PREDIKSI PENJUALAN MOBIL BEKAS MENGGUNAKAN METODE GAUSSIAN NAIVE BAYES CLASSIFIER

Proyek ini disusun untuk memenuhi tugas akhir mata kuliah Penambangan Data pada Program Sarjana Teknik Informatika Fakultas Ilmu Komputer Universitas Dian Nuswantoro Semester Ganjil Tahun Akademik 2023/2023

Dosen Pengampu
Dr. Aris Marjuni, S. Si, M. Kom

TIM PROYEK:

No.	NIM	NAMA
1.	A11.2017.10276	Sofwan Hidayat
2.	A11.2020.13205	Athiya Nahdhiana
3.	A11.2019.12030	Anggito Budhi Prasojo
4.	A11.2021.13605	Evan faiz

ABSTRAKSI PROYEK

Pasar penjualan mobil bekas kini makin diminati oleh masyarakat dan tidak kalah dengan mobil baru. Pilihan model yang ditawarkan kepada pembeli sangat beragam mulai dari model automatic / manual, mesin dengan bahan bakar diesel / bensin bahkan mesin elektrik . Proyek ini dilakukan dengan tujuan untuk mengklasifikasi model mobil mana yang paling laku terjual dan model mana yang tidak laku.

Klasifikasi dilakukan menggunakan pendekatan penambangan data dengan model Gaussian Naive Bayes Model. Dataset yang digunakan adalah dataset publik yang bersumber dari Kaggle.com. Hasil klasifikasi menunjukkan bahwa Gaussian Naive Bayes Model mampu mengidentifikasi potensi penjualan mobil bekas dengna akurasi 52%.

A. DATASET

Nama dataset:

bmw.xlsx

Sumber:

https://www.kaggle.com/datasets/adityadesai13/used-car-dataset-ford-and-mercedes

Informasi dataset:

10781 observations / records

9 attributes

Informasi Atribut:

1) model : model mobil;
2) year : tahun produksi mobil;
3) price : harga mobil;
4) transmission : tipe transmisi mesin mobil;
5) mileage : ukuran mile mobil;
6) fuelType : tipe bahan bakar pada mobil;
7) tax : pajak mobil;
8) mpg : ukuran mile per galon bahan bakar;
9) engineSize : ukuran mesin mobil;

Contoh instances:

Berikut tampilan 5 data instances pertama :

	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
0	5 Series	2014	11200	Automatic	67068	Diesel	125	57.6	2.0
1	6 Series	2018	27000	Automatic	14827	Petrol	145	42.8	2.0
2	5 Series	2016	16000	Automatic	62794	Diesel	160	51.4	3.0
3	1 Series	2017	12750	Automatic	26676	Diesel	145	72.4	1.5
4	7 Series	2014	14500	Automatic	39554	Diesel	160	50.4	3.0

B. DATA PREPROCESSING

Transformasi Data (Data Transformation), data akan diubah atau ditransformasikan menjadi bentuk yang sesuai dengan metode analisis yaitu tipe data numeric. Label encoding untuk mengonversi label kata menjadi angka. Label encoding mengacu pada proses transformasi label kata menjadi bentuk numerik. Dalam hal regresi jika memuat variabel kategori dan nilainya tidak bisa difaktorisasi dalam bentuk tingkatan, dilakukan proses dummy, setiap nilai dalam variabel itu menjadi variabel lain. Data preprocessing yang dilakukan terhadap dataset bertujuan agar dataset lebih mudah untuk diproses menggunakan metode gaussian naive bayes.

Merubah typedata menjadi numeric menggunakan label encoder

```
en = LabelEncoder()

dataset['transmission'] = en.fit_transform(dataset['transmission'])
dataset.head()
```

	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
0	5 Series	2014	11200	0	67068	Diesel	125	57.6	2.0
1	6 Series	2018	27000	0	14827	Petrol	145	42.8	2.0
2	5 Series	2016	16000	0	62794	Diesel	160	51.4	3.0
3	1 Series	2017	12750	0	26676	Diesel	145	72.4	1.5
4	7 Series	2014	14500	0	39554	Diesel	160	50.4	3.0

```
en = LabelEncoder()

dataset['model'] = en.fit_transform(dataset['model'])
dataset.head()
```

	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
0	4	2014	11200	0	67068	Diesel	125	57.6	2.0
1	5	2018	27000	0	14827	Petrol	145	42.8	2.0
2	4	2016	16000	0	62794	Diesel	160	51.4	3.0
3	0	2017	12750	0	26676	Diesel	145	72.4	1.5
4	6	2014	14500	0	39554	Diesel	160	50.4	3.0

```
en = LabelEncoder()

dataset['fuelType'] = en.fit_transform(dataset['fuelType'])
dataset.head()
```

