

Analisis Cuaca di Szeged Tahun 2006- 2016 Menggunakan Model Regresi Linier & Naïve Bayes

December 13, 2022



Nama Anggota:

- Ken Dahana - 160420115
- Viqram Ananta Wataf - 160420119



Linear Regression

Peramalan cuaca adalah proses teknologi dan ilmu pengetahuan yang digunakan untuk memprediksi kondisi atmosfer untuk lokasi yang diberikan. Analisis cuaca di Szeged Tahun 2006-2016 menggunakan model regresi linier kami dapat menentukan hubungan antara variabel cuaca seperti suhu, kelembaban, dan kecepatan angin dengan variabel waktu. Dengan menggunakan data historis cuaca di Szeged, kami dapat menentukan model regresi linier yang menunjukkan hubungan antara variabel cuaca dan waktu.

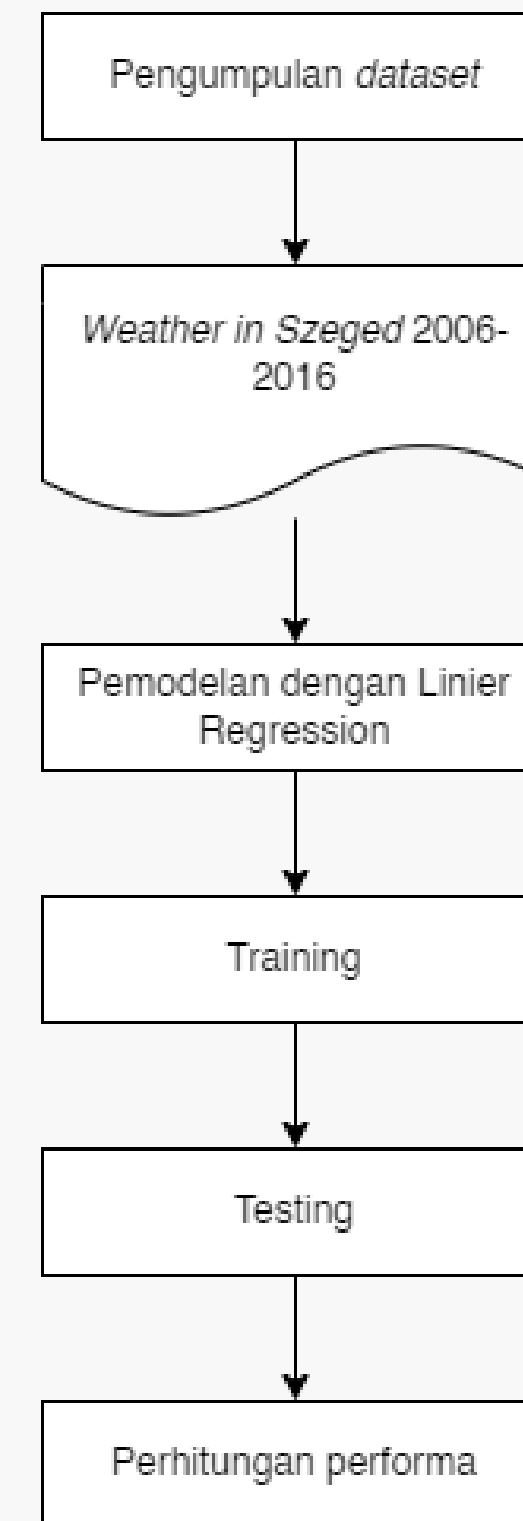


METODE PENELITIAN

Terdapat empat tahapan yang dilakukan dalam penelitian menggunakan model linear regression ini.

- Pengumpulan Dataset
- Pembentukan Model
- Training dan Testing
- Perhitungan Performa

Gambar disamping, disajikan diagram blok alur proses penelitian secara lebih detail.



