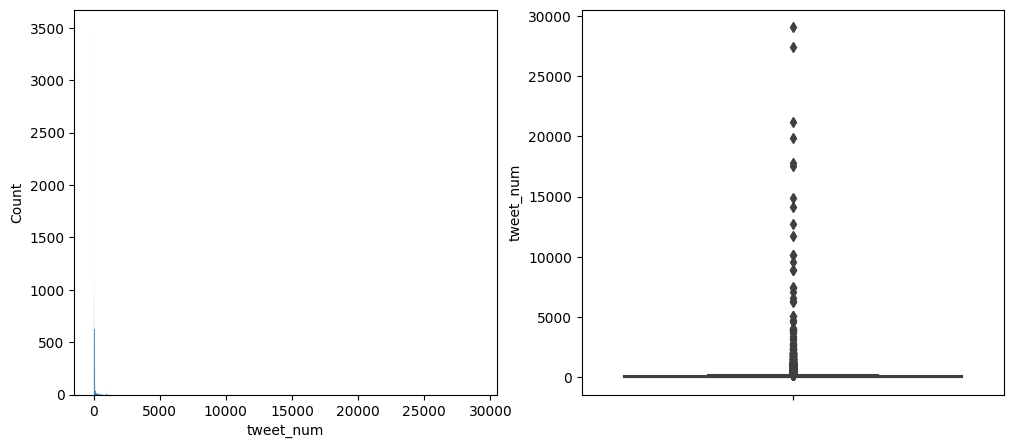




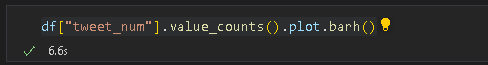
Seperti yang kita lihat, data source\_domain memiliki penyebaran yang kurang bagus karna data yang paling banyak memiliki jangkauan yang lumayan besar dengan data yang paling sedikit. Namun, disisi lain, data ini masih dapat dikatakan bagus karena tidak terdapat data outlier. Sama seperti data sebelumnya, data ini adalah data objek yang telah saya encode menjadi data numerik.

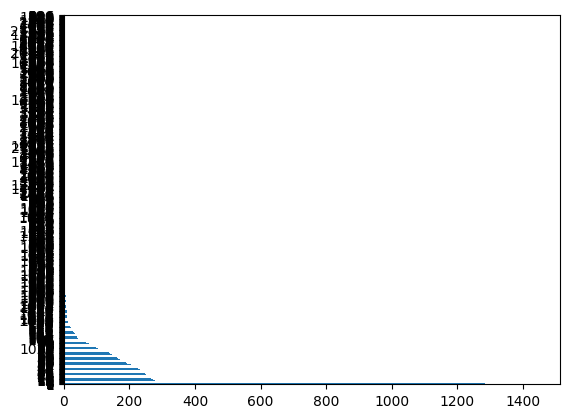
Selanjutnya, saya melakukan analisis terhadap data tweet\_num seperti berikut.





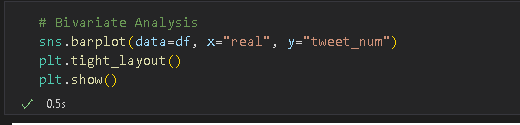
Seperti yang kita lihat, disini terdapat banyak sekali nilai outlier. Padahal data ini memang data numerik dan tidak saya encode. Namun, data ini terdapat banyak sekali outlier. Mengetahui hal ini, saya akan mengeceknya jumlah dari setiap datanya dengan value\_counts() seperti berikut.

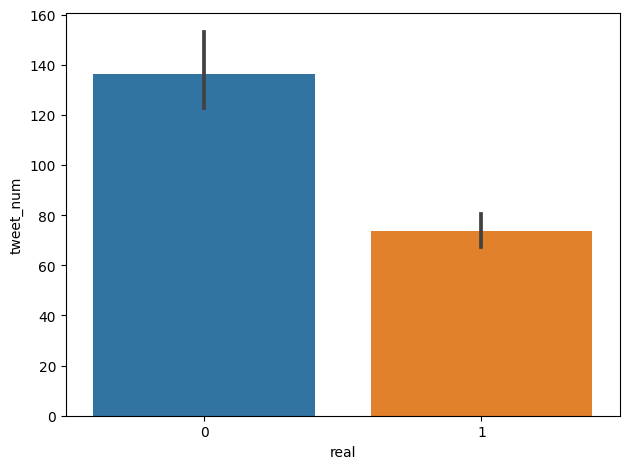




Seperti yang terlihat, ternyata penyebaran data tidak cukup bagus. Kebanyakan data terdapat pada angka sekitar 0 – 10 , selebihnya hanya terdapat 1 data saja sehingga kita dapat mengatakannya bahwa penyebaran data tidak cukup bagus dan terdapat banyaknya outlier atau noise.

