

Prediksi Kepribadian Mahasiswa Menggunakan Naïve Bayes

Muhammat Rasid Ridho^{a,*}, Sasa Ani Arnomo^b, Fifi^c, Khisal^d, Vina Fariska^e

^{abcde}Program Studi Sistem Informasi, Universitas Putera Batam, Batam

*rasid@puterabatam.ac.id, sasa@puterabatam.ac.id, fifi@puterabatam.ac.id

Abstract

College students are in a transitional phase from youth to adulthood. The transition period makes students still unstable to control their emotions. It makes his curiosity towards new things increase which then shows his personality traits. The purpose of this study was to find out how researchers collect data about personality from students, to find out how to classify personality from the data that has been collected. Research methods start from collecting data using Text Preprocessing questionnaires, Data Training, Classification, Testing, to making predictions. After applying the classification algorithm with the Naïve Bayes algorithm, the Train Score is 0.947 and the Test Score is 0.879. Trials have also been carried out to make predictions with new data whose results are correct.

Keywords: Personality; predictions; Naïve Bayes

Abstrak

Mahasiswa berada pada fase transisi dari remaja ke dewasa. Masa transisi membuat mahasiswa masih labil untuk mengendalikan emosinya. Hal itu membuat rasa ingin tahunya terhadap hal-hal baru semakin meningkat yang kemudian menunjukkan ciri-ciri kepribadiannya. Adapun tujuan penelitian ini adalah untuk mengetahui cara peneliti melakukan pengumpulan data tentang kepribadian dari mahasiswa, untuk mengetahui cara mengklasifikasikan kepribadian dari data yang sudah terkumpul. Metode penelitian mulai dari mengumpulkan data menggunakan menggunakan kuisioner Text Preprocessing, Data Training, Klasifikasi, Testing, sampai melakukan prediksi. Setelah di terapkan algoritma klasifikasi dengan algoritma Naïve Bayes menghasilkan Train Score nya 0,947 dan Test Score nya 0,879. Sudah dilakukan uji coba juga untuk dilakukan prediksi dengan data baru yang hasilnya sudah tepat.

Kata Kunci: Kepribadian; Prediksi; Naïve Bayes

1. Pendahuluan

Kepribadian adalah suatu sifat yang tercermin dan sudah dimiliki dalam diri seseorang sehingga dapat membedakannya dengan orang lain. Mahasiswa berada pada fase transisi dari remaja ke dewasa. Masa transisi membuat mahasiswa masih labil untuk mengendalikan emosinya. Hal itu membuat rasa ingin tahunya terhadap hal-hal baru semakin meningkat yang kemudian menunjukkan ciri-ciri kepribadiannya. Masalah dapat muncul dari perubahan kepribadian. Dalam lingkungan pendidikan hal ini perlu ditangani oleh bagian kemahasiswaan atau bagian Dosen Pembimbing Akademik (Alam: 2020).

Berdasarkan buku Personality Plus, tipe kepribadian manusia dapat dibagi menjadi 4: sanguinis, kolerik, melankolis, dan apatis. Seseorang dengan tipe kepribadian sanguinis lebih mengutamakan perasaan daripada pikiran. Selain itu, tipe kepribadian sanguinis juga bergairah terhadap semua orang yang ditemuinya. Kelemahan tipe kepribadian ini

tidak menyukai kesendirian, sangat tidak menyukai kesedihan, dan cenderung sering mencari cara untuk tetap bahagia. Tipe kepribadian choleric memiliki semangat yang tinggi dan selalu optimis. Namun kelemahannya adalah keras kepala, mudah tersinggung, suka mendominasi, tidak sabaran, suka keributan yang berujung pertengkaran. Tipe kepribadian melankolis sering kali merasa khawatir atau takut dan termasuk orang yang mudah menyerah. Di balik itu, seorang melankolis adalah seseorang yang sangat kreatif dan analitis. Sedangkan tipe phlegmatis tidak menyukai kekerasan dan selalu menyukai perdamaian. Seorang phlegmatis cenderung menghindari kegiatan yang berpotensi menimbulkan konflik. Plegmatis juga lebih menyukai ketenangan (Thandan: 2022).