Gambar 1. Diagram Blok Alur Proses Penelitian Linear Regression



Naïve Bayes

Peramalan cuaca adalah proses teknologi dan ilmu pengetahuan yang digunakan untuk memprediksi kondisi atmosfer untuk lokasi yang diberikan. Analisis cuaca di Szeged Tahun 2006-2016 menggunakan model Naive Bayes kami dapat mengategorikan percipitation type dan summary yang ada di dataset. Dengan ini, kami dan memprediksi percipitation type dan summary apa yang cocok untuk suatu data.

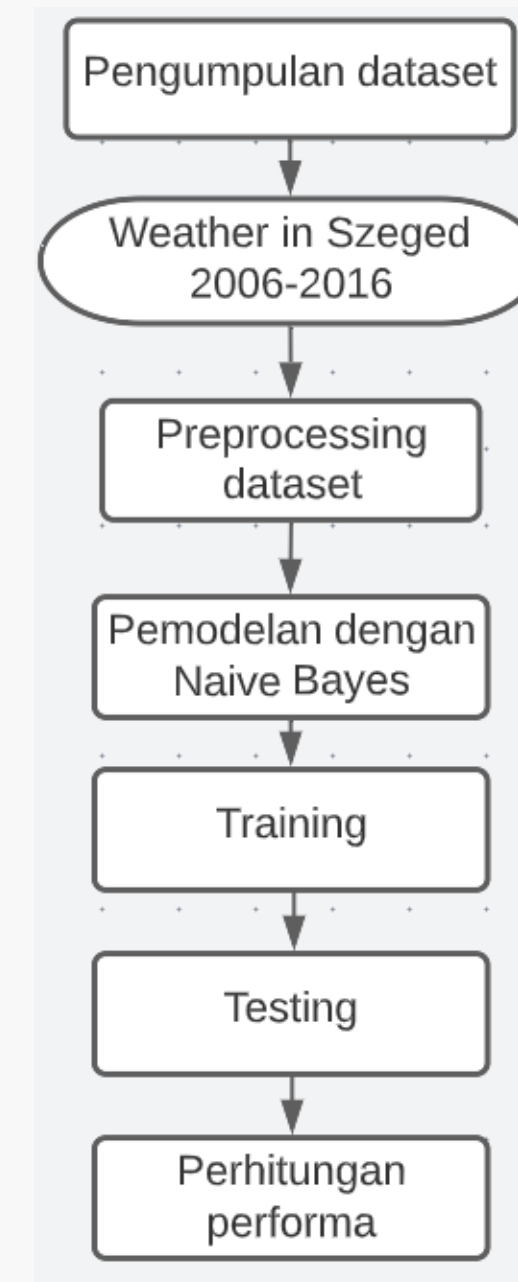


METODE PENELITIAN

Terdapat empat tahapan yang dilakukan dalam penelitian menggunakan model naive bayes.

- Pengumpulan Dataset
- Pre-processing Dataset
- Pembentukan Model
- Training dan Testing
- Perhitungan Performa

Gambar disamping, disajikan diagram blok alur proses penelitian secara lebih detail.



Gambar 2. Diagram Blok Alur Proses Penelitian Naive Bayes



METODE PENELITIAN

Pembuatan Model dan Training-Testing

Himpunan data mencakup dua variabel yang di ambil untuk tujuan penelitian ini. Model klasifikasi yang di implementasikan dalam penelitian ini adalah Regresi Linier dari library scikit-learn.

```
#generate data from computer  
#Weather in Szeged 2006-2016  
weather = pd.read_csv('/Users/Viqram  
weather.shape  
  
(96453, 12)
```

Gambar 3. Jumlah Data dan Fitur Dataset

Jumlah data yang digunakan untuk data training adalah 70% sedangkan sisanya digunakan untuk data testing. Berikut adalah visualisasi data dan fitur dari dataset yang digunakan.