Selanjutnya, kita melakukan Bivariate Analisis terhadap 2 data yaitu data real dan data tweet\_num seperti berikut.



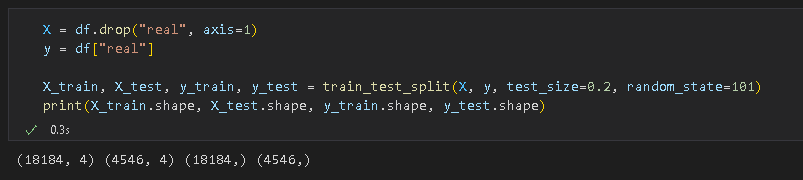


Seperti yang kita lihat, data fake memiliki nilai tweet\_num lebih besar dibanding data real.

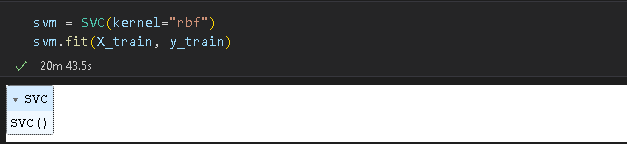
1. Model Building

Pada tahapan ini, seorang Data Scientist akan melakukan pembuatan model machine learning. Seperti requirements dari tugas ini, model yang akan saya buat adalah model Support Vector Maching, Decision Tree, dan Naïve Bayes.

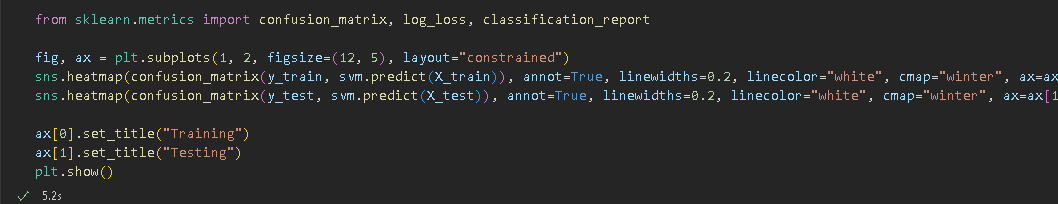
Pertama, saya membagi dataset menjadi data training dan data testing seperti berikut.