Semua manusia memiliki ciri-ciri kecerdasan, terdapat indikator pada setiap kecerdasan yang bekerja sama lintas semua budaya mulai dari kemampuan pola dasar yang diekspresikan melalui hobi dan kemampuan yang memerlukan bantuan khusus.

Metode Naive Bayes adalah metode untuk melakukan klasifikasi yang menggunakan probabilitas dan statistik dan dipresentasikan oleh seorang ilmuwan Inggris bernama Thomas Bayes. Berdasarkan pengalaman masa lalu, algoritma Naive Bayes memprediksi masa depan. Adapun tujuan penelitian ini adalah: Untuk mengetahui cara peneliti melakukan pengumpulan data tentang kepribadian dari mahasiswa. Serta untuk mengetahui cara mengklasifikasikan kepribadian dari data yang sudah terkumpul.

2. Kajian Literatur (Arial, 10, Bold)

Saat ini tes kepribadian otomatis telah dilakukan pada platform media sosial tertentu untuk prediksi sifat manusia. Penelitian saat ini mencoba kepada mahasiswa. Analisis kepribadian manusia, yang terus berupaya menganalisis perilaku pengguna di media sosial, adalah salah satu opsi yang tersedia untuk jenis data ini. Untuk tujuan ini, metode pembelajaran mesin yang berbeda dapat digunakan untuk memprediksi ciri-ciri kepribadian manusia yang berada dalam kelompok media sosial. Dengan bantuan Model Lima Besar, kegiatan yang diusulkan ini berfokus untuk memprediksi ciri-ciri kepribadian pesan sosial media. Dengan melakukan ini, model pemisahan dengan beberapa label ringan akan dikembangkan dan diuji. Meskipun ada pelajaran untuk memprediksi kepribadian dari Twitter Salah satu aspek baru dari inisiatif yang diusulkan ini adalah bahwa analisis manusia dilakukan pada kelompok sosial media daripada profil pengguna individual.

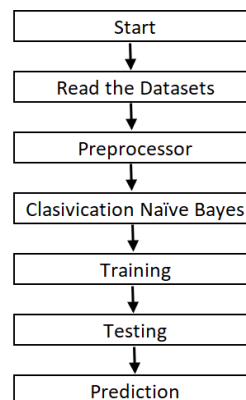
Proyek saat ini berfokus pada memprediksi ciri-ciri kepribadian yang menonjol dalam kumpulan tweet untuk masalah tertentu dan mengelompokkannya ke dalam kategori berikut.

1. Ekstroversi
2. Neuroticism
3. Agreeableness
4. Conscientiousness
5. Keterbukaan terhadap Pengalaman

Penilaian kepribadian seseorang ini memiliki dampak yang signifikan terhadap karir dan kehidupan pribadinya. Banyak perusahaan telah mulai mencuitkan pelamar berdasarkan kepribadian mereka untuk meningkatkan efisiensi kerja karena orang tersebut mengerjakan sesuatu yang dia sukai daripada sesuatu yang terpaksa dia lakukan (Liu: 2016).

3. Metode Penelitian

Berikut ini merupakan kerangka metode yang diusulkan seperti digambarkan di bawah ini:



Kerangka Metode yang di usulkan (Krisdianto: 2021). Penelitian ini akan dimulai dari pengumpulan data dengan CrawlTangle yang akan di jadikan raw material. Kemudian preprocessing ada beberapa langkah diantaranya case folding, tkenizing, filtering, cleansing, dan stemming. Tahap berikutnya melkukan klasifikasi data menggunakan Naïve Bayes, dan terakhir dilakukan visualisasi data.