	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
0	4	2014	11200	0	67068	0	125	57.6	2.0
1	5	2018	27000	0	14827	4	145	42.8	2.0
2	4	2016	16000	0	62794	0	160	51.4	3.0
3	0	2017	12750	0	26676	0	145	72.4	1.5
4	6	2014	14500	0	39554	0	160	50.4	3.0

Berikut merupakan keadaan dataset sebelum dan sesudah preprocessing :

Sebelum Preprocessing :

	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
0	5 Series	2014	11200	Automatic	67068	Diesel	125	57.6	2.0
1	6 Series	2018	27000	Automatic	14827	Petrol	145	42.8	2.0
2	5 Series	2016	16000	Automatic	62794	Diesel	160	51.4	3.0
3	1 Series	2017	12750	Automatic	26676	Diesel	145	72.4	1.5
4	7 Series	2014	14500	Automatic	39554	Diesel	160	50.4	3.0

Sesudah Preprocessing :

	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
0	4	2014	11200	0	67068	0	125	57.6	2.0
1	5	2018	27000	0	14827	4	145	42.8	2.0
2	4	2016	16000	0	62794	0	160	51.4	3.0
3	0	2017	12750	0	26676	0	145	72.4	1.5
4	6	2014	14500	0	39554	0	160	50.4	3.0

C. EXPLORATORY DATA ANALYSIS (EDA)

Berikut merupakan langkah-langkah EDA terhadap dataset dan capture hasilnya , yaitu:

- Deskripsi/informasi dataset : menggunakan referensi info()

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10781 entries, 0 to 10780
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   model         10781 non-null  int64
 1   year          10781 non-null  int64
 2   price         10781 non-null  int64
 3   transmission  10781 non-null  int64
 4   mileage       10781 non-null  int64
 5   fuelType      10781 non-null  int64
 6   tax           10781 non-null  int64
 7   mpg           10781 non-null  float64
 8   engineSize    10781 non-null  float64
dtypes: float64(2), int64(7)
memory usage: 758.2 KB
```

- Capture 10 data secara random : menggunakan referensi sample()

```
dataset.sample(10)
```

	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
8109	2	2018	19999	0	16882	0	145	62.8	2.0
4914	2	2016	18990	2	45090	0	165	51.4	3.0
1136	1	2015	12198	1	22812	0	20	74.3	1.5
6973	1	2015	16922	2	29538	4	200	43.5	2.0
5215	14	2020	37995	0	2500	4	145	34.0	2.0
7733	2	2013	7490	1	93000	4	145	47.9	1.6
7579	0	2016	17990	0	14000	4	235	37.7	3.0
2443	1	2019	28361	2	101	4	145	50.4	2.0
8002	2	2016	13699	0	72111	0	30	62.8	2.0
5789	17	2016	23912	0	89676	0	200	47.1	3.0

- Periksa apakah ada data yang kosong (missing) : menggunakan referensi isnull()

```
dataset.isnull()
```

	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...
10776	False	False	False	False	False	False	False	False	False
10777	False	False	False	False	False	False	False	False	False
10778	False	False	False	False	False	False	False	False	False
10779	False	False	False	False	False	False	False	False	False
10780	False	False	False	False	False	False	False	False	False

10781 rows × 9 columns

- Tampilkan informasi statistik dari dataset tersebut : menggunakan referensi describe()