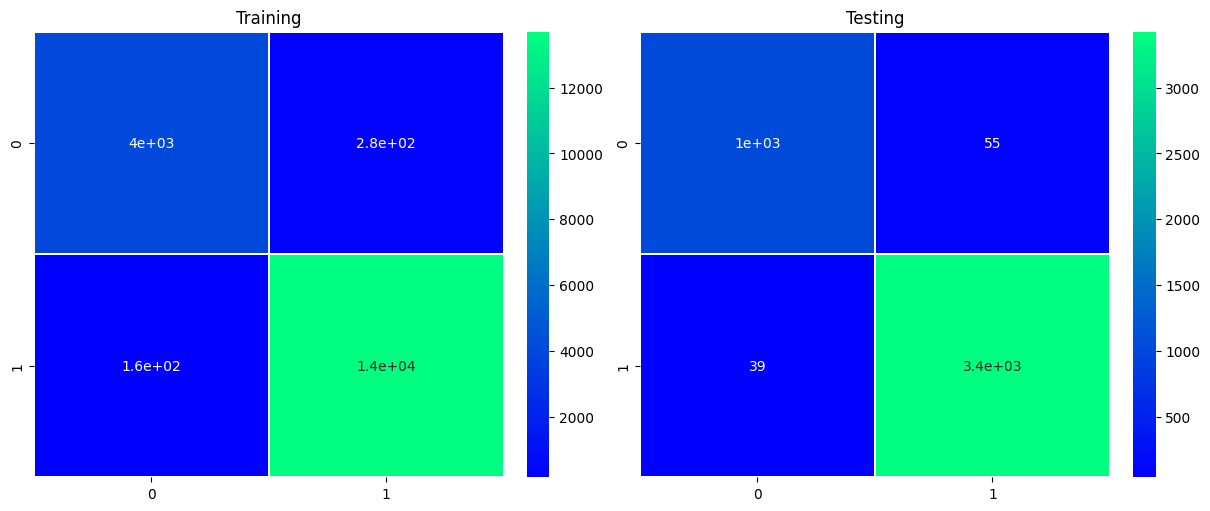


Data training yang saya dapatkan berjumlah 18 ribu data dan data testing berjumlah 4 ribu data karena saya membaginya dengan rasio 80:20. Selanjutnya, saya membuat model Support Vector Machine dan melakukan fitting terhadap data training seperti berikut.

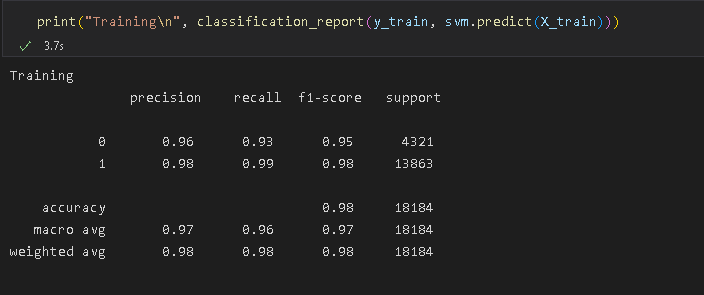


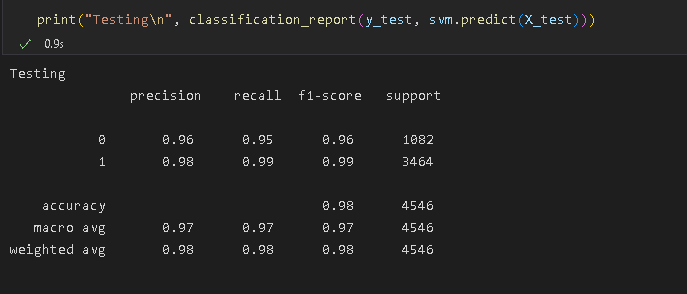
Saya membuat model SVC dengan kernel RBF dan lakukan fit terhadap data training. Lalu, saya akan lakukan mengecekan evaluasi dari model tersebut dengan confusion\_matrix dan classification\_report seperti berikut.





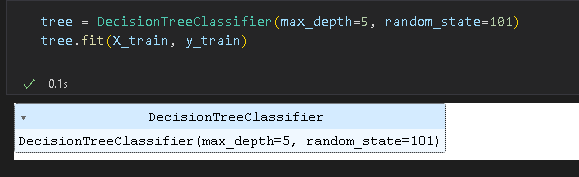
Ini adalah hasil visualisasi dari confusion\_matrix. Lalu, saya cetak presisi, recall, F-Measure, dan akurasi dari model dengan classification report.