Parameter	Spesifikasi
CPU	Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz 2.59 GHz
RAM	16 GB
Sistem Operasi	Windows 10
GPU	Intel(R) UHD Graphics

Gambar 4. Jumlah Data dan Fitur Dataset



METODE PENELITIAN

Fitur-fitur yang ada pada dataset ini yaitu:

```
Formatted Date  
Summary  
Precip Type  
Temperature (C)  
Apparent Temperature (C)  
Humidity  
Wind Speed (km/h)  
Wind Bearing (degrees)  
Visibility (km)  
Loud Cover  
Pressure (millibars)  
Daily Summary
```

Gambar 5. Fitur-fitur yang tersedia

MODEL LINEAR REGRESSION

METODE PENELITIAN

Pengumpulan Dataset

Dataset yang digunakan pada penelitian ini diperoleh dari Kaggle Data. Berisi kumpulan 96453 data cuaca yang berbeda-beda dengan 12 fitur.

Data-data di ambil dari tahun 2006-2016. Pada gambar dibawah, disajikan visualisasi dataset yang di gunakan untuk penelitian ini.

	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	Wind Bearing (degrees)	Visibility (km)	Loud Cover	Pressure (millibars)
count	96453.000000	96453.000000	96453.000000	96453.000000	96453.000000	96453.000000	96453.0	96453.000000
mean	11.932678	10.855029	0.734899	10.810640	187.509232	10.347325	0.0	1003.235956
std	9.551546	10.696847	0.195473	6.913571	107.383428	4.192123	0.0	116.969906
min	-21.822222	-27.716667	0.000000	0.000000	0.000000	0.000000	0.0	0.000000
25%	4.688889	2.311111	0.600000	5.828200	116.000000	8.339800	0.0	1011.900000
50%	12.000000	12.000000	0.780000	9.965900	180.000000	10.046400	0.0	1016.450000
75%	18.838889	18.838889	0.890000	14.135800	290.000000	14.812000	0.0	1021.090000
max	39.905556	39.344444	1.000000	63.852600	359.000000	16.100000	0.0	1046.380000

Gambar 6. Dataset Cuaca di Szeged Tahun 2006-2016



METODE PENELITIAN

Perhitungan Performa

Pada tahapan ini, dilakukan proses perhitungan metrik performa, yang meliputi nilai output y , Sum of Square Errors (SSE), Mean Squared Error (MSE), dan Coefficient of Determination (R^2).

Gambar 5 menunjukkan perhitungan nilai output y . Ini merupakan hasil perhitungan dari penjumlahan bias.

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n$$

Gambar 7. Perhitungan Nilai Output y

Gambar 6 menunjukkan rumus dari perhitungan dari sum of square errors (SSE) dengan cara mengurangi nilai y ke- i dengan nilai output y di kuadratkan.

$$SSE = \sum_{i=1}^m (y_i - (\beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \dots + \beta_n x_{ni}))^2$$

for $i = 1, 2, \dots, m$.

Gambar 8. perhitungan dari sum of square errors (SSE)

METODE PENELITIAN

Perhitungan Performa

Gambar 7 menunjukkan rumus dari perhitungan dari Mean Squared Error (MSE) dengan cara membagi hasil SSE dengan nilai m.

$$MSE = \frac{SSE}{m} = \frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2$$

Gambar 9. Perhitungan dari Mean Squared Error (MSE)

Gambar 8 menunjukkan rumus dari perhitungan dari Coefficient of Determination (R^2) dengan cara SSR dibagi dengan SST. Koefisien determinasi adalah bagian dari total variasi dalam variabel dependen (y) yang dijelaskan oleh variasi dalam variabel independen (x).

$$R^2 = \frac{SSR}{SST} = \frac{\sum_{i=1}^m (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^m (y_i - \bar{y})^2}$$

Gambar 10. Perhitungan dari Coefficient of Determination (R^2)

HASIL DAN PEMBAHASAN

Pada gambar 9 dan 10 disajikan hasil dari nilai y dan Sum of Square Errors (SSE) dengan berbagai variasi output yang telah diuji cobakan. Dari gambar 6 akan menunjukkan seberapa melenceng nilainya. Bagian ini akan menunjukkan angka yang diprediksi dikurangi dengan nilai tes sebenarnya untuk semua poin data.

```
pred = lr.predict(np.array(Xtest).reshape(-1,1))
```

```
pred
```

```
array([18.64653318, 24.41087388, 19.92543835, ..., 22.97133327,  
       28.47618597, 14.4452988 ])
```