```
dataset.describe()
```

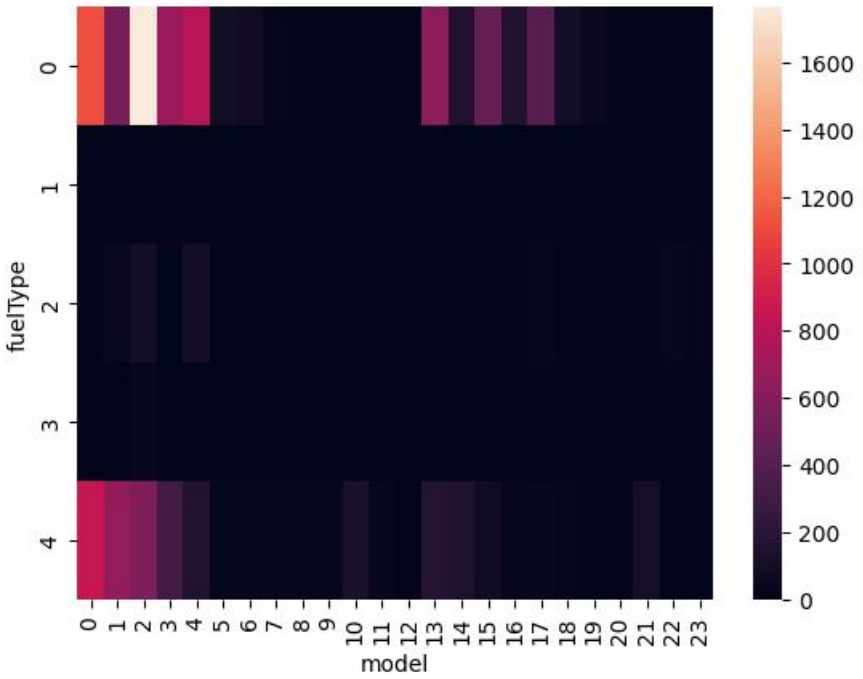
	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
count	10781.000000	10781.000000	10781.000000	10781.000000	10781.000000	10781.000000	10781.000000	10781.000000	10781.000000
mean	5.297839	2017.078935	22733.408867	1.099991	25496.986550	1.333364	131.702068	56.399035	2.167767
std	6.054688	2.349038	11415.528189	0.869297	25143.192559	1.853240	61.510755	31.336958	0.552054
min	0.000000	1996.000000	1200.000000	0.000000	1.000000	0.000000	0.000000	5.500000	0.000000
25%	1.000000	2016.000000	14950.000000	0.000000	5529.000000	0.000000	135.000000	45.600000	2.000000
50%	2.000000	2017.000000	20462.000000	1.000000	18347.000000	0.000000	145.000000	53.300000	2.000000
75%	10.000000	2019.000000	27940.000000	2.000000	38206.000000	4.000000	145.000000	62.800000	2.000000
max	23.000000	2020.000000	123456.000000	2.000000	214000.000000	4.000000	580.000000	470.800000	6.600000

- Plot correlogram (heatmap) : menggunakan referensi heatmap()

```
fuelType_cyl = (
  dataset
 .groupby('fuelType')
 .model
 .value_counts()
 .unstack()
 .fillna(0)
)
```

membuat heatmap (fuelType ke model)

```
sns.heatmap(fuelType_cyl);
```



D. PEMODELAN

Model prediksi Gaussian Naive Bayes adalah teknik dalam machine learning yang menggunakan pendekatan probabilitas dan distribusi Gaussian atau distribusi normal.

- Implementasi model Gaussian Naive Bayes pada dateset yang sudah dipreprocessing menjadi numerical value

Naive Bayes

```
#Pengecekan apakah typedata sudah menjadi numeric

dataset.head()
```

	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
0	4	2014	11200	0	67068	0	125	57.6	2.0
1	5	2018	27000	0	14827	4	145	42.8	2.0
2	4	2016	16000	0	62794	0	160	51.4	3.0
3	0	2017	12750	0	26676	0	145	72.4	1.5
4	6	2014	14500	0	39554	0	160	50.4	3.0

- Pendefinisian data variabel bebas dan tidak bebas

```
x = dataset.iloc[:, :3].values
y = dataset.iloc[:, 3].values
```

```
print (x)
```

```
[[    4  2014 11200]
 [    5  2018 27000]
 [    4  2016 16000]
 ...
 [    2  2017 13100]
 [    0  2014  9930]
 [   13  2017 15981]]
```

```
print (y)
```

```
[0 0 0 ... 1 0 0]
```

- Pendefinisian data training dan testing

```
x_train, x_test, y_train, y_test= train_test_split(x,y, test_size=0.2, random_state=123)

print("x_train = ", len(x_train))
print("x_test = ", len(x_test))
print("y_train = ", len(y_train))
print("y_test = ", len(y_test))
```

```
x_train =  8624
x_test =  2157
y_train =  8624
y_test =  2157
```

Standarisasi fitur dengan menghapus rata-rata dan penskalaan ke varians satuan. Ini berarti kita mengubah fitur sehingga memiliki rata-rata 0 dan varians 1.