Pengumpulan Dataset

Rencana penelitian ini akan mengambil menggunakan kuisisioner kepada mahasisiwa di Batam. Metode pengumpulan datanya menggunakan penyeberan kuisisioner. Adapun yang mengisi dari berbagai macam program studi.

Preprocessing

Data yang tak terstruktur akan dikumpulkan dan dilakukan taham preprocessing. Dimana merupakan tahapan yang lumayan penting untuk dilakukan. Tahapan ini dilakukan beberapa langkah yang dilaksanakan urut, penjelasannya di bawah ini:

1. Tahap Case Folding:

Tahap Case Folding adalah sebuah tahap agar mengubah semua huruf kedalam ulasan jadi huruf bukan kapital secara semuanya atau sering dikenal lowercase .

2. Tahap Tokenizing:

Tahap Tokenizing adalah sebuah tahapan membagi inputan yang telah dilaksanakan disetiap kata kedalam penyusunan.

3. Tahap Filtering:

Tahap Filtering merupakan tahapan menghapus kata yang tak punya arti.

4. Tahap Cleansing:

Tahap Cleansing adalah sebuah tahapan untuk menghilangkan tiap karakter kedalam ulasan tak termasuk dalam alfabet, hingga dapat menghapus karakter yang tak diharapkan atau tak punya makna. Sebagai contoh seperti angka, *, \$, +, maupun tautan dari sebuah web yang ada didalam suatu ulasan.

5. Tahap Stemming:

Tahap Stemming adalah tahapan mengubah tiap kata dimana ada tambahan menjadi sebuah kata dasar.

Naïve Bayes

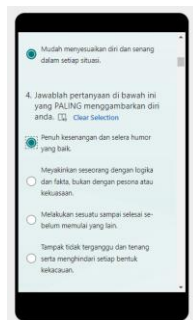
Klasifikasi Naïve Bayes dibuat mengikuti data train dalam pembelajaran yang sudah di siapkan di algoritma. Kegunaan dari algoritma yakni dimanfaatkan agar melihat estimasi kemungkinan yang ada mengikuti dari pilihan data train. Klasifikasi Naïve Bayes dikerjakan dengan cara mejadikan satu tambahan pengetahuan yang sudah ada ke yang baru.

Visualisasi

Langkah tahapan visualisasi tersebut yakni membuat tampilan output klasifikasi menggunakan klasifikasi Naïve Bayes mengikuti pembagian emosi ke polaritas setiap kata.

4. Hasil dan Pembahasan

Pengumpulan data menggunakan penyebaran kuisioner kepada para mahasiswa. Total jumlah pertanyaanya sebanyak 40. Dimana pertanyaan tersebut menggambarkan 4 kategori kepribadian dari mahasiswa yaitu 1.Sanguinis, 2. Koleris, 3. Melankolis dan 4. Plegmatis.



Kuisioner yang disebarkan kepada mahasiswa. Setelah data dikumpulkan, dilakukan pembersihan data. Seperti ada jawaban yang kosong, kita hapus. Data yang pengisiannya keliru atau bentuk kejanggalan yang lain, agar nanti hasil dari pengolahan data berikutnya lebih bagus.

Tabel Dataset yang akan diolah.

Nama	1	2	3	4	5	6	7	8	9	...	32	33	34	35	36	37	38	39	40	kepribadian	
0	1	B	D	B	D	B	D	D	D	...	B	B	B	A	B	C	D	B	C	2	
1	2	D	C	B	C	C	C	C	B	C	...	B	B	C	D	B	B	A	C	B	3
2	3	D	A	D	C	B	D	D	B	B	...	A	A	A	A	B	A	A	A	1	
3	4	C	A	D	C	A	B	D	B	D	...	C	A	C	C	B	B	D	C	B	1
4	5	B	D	A	D	D	C	C	D	C	...	C	B	C	C	C	C	D	A	3	
...
160	161	B	A	D	C	C	B	A	D	D	...	C	B	C	D	A	D	D	A	D	1
161	162	B	A	C	C	C	D	C	D	D	...	D	D	C	C	C	C	D	B	4	
162	163	A	C	D	C	D	C	D	B	...	D	D	C	C	C	C	B	C	C	3	
163	164	C	A	D	A	A	D	C	D	D	...	B	B	A	D	D	D	D	D	C	4
164	165	A	A	A	C	C	B	C	C	D	...	B	B	C	A	C	B	B	B	A	3