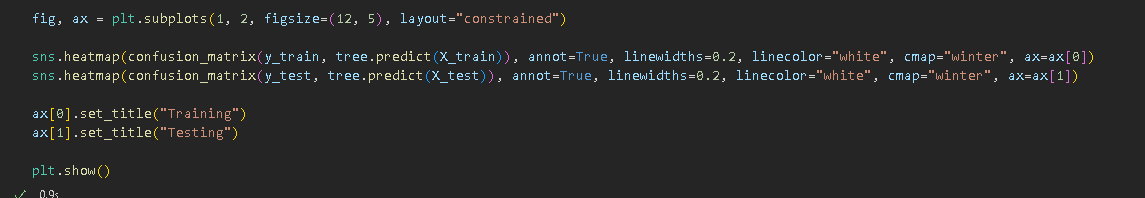


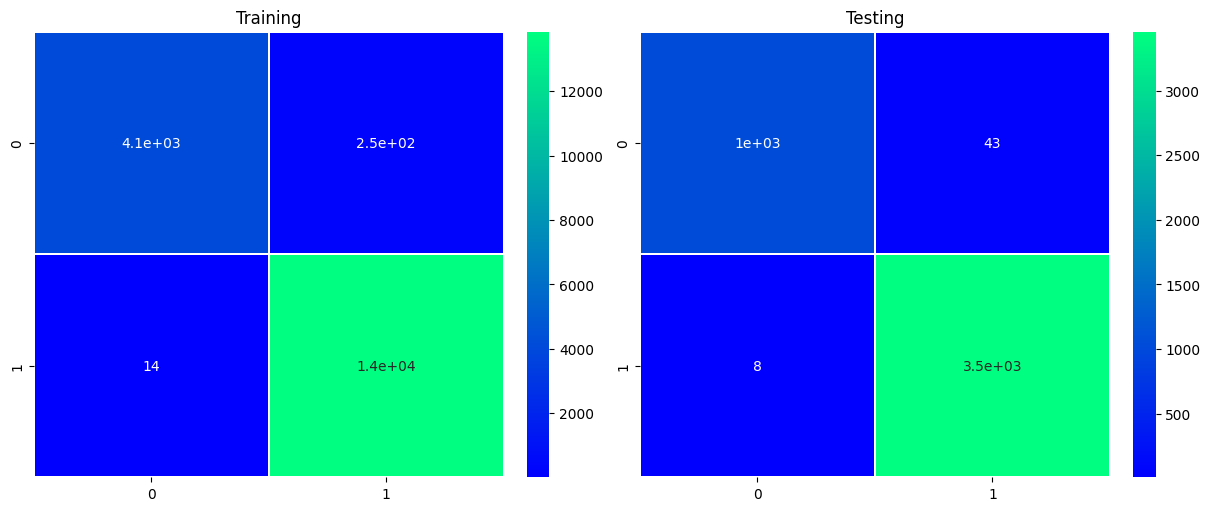
Seperti yang terlihat, data performance dari training dan testing cenderung sama dengan akurasi yang sama.

Selanjutnya, saya membuat model Decision Tree Classifier dengan max\_depth 5 dan random\_state 101. Lalu, saya fit dengan data training seperti berikut.

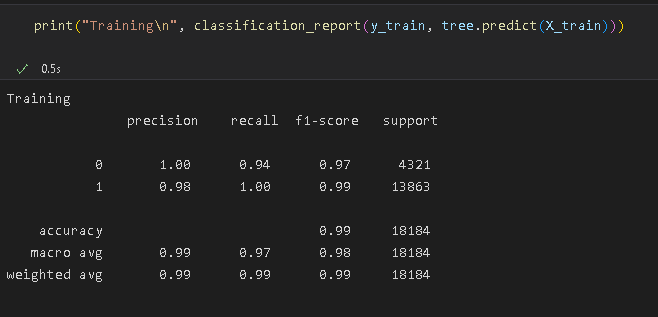


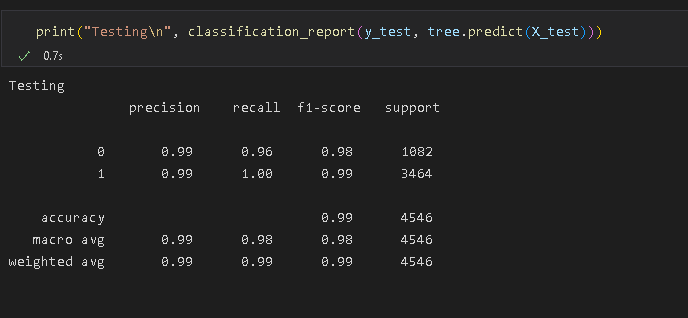
Sama seperti langkah sebelumnya, saya lakukan cek terhadap performance dari model ini dengan confusion\_matrix dan classification\_report.





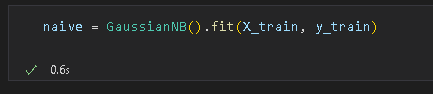
Seperti yang kita lihat, data testing ternyata cukup kecil False positive dan False negativenya dibanding pada model sebelumnya. Lalu, saya cetak presisi, recall, F-Measure, dan akurasi dari model dengan classification\_report.