```
sc = StandardScaler()

x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
```

```
x_train
```

```
array([[-0.8701173 ,  0.82024879, -0.16552358],
       [ 1.61906588, -0.03443925, -0.18953984],
       [-0.70417176, -0.88912729, -1.34275712],
       ...,
       [-0.8701173 , -0.03443925, -0.71353103],
       [ 1.95095697,  0.82024879,  1.85716975],
       [-0.8701173 , -1.74381533, -1.08111085]])
```

```
x_test
```

```
array([[ 1.28717479, -0.46178327, -0.28577955],
       [ 1.28717479, -0.03443925, -0.86208253],
       [ 0.78933815,  0.82024879,  2.01978168],
       ...,
       [-0.8701173 , -0.03443925, -0.37546938],
       [-0.37228067,  0.82024879,  0.59120703],
       [-0.8701173 , -1.74381533, -1.02801307]])
```

```
y_train
```

```
array([1, 0, 1, ..., 2, 0, 1])
```

```
y_test
```

```
array([2, 1, 2, ..., 2, 2, 1])
```

- Hasil eksperimen dan pengukuran/evaluasi model

KLASIFIKASI

```
classifier = GaussianNB()
classifier.fit(x_train,y_train)
```

```
▾ GaussianNB
GaussianNB()
```

PREDIKSI

```
y_prediksi = classifier.predict(x_test)
y_prediksi
```

```
array([0, 0, 2, ..., 1, 2, 1])
```

PROBABILITAS

```
classifier.predict_proba(x_test)
```

```
array([[5.47549981e-01, 5.60441957e-02, 3.96405823e-01],
       [4.72904449e-01, 8.62184472e-02, 4.40877104e-01],
       [1.58770871e-01, 1.21413237e-07, 8.41229008e-01],
       ...,
       [2.18273455e-01, 4.84384683e-01, 2.97341862e-01],
       [2.34568813e-01, 1.66160711e-02, 7.48815116e-01],
       [1.42755460e-01, 8.50453651e-01, 6.79088924e-03]])
```

```
cm = confusion_matrix(y_test, y_prediksi)
print(cm)
```

```
[[103 308 296]
 [ 46 349 117]
 [ 55 201 682]]
```

AKURASI

```
akurasi = classification_report(y_test,y_prediksi)
print(akurasi)
```

```
              precision    recall  f1-score   support

           0       0.50      0.15      0.23       707
           1       0.41      0.68      0.51       512
           2       0.62      0.73      0.67       938

    accuracy                           0.53      2157
   macro avg       0.51      0.52      0.47      2157
weighted avg       0.53      0.53      0.49      2157
```

```
akurasi = accuracy_score(y_test,y_prediksi)
print("Tingkat Akurasi : %d persen"%(akurasi*100))
```

```
Tingkat Akurasi : 52 persen
```

```
ydata = pd.DataFrame()
ydata['y_test'] = pd.DataFrame(y_test)
ydata['y_prediksi'] = pd.DataFrame(y_prediksi)
ydata
```

	y_test	y_prediksi
0	2	0
1	1	0
2	2	2
3	2	2
4	2	1
...	...	...
2152	1	1
2153	2	0
2154	2	1
2155	2	2
2156	1	1

2157 rows × 2 columns

```
ydata.to_excel('numpybmwdata.xlsx', index=False)
```

E. KESIMPULAN

Naive Bayes adalah salah satu algoritma yang sering digunakan dalam pengenalan pola, analisis teks, klasifikasi dokumen, dan banyak aplikasi lainnya. Algoritma Naive Bayes mengandalkan asumsi dasar yang cukup sederhana, yaitu bahwa semua atribut (fitur) yang digunakan dalam klasifikasi adalah independen satu sama lain. Oleh karena itu, istilah "naive" digunakan, karena dalam dunia nyata, atribut seringkali tidak benar-benar independen. Namun, asumsi ini mempermudah perhitungan matematis dan sering kali menghasilkan hasil yang cukup baik, terutama dalam kasus klasifikasi teks. Rumus dasar Naive Bayes adalah berdasarkan pada Teorema Bayes, yang menghubungkan probabilitas suatu peristiwa dengan probabilitas peristiwa lain yang berhubungan. Dalam konteks klasifikasi, Naive Bayes digunakan untuk menghitung probabilitas bahwa sebuah sampel data termasuk dalam suatu kelas tertentu.

F. KONTRIBUSI ANGGOTA

Tuliskan persentase kontribusi masing-masing anggota dalam pekerjaan tugas ini. Kontribusi diukur dari peran aktif anggota, khususnya dalam kerja sama tim untuk menyelesaikan proyek ini.

No.	NIM	NAMA	KONTRIBUSI DAN KEAKTIFAN (%)
1.	A11.2017.10276	Sofwan Hidayat	25%
2.	A11.2020.13205	Athiya Nahdhiana	25%
3.	A11.2019.12030	Anggito Budhi Prasojo	25%
4.	A11.2021.13605	Evan faiz	25%