165 rows x 42 columns

Sebelum dilakukan train data dan testing peneliti melakukan pebagian data terdulu. Mana data yang akan di jadikan data trainng dan mana data yang akan dijadikan data testing.

```
X = df.drop(columns="kepribadian")
y = df.kepribadian

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=5)
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

Adapun penjelasannya sebagai berikut:

1. `X = df.drop(columns="kepribadian")`: Di sini, kita menggunakan DataFrame `df` dan menghapus kolom "kepribadian" menggunakan metode `drop()`. Variabel `X` akan berisi DataFrame yang terdiri dari fitur-fitur yang digunakan untuk melatih model.
2. `y = df.kepribadian`: Di sini, kita mengambil kolom "kepribadian" dari DataFrame `df` dan menyimpannya dalam variabel `y`. Variabel `y` akan berisi target atau label yang sesuai dengan fitur-fitur dalam `X`. Dalam konteks ini, kita mengasumsikan bahwa "kepribadian" adalah kolom target yang ingin diprediksi.
3. `train_test_split(X, y, test_size=0.2, random_state=5)`: Fungsi `train_test_split` digunakan untuk membagi dataset menjadi subset pelatihan (training set) dan subset pengujian (test set). Argumen pertama adalah `X`, yang merupakan DataFrame fitur, dan argumen kedua adalah `y`, yang merupakan Series target. Argumen `test_size=0.2` menunjukkan bahwa 20% data akan digunakan untuk pengujian. `random_state=5` digunakan untuk mengontrol randomization sehingga hasil pembagian dataset dapat direproduksi dengan konsisten.
4. `X_train, X_test, y_train, y_test`: Hasil dari pemanggilan `train_test_split` disimpan dalam empat variabel terpisah. `X_train` berisi subset pelatihan fitur, `X_test` berisi subset pengujian fitur, `y_train` berisi subset pelatihan target, dan `y_test` berisi subset pengujian target.
5. `X_train.shape, X_test.shape, y_train.shape, y_test.shape`: Kode ini digunakan untuk menampilkan bentuk (shape) dari setiap subset. `X_train.shape` memberikan jumlah baris dan kolom dalam `X_train`, `X_test.shape` memberikan jumlah baris dan kolom dalam `X_test`, `y_train.shape` memberikan jumlah elemen dalam `y_train`, dan `y_test.shape` memberikan jumlah elemen dalam `y_test`.

Dengan melihat hasil bentuk (shape) subset pelatihan dan pengujian, kita dapat memastikan bahwa pembagian dataset telah dilakukan dengan benar sesuai proporsi yang ditentukan. Subset pelatihan (`X_train` dan `y_train`) akan digunakan untuk melatih model, sementara subset pengujian (`X_test` dan

Split Data untuk Train dan Test

`y_test`) akan digunakan untuk menguji kinerja model.

Preprocessor

Preprocessing (pre-pemrosesan) dalam konteks data merupakan serangkaian langkah atau teknik yang diterapkan pada data mentah sebelum data tersebut digunakan untuk membangun model atau melakukan analisis lebih lanjut. Tujuannya adalah untuk mengubah data menjadi format yang lebih cocok, memperbaiki masalah yang ada, dan mempersiapkannya agar dapat memberikan hasil yang lebih baik saat dianalisis atau digunakan dalam model machine learning. Adapun kode yang di eksekusi sebagai berikut:

```
preprocessor = ColumnTransformer([
    ('numeric', num_pipe(), ["A", "B", "C", "D"])
])
```

Dalam kode yang diberikan, `preprocessor` adalah sebuah objek dari kelas `ColumnTransformer`. Ini adalah bagian dari pipeline pemrosesan data yang digunakan untuk mengubah data mentah sebelum digunakan untuk membangun model atau melakukan analisis lebih lanjut. Penjelasan mengenai `preprocessor`:

1. `ColumnTransformer` adalah suatu objek yang memungkinkan untuk menerapkan serangkaian transformer pada kolom-kolom tertentu dalam dataset secara terpisah. Hal ini memungkinkan untuk melakukan pemrosesan yang berbeda pada kolom-kolom dengan tipe data yang berbeda, seperti data numerik dan kategorikal.
2. `(('numeric', num_pipe(), ["A", "B", "C", "D"]))` adalah sebuah tuple yang memberikan spesifikasi untuk transformer yang akan digunakan pada kolom numerik. Di sini:
3. `numeric` adalah nama atau label yang diberikan kepada transformer ini. Dapat memberikan nama yang sesuai dengan preferensi kita.
4. `num_pipe()` adalah fungsi yang digunakan untuk membangun pipeline pemrosesan data pada kolom numerik.
5. `["A", "B", "C", "D"]` adalah daftar nama kolom numerik yang ingin diproses menggunakan `num_pipe()`. Dalam contoh ini, kolom numerik yang akan diproses adalah "A", "B", "C", dan "D".

Dengan menggunakan `ColumnTransformer` seperti yang ditunjukkan dalam kode di atas, Anda dapat menggabungkan transformer yang berbeda untuk kolom numerik dan kategorikal dalam satu objek preprocessor. Ketika menjalankan pipeline pemrosesan data dengan menggunakan `preprocessor.transform(df)`, `ColumnTransformer` akan secara otomatis menerapkan transformer yang sesuai pada

kolom-kolom numerik yang ditentukan ("A", "B", "C", dan "D") dan menjaga kolom-kolom kategorikal tetap tidak berubah.

Misalnya, jika memiliki dataframe `df` yang berisi kolom-kolom "A", "B", "C", "D" sebagai kolom numerik bersama dengan kolom-kolom lainnya, dapat menerapkan pipeline pemrosesan data pada kolom numerik tersebut dengan menggunakan `preprocessor.transform(df)`. Hasilnya akan menjadi dataframe yang telah diproses dengan langkah-langkah pemrosesan data numerik yang telah ditentukan.

4.3 Naïve Bayes.

Klasifikasi Naïve Bayes dibuat mengikuti data train dalam pembelajaran yang sudah di siapkan di algoritma. Kegunaan dari algoritma yakni dimanfaatkan agar melihat estimasi kemungkinan yang ada mengikuti dari pilihan data train. Klasifikasi Naïve Bayes dikerjakan dengan cara mejadikan satu tambahan pengetahuan yang sudah ada ke yang baru.

