LAPORAN TUGAS BESAR II

IF3170 INTELIGENSI BUATAN

IMPLEMENTASI ALGORITMA DARI EXPLORATORY DATA ANALYSIS (KNN DAN NAIVE-BAYES)

Diajukan sebagai tugas besar Mata Kuliah IF3170 Inteligensi Buatan pada Semester I

Tahun Akademik 2023/2024

Anggota Kelompok:

Bintang Hijriawan	13521003
Jason Rivalino	13521008
Christophorus Dharma Winata	13521009
M. Malik I. Baharsyah	13521029

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2023/2024

DAFTAR ISI

BAB I
DESKRIPSI TUGAS

Dari pemrosesan dan EDA yang dilakukan pada tugas kecil 2, pada tugas besar 2 ini akan mengimplementasikan algoritma pembelajaran mesin yang telah kalian pelajari di kuliah, yaitu KNN dan Naive-Bayes. Data yang digunakan sama seperti data pada tugas kecil 2. Latihlah model dengan menggunakan data latih, kemudian validasi hasil dengan menggunakan data validasi untuk mendapatkan *insight* seberapa baik model melakukan generalisasi.

Untuk implementasi algoritma dari KNN dan Naive-Bayes, implementasi dilakukan dengan dua cara yaitu melalui *scratch* dan menggunakan pustaka *skicit-learn*. Untuk Model yang ada harus bisa di-*save* dan di-*load*. Implementasinya dibebaskan (misal menggunakan .txt, .pkl, dll). Terakhir, terdapat penilaian bonus yaitu dengan melakukan submisi pada aplikasi Kaggle.

BAB II
PENJELASAN ALGORITMA PROGRAM

2.1. Penjelasan Singkat Algoritma K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) adalah algoritma pembelajaran mesin (machine learning) yang digunakan untuk pemrosesan klasifikasi dan regresi. Algoritma ini merupakan jenis learning yang tidak parametrik, yang berarti tidak ada asumsi tentang distribusi data atau struktur model sebelumnya. Algoritma KNN bekerja berdasarkan prinsip bahwa objek yang memiliki ciri serupa cenderung berada dekat antara satu dengan lainnya.

Adapun untuk tahapan dari implementasi algoritma KNN dalam pencarian akurasi nilai prediksi dari data latih terhadap data validasi adalah sebagai berikut:

1. Memisahkan *dataset* menjadi dua bagian utama, yaitu data latih (*training set*) dan data validasi (*validation set*)
2. Menentukan nilai K, yaitu jumlah tetangga terdekat yang akan digunakan untuk membuat prediksi. Nilai K dapat ditentukan berdasarkan pengujian dan validasi model.
3. Mengukur jarak dengan menghitung jarak antara titik data yang akan diprediksi dengan semua titik data lainnya. Jarak dapat diukur dengan metrik seperti Euclidean *distance*, Manhattan *distance*, Minkowski *distance*, dll, tergantung pada kasus penggunaan.
4. Mengidentifikasi tetangga terdekat, yaitu dengan memilih sejumlah K tetangga terdekat berdasarkan jarak yang diukur.
5. Voting (Klasifikasi) atau Average (Regresi). Untuk permasalahan klasifikasi, algoritma melakukan "voting" berdasarkan kelas dari tetangga terdekat dan kelas yang paling banyak muncul menjadi hasil prediksi. Untuk permasalahan regresi, algoritma menghitung rata-rata nilai target dari tetangga terdekat.
6. Membuat prediksi dengan menggunakan hasil voting atau average dari langkah no. 5

2.2. Penjelasan Singkat Algoritma Naive-Bayes

Algoritma Naive Bayes merupakan algoritma klasifikasi yang berdasarkan pada teorema Bayes dengan asumsi bahwa setiap fitur dari suatu kelas adalah independen satu sama lain. Dengan kata lain, algoritma ini menganggap bahwa adanya suatu fitur dalam kelas tertentu tidak memberikan informasi tentang keberadaan fitur lainnya dalam kelas yang sama. Meskipun asumsi ini terkadang tidak sesuai dengan realitas, tetapi algoritma Naive Bayes dalam banyak kasus mampu memberikan hasil yang baik dan cepat untuk data dengan dimensi tinggi.