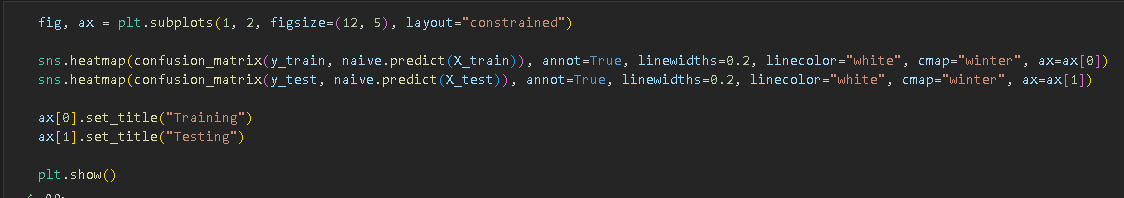


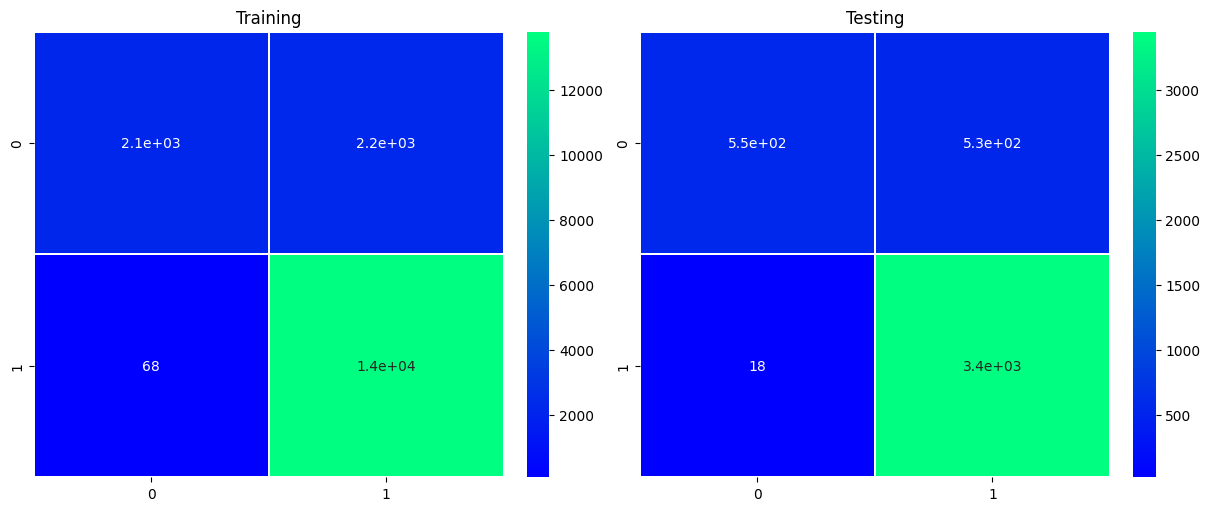
Seperti yang kita lihat, ternyata memang benar model Decision Tree memiliki presisi, recall, F-Measure dan akurasi yang lumayan baik dibanding model sebelumnya yang telah kita buat.

Selanjutnya, saya melakukan pembuatan model Naïve Bayes seperti berikut dan fit terhadap data training seperti berikut.

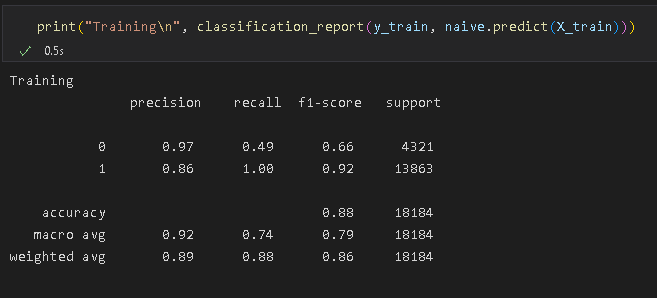


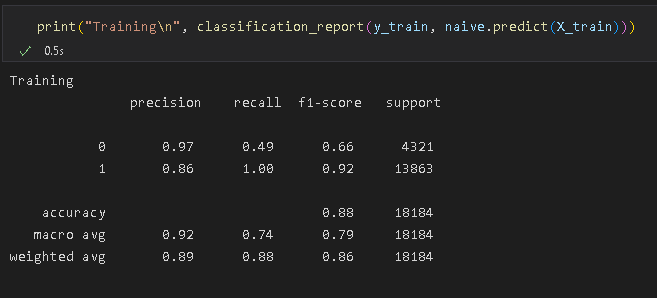
Sama seperti tahapan sebelumnya, saya melakukan evaluasi dengan confusion\_matrix dan classification\_report.