```
from sklearn.naive_bayes import GaussianNB
pipeline = Pipeline([
    ('prep', preprocessor),
    ('algo', GaussianNB())
])
```

Dalam kode yang berikan, terdapat penggunaan `Pipeline` dari scikit-learn dan integrasinya dengan `preprocessor` serta model Naive Bayes Gaussian. Berikut ini penjelasan langkah-langkahnya:

1. `from sklearn.naive_bayes import GaussianNB`: Pada baris ini, kita mengimpor kelas `GaussianNB` dari modul `naive_bayes` dalam library scikit-learn. `GaussianNB` adalah implementasi Naive Bayes dengan asumsi bahwa fitur-fitur numerik diikuti oleh distribusi Gaussian.
2. `pipeline = Pipeline([('prep', preprocessor), ('algo', GaussianNB())]`: Di sini, kita mendefinisikan sebuah objek `Pipeline`. `Pipeline` digunakan untuk menggabungkan berbagai tahapan pemrosesan data dan model ke dalam satu aliran kerja yang terstruktur.
 - a) `('prep', preprocessor)`: Ini adalah tahapan pertama dalam pipeline dengan nama `prep`. `preprocessor` yang telah didefinisikan sebelumnya akan digunakan di tahap ini untuk melakukan pemrosesan data pada fitur-fitur.
 - b) `('algo', GaussianNB())`: Ini adalah tahapan kedua dalam pipeline dengan nama `algo`. Di sini, kita menggunakan model `GaussianNB` sebagai algoritme yang akan digunakan untuk melatih dan melakukan prediksi pada data yang telah diproses sebelumnya

Dengan menggunakan `Pipeline` seperti yang ditunjukkan dalam kode di atas, dapat memasukkan beberapa tahapan pemrosesan

data (dalam hal ini, `preprocessor`) dan model (dalam hal ini, `GaussianNB()`) dalam satu aliran kerja yang teratur. Ini mempermudah dalam menjalankan pipeline secara keseluruhan dengan memanggil metode seperti `fit()` untuk melatih model dan `predict()` untuk melakukan prediksi.

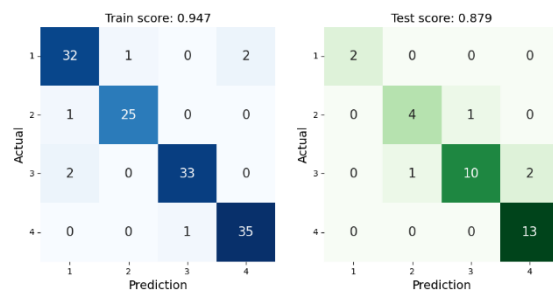
Misalnya, dapat melatih model Naive Bayes Gaussian pada data yang telah melalui pemrosesan menggunakan `pipeline.fit(X_train, y_train)` dan kemudian melakukan prediksi pada data pengujian menggunakan `pipeline.predict(X_test)`. Hal ini memungkinkan untuk mengintegrasikan pemrosesan data dan model ke dalam satu aliran kerja yang lebih efisien dan mudah digunakan.

Train dan Test

Instruksi yang digunakan adalah sebagai berikut:

```
pipeline.fit(X_train, y_train)
pipeline.score(X_train, y_train)
pipeline.score(X_test, y_test)
```

Perintah `pipeline.fit(X_train, y_train)` merupakan metode `fit()` dari objek pipeline digunakan untuk melatih model pada data pelatihan. Data pelatihan terdiri dari fitur-fitur (`X_train`) dan target (`y_train`). Dalam konteks ini, pipeline akan menjalankan langkah-langkah pemrosesan data yang telah ditentukan dalam `preprocessor` terlebih dahulu pada `X_train`. Kemudian, model Naive Bayes Gaussian yang ditentukan dalam tahap 'algo' akan dilatih menggunakan data yang telah diproses. Ini berarti pipeline akan menggabungkan pemrosesan data (`preprocessing`) dan pembuatan model dalam satu aliran kerja yang terstruktur. Setelah menjalankan metode `fit()`, model akan belajar dari data pelatihan dan siap digunakan untuk melakukan prediksi pada data baru. Adapun `train score` nya 0.0,947 dan `test score` nya 0.879. Untuk melihat secara visual bisa dilihat gambar dibawah ini.



Gambar Hasil Train dan Test

Prediksi

Prediksi digunakan untuk menghasilkan nilai atau kelas yang diprediksi oleh model

berdasarkan input yang diberikan. Dalam konteks pemodelan dan pembelajaran mesin (machine learning), prediksi adalah proses menggunakan model yang telah dilatih untuk memperkirakan atau memprediksi output yang diharapkan berdasarkan input yang diberikan. Tujuan utama dari prediksi ini untuk mengambil data baru yang tidak diketahui dan menggunakannya untuk menghasilkan hasil yang diharapkan berdasarkan model yang telah dilatih sebelumnya. Dengan memprediksi nilai atau kelas yang diharapkan, kita dapat mengambil keputusan atau mengambil tindakan berdasarkan hasil prediksi tersebut.