Tahapan implementasi algoritma Naive Bayes dalam pencarian akurasi nilai prediksi dari data latih terhadap data validasi umumnya melibatkan langkah-langkah berikut:

1. Menyesuaikan model klasifikasi Naive Bayes dengan jenis data. Salah satu contohnya adalah Naive Bayes Gaussian untuk data kontinu dan Naive Bayes Multinomial untuk data diskrit.
2. Memisahkan *dataset* menjadi dua bagian utama, yaitu data latih (*training set*) dan data validasi (*validation set*).
3. Melakukan *preprocessing data*, seperti membersihkan data dari *missing values*, melakukan normalisasi atau standarisasi data apabila diperlukan, dan mengonversi data ke format yang sesuai dengan model Naive Bayes terpilih.
4. Menggunakan data latih untuk melatih model Naive Bayes. Probabilitas prior dihitung untuk setiap kelas dan estimasi probabilitas kondisional untuk setiap fitur dalam setiap kelas.
5. Setelah model Naive Bayes dilatih, probabilitas untuk setiap kelas dihitung dan pilih kelas dengan probabilitas tertinggi sebagai hasil prediksi.
6. Mengevaluasi kinerja dan optimasi model menggunakan metrik-metrik seperti akurasi, presisi, *recall*, atau F1-*score* tergantung pada karakteristik klasifikasi yang diinginkan serta melakukan perbaikan pada masalah tertentu selama proses evaluasi.
7. Setelah model dianggap sudah mumpuni/memadai, model dapat diuji pada *dataset* eksternal yang belum pernah dilihat/ditemui sebelumnya untuk mengukur kemampuan prediksi model.

BAB III
ANALISIS HASIL PREDIKSI

3.1. Hasil Prediksi dari Algoritma KNN pada *scratch*

Pencarian hasil prediksi dengan menggunakan Algoritma KNN memiliki alur sebagai berikut:

a. Dimulai dengan membagi data latih dan data valid menjadi prediktor dan target
b. Melakukan proses fitting dengan langkah sebagai berikut:

   Adapun pengujian dan validasi model adalah sebagai berikut:

   1. Satu data validasi diambil dan dilakukan kalkulasi *euclidean distance* dari data tersebut dengan data latih dan hasil kalkulasi dicatat
   2. Setelah data sudah dicatat, data distance diurutkan secara membesar
   3. Lakukan mean dari atribut target k data pertama dengan k adalah $k = 1, 2, 3, \dots, n$ dan $n =$ banyak data latih dan catat semua nilai mean tersebut. Atribut target adalah kolom pada dataset yang ditentukan dari kolom-kolom lainnya, lain halnya dengan atribut prediksi yang mana merupakan atribut penentu dari atribut target.
   4. Nilai mean dalam hal ini menjadi nilai yang merupakan hasil prediksi dari model KNN terhadap atribut target
   5. Nilai prediksi ini dibandingkan dengan nilai target pada data valid pada langkah 1. Jika nilai prediksi berada pada rentang $[x - 0.5, x + 0.5)$ dengan x adalah atribut target data valid, nilai x dan nilai prediksi juga disesuaikan agar sesuai dengan rentang nilai yang mungkin pada atribut target.
   6. Lakukan perhitungan banyak prediksi benar dari semua nilai k, dan ambil k yang paling banyak menghasilkan prediksi benar
   7. Jika pada pencarian ditemukan nilai k dengan kebenaran yang sama, akan diambil nilai yang lebih kecil untuk mencegah terjadinya overfitting

Berdasarkan hasil k yang didapat, didapatkan hasil prediksi yaitu sebanyak 563 kali dengan total validasi pada 600 data sehingga tingkat keakuratan data didapat sebesar 93,88%.