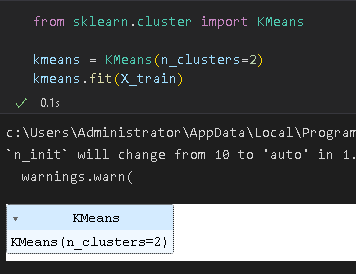
Seperti yang kita lihat, model memiliki False positive dan False Negative yang jauh lebih besar dibanding model sebelumnya.



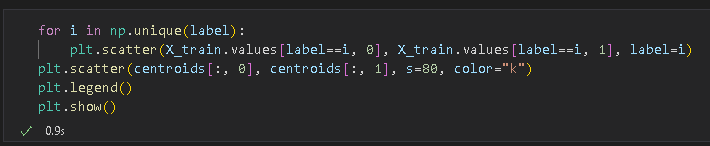


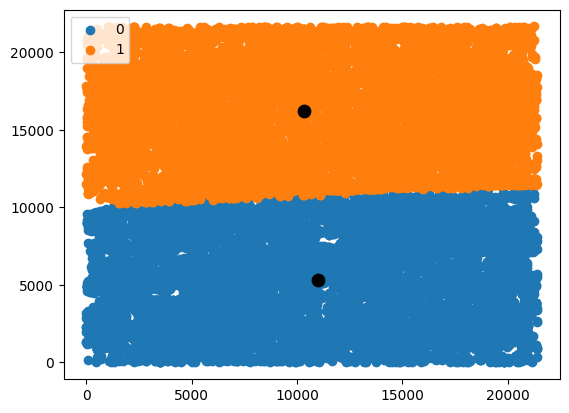
Seperti yang kita lihat, ternyata memang benar bahwa model ini memiliki performance yang lebih buruk dari model sebelumnya.

Selanjutnya, saya membuat model clustering dengan KMeans dengan cluster 2 seperti berikut.

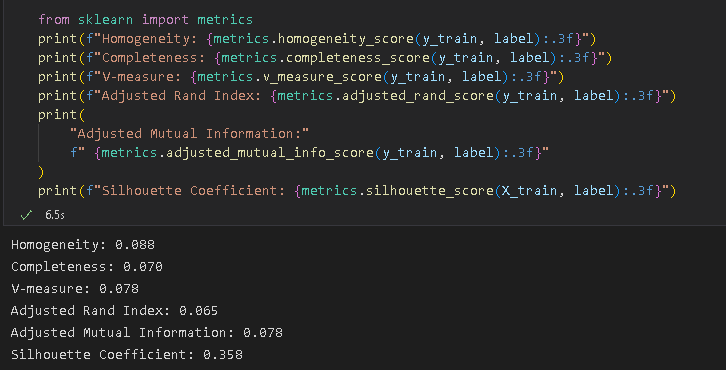


Lalu, saya visualisasikan hasil clusteringnya seperti berikut.



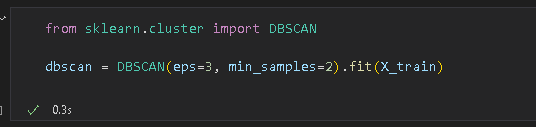


Seperti yang kita lihat, ternyata model memprediksi bahwa data real lebih besar dibanding data fake sehingga data real berada diatas sedangkan data fake berada dibawah. Lalu, saya mengecek performance dari model ini seperti berikut.

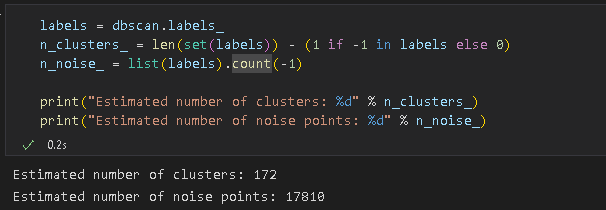


Ternyata model bisa dikatakan kurang baik melihat silhouettenya hanya 35%.

Selanjutnya, saya membuat model DBSCAN (Density Based Spatial Clustering of Application with Noise) seperti berikut.