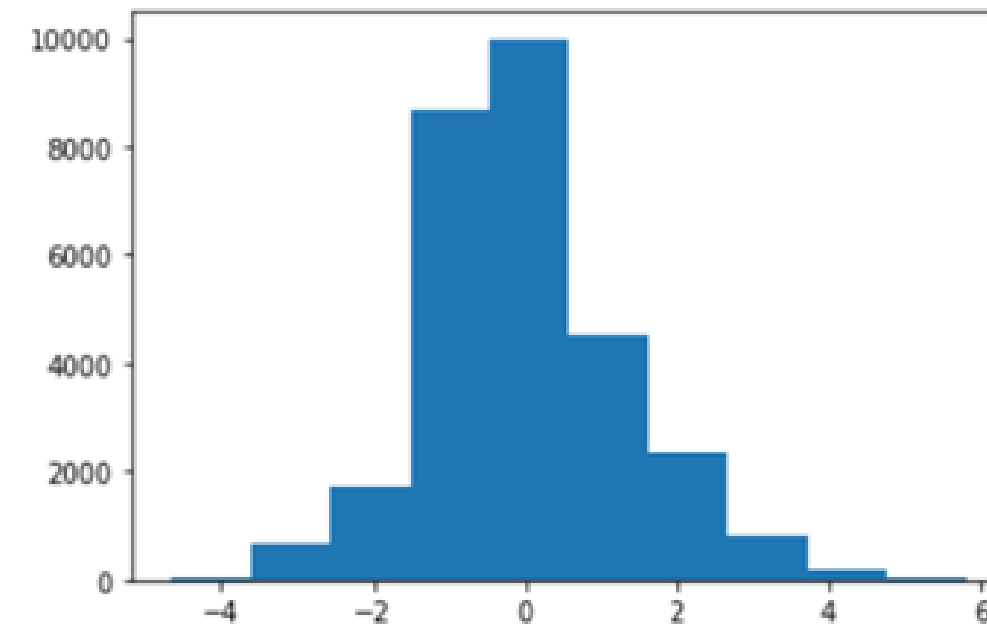
Gambar 11. Nilai Output y



HASIL DAN PEMBAHASAN

Sedangkan pada gambar 10 menampilkan data visualisasi menggunakan histogram untuk melihat seberapa "melenceng" per nilainya. Graf dibawah menunjukkan distribusi error. Kebanyakan, nilai yang muncul kurang lebih mendekati 0.

```
#Residual = Observed value - Predicted value  $e = y - \hat{y}$ .  
residuals = pred - ytest  
plt.hist(residuals)  
  
(array([ 45., 649., 1727., 8665., 9990., 4502., 2369., 788., 168.,  
        33.]),  
 array([-4.65707237, -3.61042429, -2.56377621, -1.51712812, -0.47048004,  
        0.57616804, 1.62281613, 2.66946421, 3.71611229, 4.76276038,  
        5.80940846]),  
 <BarContainer object of 10 artists>)
```



Gambar 12. Visualisasi hasil Sum of Square Errors (SSE)



HASIL DAN PEMBAHASAN

Pada gambar 11 disajikan hasil dari Mean Squared Error (MSE)

```
mse = mean_squared_error(ytest, pred)
print("MSE: ", mse)
```

```
MSE: 1.685483333321293
```

Gambar 13. Hasil Mean Squared Error (MSE)

HASIL DAN PEMBAHASAN

Kemudian pada gambar 12 menunjukkan hasil dari Coefficient of Determination (R^2)

```
Xtest2d = Xtest.values.reshape(-1,1)
pred2d = pred.reshape(-1,1)
r2 = lr.score(Xtest2d, pred2d)
print("R^2: ", r2)
```

```
R^2: 1.0
```