Data-data dari k yang didapat dari fitting dicatat dengan akurasi seperti berikut

Result of 10 best k values

	k	frequency
18	19	563
24	26	563
17	18	562
7	8	561
11	12	560
55	58	560
20	21	560
59	62	560
53	56	560
208	24	560

Nilai-nilai k yang didapatkan berdasarkan fitting

```
K values with the most correct predictions are K = [19, 26]
With said K value(s) being correct 563 times out of 600 (0.9383333333333334)
According to sklearn.metrics, the validity score of the model with accuracy_score() is 0.9383333333333334
```

Hasil akurasi dari algoritma KNN

Berdasarkan hasil yang akan dijelaskan di bawah, didapatkan bahwa algoritma KNN dapat memberikan nilai yang lebih tinggi jika dibandingkan dengan implementasi pencarian menggunakan algoritma Naive-Bayes. Hal ini disebabkan pada algoritma KNN dapat menangani data yang bersifat *noisy* dengan memanfaatkan nilai *k* pada saat fitting data latih dan data validasi sehingga bisa dipergunakan untuk prediksi data yang berukuran besar.

3.2. Hasil Prediksi dari Algoritma Naive-Bayes pada *scratch*

Untuk pencarian hasil prediksi dengan menggunakan Algoritma Naive-Bayes, alur yang dilakukan adalah sebagai berikut:

a. Melakukan pemisahan data antara data latih dan juga data validasi. Pada program yang ada, data latih ditandai dengan variabel x dan data validasi ditandai dengan huruf y. Selain melakukan pemisahan antara data, pada kolom juga dilakukan pemisahan berdasarkan kolom atribut dan kolom target. Sehingga pada akhir didapatkan total sebanyak empat variabel.

b. Melakukan proses *fitting* yaitu untuk memisahkan kolom target menjadi beberapa kelompok berdasarkan nilai yang dimiliki antara 0, 1, 2, dan 3. Pada setiap kelompok, akan dilakukan perhitungan terhadap nilai rata-rata, variansi, dan probabilitas prior. Perhitungan nilai yang ada ini memiliki tujuan untuk membantu pencarian prediksi dengan rumus Gaussian.

c. Menjalankan proses prediksi yang diawali dengan menginisiasi array kosong yang nantinya akan menjadi tempat untuk nilai-nilai yang didapat dari proses prediksi.

d. Melakukan perulangan untuk menghitung nilai probabilitas yang ada pada tiap kelas (priors) dan pada tiap atribut (likelihood). Perhitungan untuk priors dilakukan dengan memanfaatkan logaritma dan untuk likelihood, perhitungan akan memanfaatkan rumus distribusi probabilitas Gaussian. Untuk rumus distribusi probabilitas Gaussian, rumus yang ada yaitu sebagai berikut:

$$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$\sigma = variansi$$
$$\mu = mean$$

Rumus distribusi probabilitas Gaussian

Sumber: https://www.i2tutorials.com/advantages-and-disadvantages-of-naive-bayes-classifier/

e. Melakukan kalkulasi untuk menentukan posteriors dengan melakukan penambahan antara nilai priors dengan nilai posteriors

f. Dari semua posteriors yang ada, kemudian melakukan pemilihan untuk posteriors dengan nilai yang tertinggi (maksimum) untuk dikembalikan sebagai daftar nilai prediksi
g. Melakukan perhitungan akurasi data dengan mengkalkulasikan perbandingan dengan menghitung kesamaan antara nilai kolom target untuk data validasi dengan nilai prediksi yang didapatkan dan hasilnya dibagi dengan total seluruh data yang ada.
h. Setelah melakukan percobaan, kemudian data model yang ada akan disimpan dalam bentuk .txt dengan memanfaatkan library pickle dan dilakukan pengujian pencarian akurasi dengan