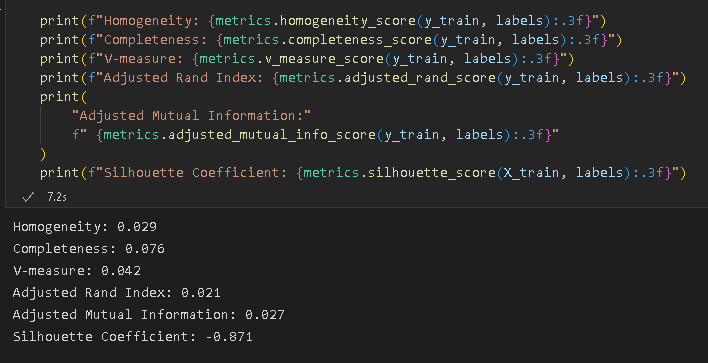


Lalu, saya cek estimasi dari cluster yang telah terbuat dan data noise yang terprediksi seperti berikut.



Seperti yang terlihat, cluster yang terbentuk cukup besar yaitu 172 cluster dan noise yang sangat besar yaitu 17 ribu data.

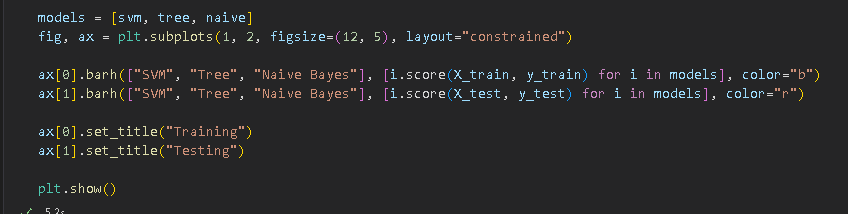
Lalu, Saya mengecek performance dari model ini seperti berikut.

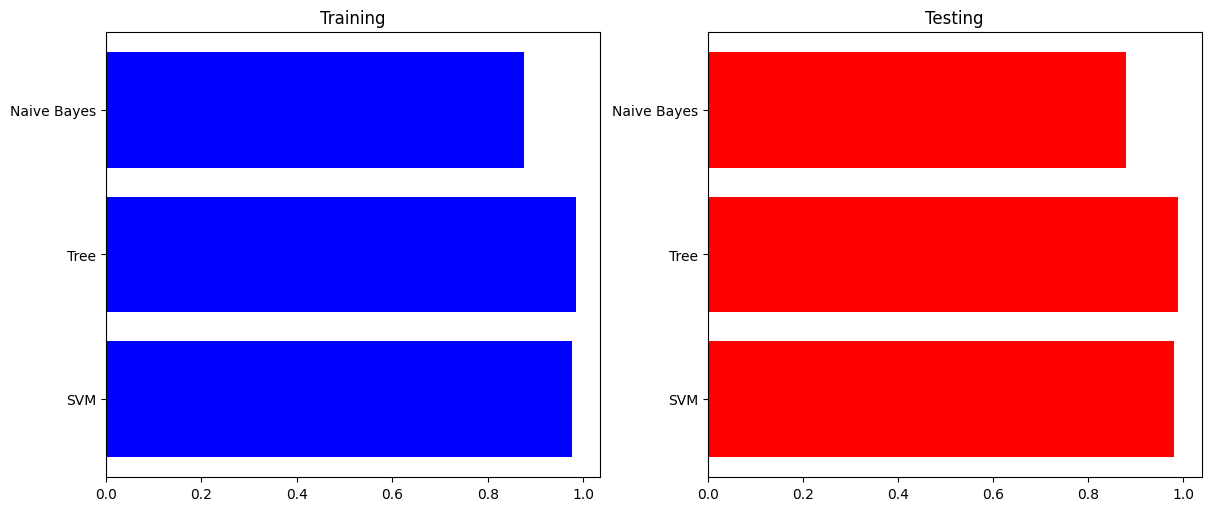


Model memiliki performance yang lumayan dibanding model sebelumnya.

1. Model Evaluation

Pada tahapan sebelumnya kita telah mengecek evaluasi dari masing – masing model, selanjutnya adalah tahap akhir dari evaluasi model. Kita bisa membandingkan model – model tersebut dengan memvisualiasikannya seperti berikut.





Seperti yang kita lihat, model terbaik sejauh ini yang bisa gunakan adalah Decision Tree dan Support Vector Machine. Dengan dimensi data yang berjumlah hanya 5, dapat disarankan untuk memakai Decision Tree. Decision Tree merupakan model yang menarik untuk dipelajari karena pohon dari decision tree hanya berisi if dan else yang dapat kita pahami dengan mudah sebab dari model ini seperti berikut.