Gambar 14. Hasil Coefficient of Determination (R^2)

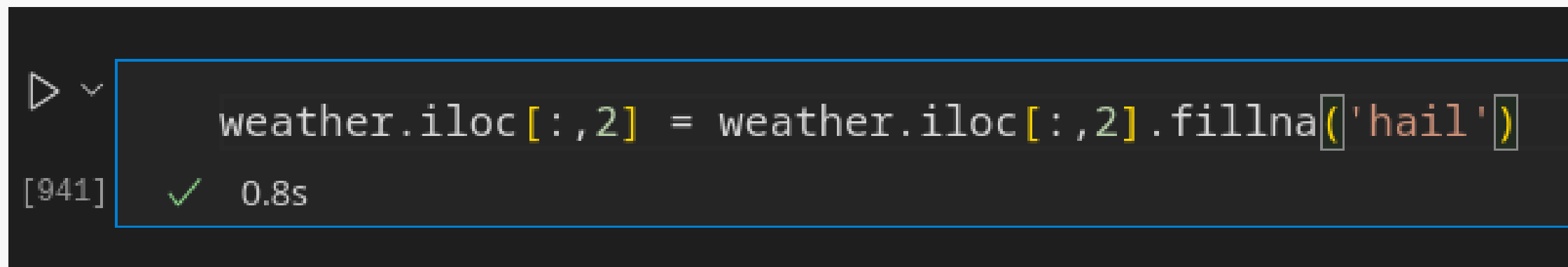
MODEL GAUSSIAN NAIVE BAYES

METODE PENELITIAN

Pengumpulan Dataset dan Preprocessing

Dataset yang digunakan pada penelitian ini diperoleh dari Kaggle Data. Berisi kumpulan 96453 data cuaca yang berbeda-beda dengan 12 fitur.

Agar dapat digunakan pada model naive bayes ini, dataset harus dibersihkan dari nilai yang mengandung NaN/null, pada kasus ini, kolom "Percip Type" memiliki beberapa nilai null.



```
weather.iloc[:,2] = weather.iloc[:,2].fillna('hail')
```

[941] ✓ 0.8s

Gambar 15. Dataset Cuaca di Szeged Tahun 2006-2016 yang telah digantikan nilai null



METODE PENELITIAN

Preprocessing lanjutan

Agar dapat digunakan pada model naive bayes ini, dataset harus dibersihkan dari segala fitur yang tidak dibutuhkan sehingga terbentuk seperti yang ditunjuk pada gambar 14

```
filtered_weather = weather.drop(['Formatted Date', 'Summary', 'Daily Summary', 'Loud Cover', 'Precip Type'],axis=1)
filtered_weather.head()
# print(filtered_weather)
```

✓ 0.1s

	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	Wind Bearing (degrees)	Visibility (km)	Pressure (millibars)
0	9.472222	7.388889	0.89	14.1197	251.0	15.8263	1015.13
1	9.355556	7.227778	0.86	14.2646	259.0	15.8263	1015.63
2	9.377778	9.377778	0.89	3.9284	204.0	14.9569	1015.94
3	8.288889	5.944444	0.83	14.1036	269.0	15.8263	1016.41
4	8.755556	6.977778	0.83	11.0446	259.0	15.8263	1016.51

Gambar 16. Dataset Cuaca di Szeged Tahun 2006-2016 yang telah di filter



METODE PENELITIAN

Preprocessing lanjutan

Pada tahap ini, akan dihitung probabilitas terpilihnya suatu keputusan satu variabel memiliki nilai tertentu. Jika ingin menentukan probabilitas total maka harus dikalikan hasil probabilitas semua variabelnya

$$P(x_i | y = c_k) = \frac{1}{\sqrt{2\pi}\sigma_{c_k}} \exp\left(-\frac{(x_i - \mu_{c_k})^2}{2\sigma_{c_k}^2}\right)$$