Berdasarkan hasil percobaan yang dilakukan oleh kelompok kami, didapatkan hasil untuk perbandingan kesamaan yaitu sebesar 469 data dengan total data yang ada sebanyak 600 sehingga tingkat keakuratan data didapat sebesar 78,16%.

```
Jumlah kolom target data validasi yang sama dengan hasil prediksi: 469
Jumlah baris total data validasi: 600
```

```
Hasil akurasi dengan Naive-Bayes sebesar 0.7816666666666666
```

Untuk persebaran kesamaan nilai, dilakukan pengecekan dengan memanfaatkan *confusion matrix* untuk menemukan jumlah kesamaan nilai. Dari confusion matrix, didapat hasil sebagai berikut.

a. Untuk nilai yang sama dari kelompok range harga bernilai 0 yaitu sebanyak 125 data
b. Untuk nilai yang sama dari kelompok range harga bernilai 1 yaitu sebanyak 93 data
c. Untuk nilai yang sama dari kelompok range harga bernilai 2 yaitu sebanyak 110 data
d. Untuk nilai yang sama dari kelompok range harga bernilai 3 yaitu sebanyak 141 data

Predicted	0	1	2	3
Validation				
0	125	17	0	0
1	18	93	33	0
2	0	30	110	15
3	0	0	18	141

Berdasarkan hasil yang ada, didapat bahwa untuk akurasi yang ada memang tidak terlalu tinggi dan masih lebih rendah jika dibandingkan dengan pengecekan prediksi dengan menggunakan algoritma KNN. Hal ini dapat terjadi dikarenakan terdapat beberapa kelemahan yang ada pada algoritma Naive-Bayes seperti memungkinkan terjadinya frekuensi bernilai 0 ketika ditemukan nilai pada data validasi yang sama sekali tidak terdapat pada data uji yang mengakibatkan tidak dapat membuat data prediksi. Selain itu, tiap atribut yang ada dalam prediksi ini memiliki sifat independen yang kuat tetapi dalam kebanyakan kasus, atribut yang ada pasti memiliki relasi dan tidak independen sepenuhnya.

3.3. Perbandingan hasil Algoritma KNN dari *scratch* dengan hasil pada Pustaka

Pencarian hasil prediksi dengan algoritma KNN pada pustaka sklearn dilakukan dengan cara berikut:

a. Melakukan pemisahan kolom target pada data latih dan data validasi. Kolom target untuk data latih dan data validasi ditandai dengan variabel y_train dan y_test, dan kolom lainnya untuk data latih dan data validasi ditandai dengan variabel x_train dan x_test
b. Melakukan pencarian terhadap nilai k terbaik dengan looping dan menjalankan fungsi sklearn untuk pencarian dengan metode KNN. Metode KNN dengan sklearn dituliskan dengan nama fungsi KNeighborClassifier(*k, weight, p*) yang menerima parameter k yakni banyak k neighbor, weight yang diisi dengan “distance” dimana artinya setiap data akan lebih condong untuk dipilih jika mempunyai distance lebih kecil, dan p yang diisi dua untuk menjelaskan penghitungan distance dengan euclidean distance

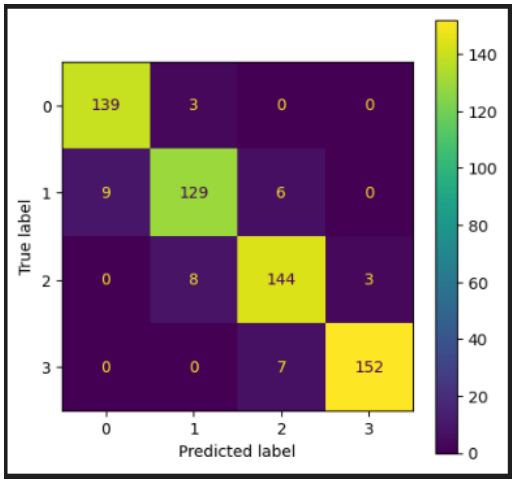
c. Membuat prediksi dengan method KNN_Skicit.predict(data_test). Fungsi ini akan mengembalikan array yang berisi nilai prediksi untuk setiap baris di data_test
d. Melakukan perubahan nilai akurasi berdasarkan kondisi, jika ditemukan kondisi nilai yang lebih baik berdasarkan pencarian k, akan melakukan perubahan nilai dari akurasi yang ada. Jika sudah tidak ditemukan nilai akurasi yang lebih baik (nilai accuracy temp sudah sama dengan nilai accuracy), program akan memasukkan nilai i dari looping sebagai nilai k
e. Menampilkan nilai akurasi akhir, classification report, serta membuat confusion matrix menggunakan pustaka sklearn.metrics