```
X_pred = pd.read_csv("DataKepribadian7Tes.csv")
X_pred

   A  B  C  D
0  3  16  6  15

pipeline.predict(X_pred)

array([2], dtype=int64)

X_pred["kepribadian"] = pipeline.predict(X_pred)
X_pred

   A  B  C  D  kepribadian
0  3  16  6  15  2
```

Keterangan:

1. `X_pred = pd.read_csv("data/testing.csv")`: Pada baris ini, kita menggunakan fungsi `read_csv()` dari library `pandas` (`pd`) untuk membaca file CSV yang berisi data pengujian. File CSV tersebut diberikan dalam parameter "DataKepribadian.csv". Perhatikan bahwa path file dapat berbeda tergantung pada struktur direktori Anda. Data yang dibaca dari file CSV akan disimpan dalam variabel `X_pred`. Setelah menjalankan kode di atas, `X_pred` akan berisi data pengujian yang telah dibaca dari file CSV. Anda dapat menggunakan `X_pred` untuk melakukan prediksi dengan model yang telah dilatih sebelumnya menggunakan `pipeline.predict(X_pred)`.
2. Dalam kode yang Anda berikan, `pipeline.predict(X_pred)` digunakan untuk melakukan prediksi menggunakan model yang telah dilatih pada data pengujian `X_pred`. `pipeline.predict(X_pred)`: Di sini, kita menggunakan metode `predict()` dari objek pipeline untuk melakukan prediksi pada data pengujian `X_pred`. Metode `predict()` akan menerapkan langkah-langkah pemrosesan data yang telah ditentukan dalam `preprocessor` terlebih dahulu pada `X_pred`, kemudian menggunakan model yang telah dilatih untuk menghasilkan prediksi pada data tersebut. Hasil prediksi akan dikembalikan sebagai array atau objek `Seri`, yang berisi prediksi kelas atau nilai yang dihasilkan oleh model untuk setiap sampel dalam `X_pred`. Dengan menjalankan `pipeline.predict(X_pred)`, Anda dapat memperoleh hasil prediksi dari model pada

data pengujian `X_pred`. Anda dapat menggunakan hasil prediksi ini untuk evaluasi lebih lanjut atau keperluan lain yang sesuai dengan tujuan analisis Anda.

3. Hasil yang ditampilkan, yaitu `array([0], dtype=int64)`, merupakan contoh hasil prediksi yang diberikan oleh model pada satu sampel data pengujian. Nilai 0 dalam array tersebut menunjukkan prediksi kelas atau nilai yang dihasilkan oleh model untuk sampel tersebut.

4. Perlu dicatat bahwa jenis nilai yang ditampilkan dalam `dtype` (`int64` dalam contoh ini) tergantung pada tipe data target yang digunakan dalam pemodelan. Dalam hal ini, tipe data target yang digunakan adalah tipe data integer dengan ukuran 64-bit. Jika Anda memiliki lebih dari satu sampel data pengujian, maka hasil prediksi akan ditampilkan sebagai array dengan ukuran yang sesuai dengan jumlah sampel tersebut. Misalnya, jika Anda memiliki 10 sampel data pengujian, hasil prediksi bisa terlihat seperti `array([0, 1, 0, 1, 0, 1, 1, 0, 0, 1], dtype=int64)`, di mana masing-masing nilai dalam array menunjukkan prediksi untuk setiap sampel data.

5. Dalam kode yang `X_pred["kepribadian"] = pipeline.predict(X_pred)` digunakan untuk menambahkan kolom "kepribadian" ke data pengujian `X_pred` yang berisi hasil prediksi yang diberikan oleh model. Penjelasan langkah-langkahnya:

`X_pred["kepribadian"]`: Dengan menggunakan `X_pred["kepribadian"]`, kita merujuk pada kolom "kepribadian" di dalam `DataFrame X_pred`. Jika kolom tersebut belum ada, maka kolom baru akan dibuat. `pipeline.predict(X_pred)`: Di sini, kita menggunakan metode `predict()` dari objek pipeline untuk melakukan prediksi pada data pengujian `X_pred`. Metode `predict()` akan menerapkan langkah-langkah pemrosesan data yang telah ditentukan dalam preprocessor terlebih dahulu pada `X_pred`, kemudian menggunakan model yang telah dilatih untuk menghasilkan prediksi pada data tersebut. Hasil prediksi akan menjadi nilai-nilai yang akan ditambahkan ke kolom "kepribadian" di `X_pred`. `X_pred`: Setelah menjalankan pernyataan `X_pred["kepribadian"] = pipeline.predict(X_pred)`, `DataFrame X_pred` akan terupdate dengan penambahan kolom "kepribadian" yang berisi hasil prediksi dari model.

Dengan menambahkan kolom hasil prediksi ke `DataFrame X_pred`, Anda dapat dengan mudah melihat prediksi yang diberikan oleh model untuk setiap sampel data pengujian dan melakukan analisis lebih lanjut jika diperlukan. Jika dilihat dari koding diatas maka hasil prediksinya adalah angka 2, yang artinya mahasiswa tersebut termasuk yang memiliki kepribadian Koleris.

5. Kesimpulan dan Saran

Hasil kegiatan penelitian yang dimana pengambilan dataset menggunakan penyebaran kuisioner kepada mahasiswa di Kota Batam. Dan sudah dilakukan preprocessing dan dilakukan pembagian data train dan data testing serta telah diterapkan klasifikasi dengan algoritma Naïve Bayes menghasilkan Train Score nya 0,947 dan Test Scorenya 0,879. Sudah dilakukan uji coba juga untuk dilakukan prediksi dengan data baru yang hasilnya tepat. Adapun saran diharapkan untuk pengambilan data kedepan tidak menggunakan kuisioner agar hasilnya tidak biar. Bisa menggunakan metode lain seperti crawling data dari mahasiswa di sosial media. Atau dengan cara scraping web sosial media.

Ucapan Terima Kasih (Arial, 10, Bold)

Terima kasih kepada semua pihak yang memdukung penuh kegiatan penelitian ini baik rekan dosen, mahasiswa serta Universitas Putera Batam yang memberikan support berupa fasilitas dan dana.

Daftar Pustaka (Arial, 10, Bold)

- D. M. Alam, A. T. D. Prabowo, A. Prabono, And M. W. Pratama, "Klasifikasi Karakteristik Kepribadian Mahasiswa Universitas Amikom Yogyakarta Dengan Menggunakan Metode Naive Bayes," Pp. 17–28, 2020.
- Fitriana, Frizka (2021) Analisis Sentimen Opini Terhadap Vaksin Covid-19 pada Media Sosial Twitter Menggunakan Support Vector Machine dan Naive Bayes. *Jurnal Komtika (Komputasi dan Informatika)* Vol. 5 No. 1 2021
- Fahrudy. D, dkk. 2022. Intelligent System For Classification Of Student Personality With Naive Bayes Algorithm. *SINTECH JOURNAL E-ISSN 2598-9642* Vol. 5 No 1 – April 2022
- Krisdiyanto, Taofik (2021). Analisis Sentimen Opini Masyarakat Indonesia Terhadap Kebijakan PPKM pada Media Sosial Twitter Menggunakan Naïve Bayes Clasifiers, *Journal CoreIT*. Vol 7, No.1 2021
- Liu L., Preotiuc-Pietro D., Samani Z. R., Moghaddam M. E, Ungar L. H., Analyzing Personality through Social Media Profile Picture Choice. In Tenth international AAAI conference on web and social media (ICWSM), 2016, 211-220.

Tadhan and F. R. Purba, "Aplikasi tes personality plus berbasis web," vol. Vol. 01 No, pp.327–337, 2022.