Gambar 17. Rumus Gaussian Naive Bayes

PADA VARIABEL PERCIPITATION TYPE

HASIL DAN PEMBAHASAN

Pada pengategorian variabel Percipitation Type, kita akan mengodekan setiap nilai string yang ada menjadi integer sehingga bisa dikenali dan diproses oleh library Gaussian Naive Bayes

```
from sklearn.model_selection import train_test_split
import pandas as pd
weathernp = filtered_weather.to_numpy()
percipnp = weather['Precip Type'].to_numpy()
print(percipnp)
unique, codedpercipnp = np.unique(percipnp, return_inverse=True)
print(codedpercipnp)
```

[979] ✓ 0.9s

```
... ['rain' 'rain' 'rain' ... 'rain' 'rain' 'rain']
     [1 1 1 ... 1 1 1]
```

Gambar 18. Mengodifikasi Percipitation Type sehingga berbentuk angka

HASIL DAN PEMBAHASAN

Setelah melakukan pengkodean dan membagi data mana saja yang menjadi test dan train, maka akan diproses perhitungannya menggunakan `gnb.fit` dan `gnb.predict`. Kemudian akan dihitung keakurasian prediksi tersebut

```
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score
gnb = GaussianNB()
gnb.fit(xtrain,ytrain)
ypred = gnb.predict(xtest)
acc = 100*accuracy_score(ytest,ypred)
print(f"Accuracy= {round(acc,3)}%")
```

[1046] ✓ 0.8s

... Accuracy= 92.962%

Gambar 18. Menghitung Gaussian Naive Bayes dan mengevaluasi keakurasiannya

HASIL DAN PEMBAHASAN

Kemudian, cari hubungan nilai antara variabel-variabel yang ada dengan Percipitation Type untuk mengetahui seberapa berat suatu variabel mempengaruhi hasil.

```
for i in filtered_weather.columns:
    print(f'Correlation between {i} and Precip Type class: {np.corrcoef(filtered_weather[i], codedpercipnp)[0,1]}')
[1047] ✓ 0.5s
```

... Correlation between Temperature (C) and Precip Type class: -0.5422453613181669
Correlation between Apparent Temperature (C) and Precip Type class: -0.5452639564269441
Correlation between Humidity and Precip Type class: 0.22389594188257944
Correlation between Wind Speed (km/h) and Precip Type class: -0.06831941192287515
Correlation between Wind Bearing (degrees) and Precip Type class: -0.040390018975976556
Correlation between Visibility (km) and Precip Type class: -0.29199925464572113
Correlation between Pressure (millibars) and Precip Type class: 0.006472496786503377

Gambar 18. Menghitung korelasi antara semua variable dengan Percipitation Type

PADA VARIABEL SUMMARY

Prosesnya mirip dengan Percipitation Type

HASIL DAN PEMBAHASAN

```
from sklearn.model_selection import train_test_split
import pandas as pd
weathernp = filtered_weather.to_numpy()
sumnp = weather['Summary'].to_numpy()
print(sumnp)
unique, codedsumnp = np.unique(sumnp, return_inverse=True)
print(codedsumnp)
```

[1081] ✓ 0.1s

```
... ['Partly Cloudy' 'Partly Cloudy' 'Mostly Cloudy' ... 'Partly Cloudy'
     'Partly Cloudy' 'Partly Cloudy']
[19 19 17 ... 19 19 19]
```