Berdasarkan hasil percobaan, didapatkan hasil benar sebanyak 564 data dari total 600 data, sehingga nilai akurasi yang didapat sebesar 94%

```
k = [56]
accuracy = 0.94
```

Untuk melihat persebaran kesamaan nilai, dapat dilakukan dengan melihat confusion matrix berikut:

```
Confusion Matrix:
[[139   3   0   0]
 [  9 129   6   0]
 [  0   8 144   3]
 [  0   0   7 152]]
```



Dari confusion matrix diatas, didapatkan hasil sebagai berikut:

a. Untuk nilai yang sama dari kelompok range harga bernilai 0 yaitu sebanyak 139 data
b. Untuk nilai yang sama dari kelompok range harga bernilai 1 yaitu sebanyak 129 data
c. Untuk nilai yang sama dari kelompok range harga bernilai 2 yaitu sebanyak 144 data
d. Untuk nilai yang sama dari kelompok range harga bernilai 3 yaitu sebanyak 152 data

Dari hasil analisis yang ada, hasil algoritma KNN yang diimplementasikan dengan menggunakan *library* sklearn menunjukkan adanya sedikit perbedaan dengan hasil yang ada secara *scratch*. Pada *scratch*, nilai akurasi prediksi yang dihasilkan yaitu sebesar 93,83% dengan selisih satu kesamaan nilai yang berbeda. Meskipun terdapat perbedaan, tetapi perbedaan tersebut nilainya tidak terlalu besar dan sudah cukup dekat antara kode implementasi *scratch* dengan penggunaan library sklearn.

3.4. Perbandingan hasil Algoritma Naive-Bayes dari *scratch* dengan hasil pada Pustaka

Pencarian hasil prediksi dengan algoritma Naive-Bayes pada pustaka sklearn dilakukan dengan cara berikut:

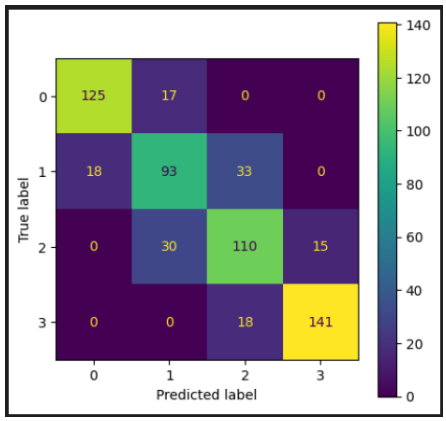
a. Melakukan pemisahan kolom target pada data latih dan data validasi. Kolom target untuk data latih dan data validasi ditandai dengan variabel y_train dan y_test, dan kolom lainnya untuk data latih dan data validasi ditandai dengan variabel x_train dan x_test
b. Membuat prediksi dengan method classifier.predict(data_test). Fungsi ini akan mengembalikan array yang berisi nilai prediksi untuk setiap baris di data_test
c. Menghitung akurasi dan classification report, serta membuat confusion matrix menggunakan pustaka sklearn.metrics