Gambar 19. Mengodifikasi Summary sehingga berbentuk angka

HASIL DAN PEMBAHASAN

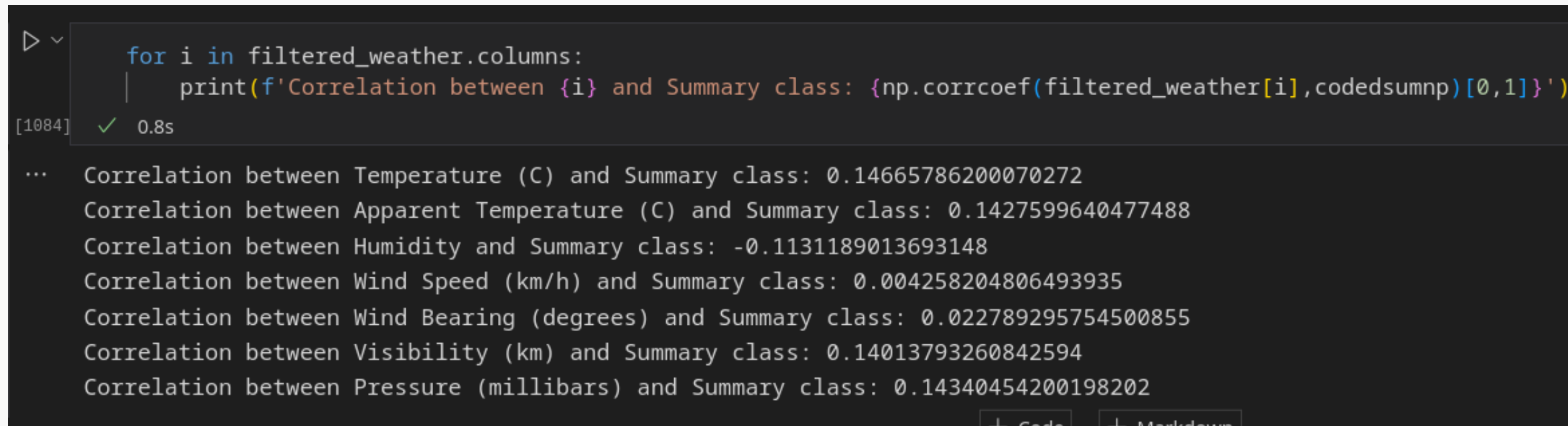
```
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score
gnb = GaussianNB()
gnb.fit(xtrain,ytrain)
ypred = gnb.predict(xtest)
acc = 100*accuracy_score(ytest,ypred)
print(f"Accuracy= {round(acc,3)}%")
```

[1083] ✓ 0.1s

... Accuracy= 44.44%

Gambar 20. Menghitung Gaussian Naive Bayes dan mengevaluasi keakurasiannya

HASIL DAN PEMBAHASAN



```
for i in filtered_weather.columns:
    print(f'Correlation between {i} and Summary class: {np.corrcoef(filtered_weather[i], codedsumnp)[0,1]}')
```

[1084] ✓ 0.8s

... Correlation between Temperature (C) and Summary class: 0.14665786200070272
Correlation between Apparent Temperature (C) and Summary class: 0.1427599640477488
Correlation between Humidity and Summary class: -0.1131189013693148
Correlation between Wind Speed (km/h) and Summary class: 0.004258204806493935
Correlation between Wind Bearing (degrees) and Summary class: 0.022789295754500855
Correlation between Visibility (km) and Summary class: 0.14013793260842594
Correlation between Pressure (millibars) and Summary class: 0.14340454200198202

Gambar 18. Menghitung korelasi antara semua variable dengan Summary



Kesimpulan

Hasil analisis menunjukkan bahwa ada hubungan yang signifikan antara variabel cuaca dengan variabel independen lainnya, sehingga dapat digunakan untuk memprediksi perubahan cuaca di Szeged. Kemudian dari pengujian dataset dengan menggunakan model Regresi Linier mendapatkan hasil yang sempurna pada Coefficient of Determination (R^2) = 1.0 sedangkan pada Mean Squared Error (MSE) = 1.68548. Selain menggunakan model Regresi Linear, prediksi cuaca dapat dilakukan menggunakan model Gaussian Naive Bayes. Pada variabel precipitation type, dapat diprediksi secara akurat akan memiliki jenis hujan yang seperti apa. Namun, pada variabel Summary, tidak dapat diprediksi secara akurat dikarenakan banyaknya value yang ada untuk model yang terbatas.

Dengan demikian model regresi linier dan gaussian naive bayes menggunakan variabel precipitation type cukup akurat memprediksi perubahan cuaca di Szeged.



Terima Kasih.

Apakah ada pertanyaan?