Berdasarkan hasil percobaan, didapatkan hasil benar sebanyak 469 data dari total 600 data, sehingga nilai akurasi yang didapat sebesar 78.17%

```
Jumlah prediksi yang benar: 469
Jumlah total baris: 600
Accuracy: 0.7816666666666666

Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.88      0.88       142
           1       0.66      0.65      0.65       144
           2       0.68      0.71      0.70       155
           3       0.90      0.89      0.90       159

    accuracy                           0.78       600
   macro avg       0.78      0.78      0.78       600
weighted avg       0.78      0.78      0.78       600
```

Untuk melihat persebaran kesamaan nilai, dapat dilakukan dengan melihat confusion matrix berikut:

```
Confusion Matrix:
[[125  17   0   0]
 [ 18  93  33   0]
 [  0  30 110  15]
 [  0   0  18 141]]
```



Dari confusion matrix diatas, didapatkan hasil sebagai berikut:

e. Untuk nilai yang sama dari kelompok range harga bernilai 0 yaitu sebanyak 125 data
f. Untuk nilai yang sama dari kelompok range harga bernilai 1 yaitu sebanyak 93 data
g. Untuk nilai yang sama dari kelompok range harga bernilai 2 yaitu sebanyak 110 data
h. Untuk nilai yang sama dari kelompok range harga bernilai 3 yaitu sebanyak 141 data

Dari hasil analisis yang ada, hasil algoritma Gaussian Naive-Bayes dari pustaka sklearn menunjukan hasil yang sama dengan algoritma Naive-Bayes yang diaplikasikan dengan menggunakan scratch dengan nilai akurasi dan persebaran kesamaan nilai yang sama. Hal ini menunjukkan bahwa implementasi algoritma yang dilakukan dengan scratch sudah sesuai dan benar.

3.5. Pemrosesan Submisi Kaggle

Proses untuk melakukan submisi pada kompetisi Kaggle dapat diuraikan sebagai berikut:

1. Melakukan *training* model KNN yang telah dibuat dengan *data_train.csv* sebagai data latih dan *data_validation.csv* sebagai data validasi. Langkah ini bertujuan untuk menemukan nilai optimal untuk parameter k dalam model.
2. Setelah model KNN dilatih, langkah selanjutnya adalah melakukan prediksi terhadap data uji dari *test.csv*. Hasil prediksi ini berupa *array* nilai untuk setiap baris data uji, yang merepresentasikan kategori yang diprediksi.

3. Hasil prediksi yang awalnya berupa *array integer* akan diubah menjadi struktur data DataFrame. Data ini kemudian disimpan sebagai file CSV dengan nama *submission.csv*. *File* ini akan berisi dua kolom, yaitu 'id' (identifikasi data uji) dan 'price_range' (hasil prediksi).
4. Setelah file *submission.csv* dibuat, langkah terakhir adalah mengunggahnya ke laman kompetisi Kaggle "Tugas Besar 2 IF3170 2023/2024". Proses ini dilakukan melalui halaman submisi pada platform Kaggle.

BAB IV
PEMBAGIAN KERJA

Nama	NIM	Pembagian Kerja
Bintang Hijriawan	13521003	Algoritma KNN, laporan
Jason Rivalino	13521008	Algoritma Naive Bayes, implementasi dengan pustaka sklearn Naive Bayes, laporan
Christophorus Dharma Winata	13521009	Algoritma KNN, implementasi KNN dengan pustaka sklearn KNN, submisi Kaggle, laporan
M. Malik I. Baharsyah	13521029	Algoritma Naive Bayes, submisi Kaggle, laporan

DAFTAR PUSTAKA

[1] https://www.datacamp.com/tutorial/k-nearest-neighbor-classification-scikit-learn

[2] https://www.geeksforgeeks.org/ml-implementation-of-knn-classifier-using-sklearn/

[3] https://www.datacamp.com/tutorial/naive-bayes-scikit-learn

[4] https://machinelearningmastery.com/naive-bayes-classifier-scratch-python/

LAMPIRAN

Github: **https://github.com/jasonrivalino/Tubes2AI-EDAImplemetation**