Laporan Tugas Preprocessing Dataset Weather Type Classification

Dosen Pengampu : Nelly Indriani W,S.Si., M.T.

Disusun Oleh :

Nama : Farhan Nawwafal Pramudia.

Prodi : Teknik Informatika.

NIM : 10123470.

Matakuliah : Pembelajaran Mesin.

TEKNIK INFORMATIKA

UNIVERSITAS KOMPUTER INDONESIA

PROVINSI JAWA BARAT

TAHUN 2025

BAB I

Dataset yang Digunakan

Pada tugas kali ini, saya menggunakan dataset dari open data, yaitu Kaggle. Saya menggunakan dataset yang berjudul *Weather Type Classification*. Data ini memiliki 13.200 baris dan 11 kolom. *Weather Type Classification* merupakan sebuah dataset yang dibuat untuk meniru data cuaca, yang bertujuan untuk tugas klasifikasi. Dalam kasus ini, di masa depan, saya akan melakukan klasifikasi untuk menentukan apakah cuaca esok hari termasuk ke dalam salah satu 4 kategori dari tipe cuaca yang mana.

Data ini memiliki 11 kolom, berikut kolom beserta penjelasannya:

Kolom	**Keterangan**
Temperature	Data suhu udara rata-rata yang memiliki satuan derajat celcius
Humidity	Data kelembaban yang memiliki satuan persen
Wind Speed	Data kecepatan angin dalam satuan kilometer per jam
Precipitation	Data pengendapan yang memiliki satuan persen
Cloud Cover	Data yang menjelaskan cuaca
Atmospheric Pressure	Data yang menjelaskan tekanan atmosfir yang memiliki satuan hPa
UV Index	Data sinar UV
Season	Data yang menjelaskan musim dari sebuah cuaca
Visibility	Data yang menjelaskan seberapa jauh objek bisa dilihat. Artinya, semakin jauh, kabut semakin tipis
Location	Data yang menjelaskan di mana cuaca direkam

Weather Type	Data yang menjelaskan tipe dari cuaca. Ada 4 tipe: Sunny, Rainy, Cloudy dan Snowy

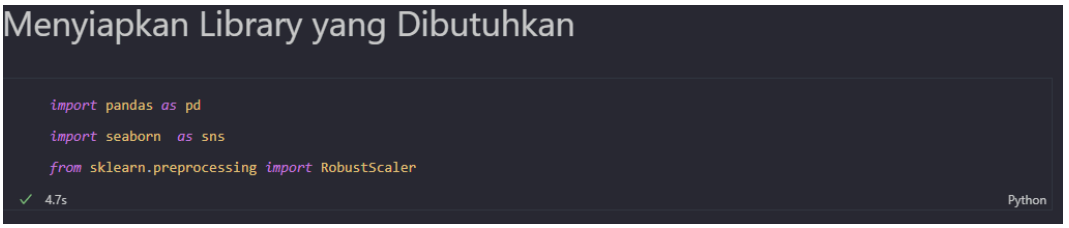
Karena data ini bukan data asli, yang merepresentasikan cuaca di dunia nyata, karena data ini digunakan untuk tugas klasifikasi yang dilakukan oleh seorang data scientist, mahasiswa atau siswa yang masih dikategorikan sebagai pemula.

BAB II

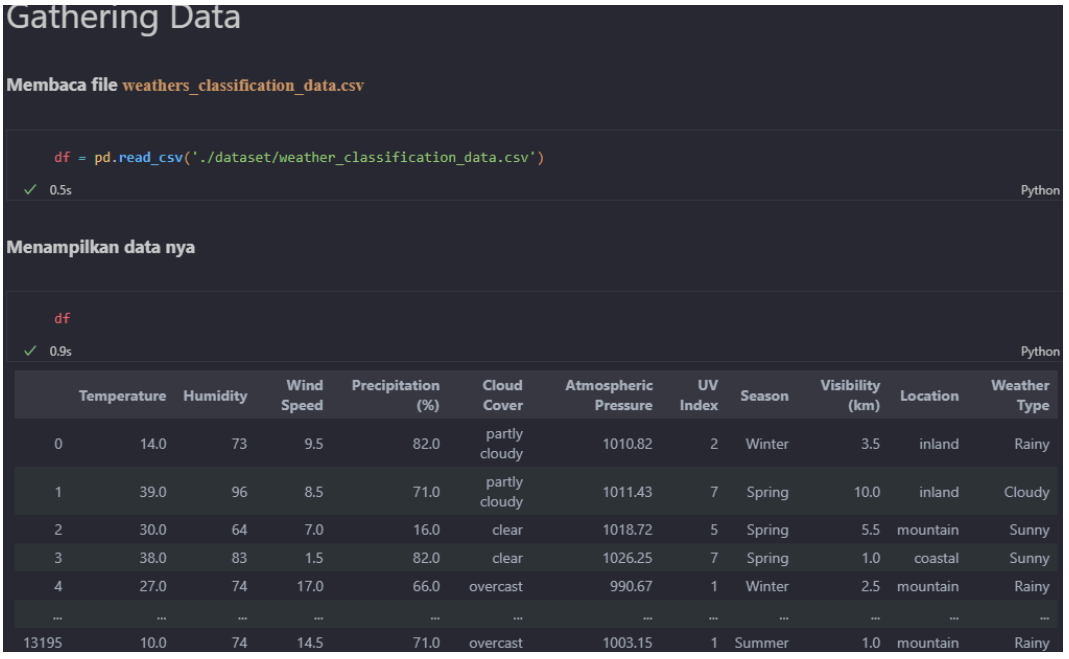
Exploratory Data Analysis

Sebelum melakukan *preprocessing*, saya melakukan hal yang penting terlebih dahulu dalam proses *preprocessing*, yaitu tahapan yang dikenal sebagai EDA atau *Exploratory Data Analysis*. Di sini saya melakukan beberapa hal sebelum melakukan proses pembersihan data:

1. Menyiapkan Library yang Dibutuhkan: Semua pustaka yang dibutuhkan dalam proses *preprocessing*.



2. Gathering Data: Membaca dan menampilkan data yang saya gunakan.



	Temperature	Humidity	Wind Speed	Precipitation (%)	Cloud Cover	Atmospheric Pressure	UV Index	Season	Visibility (km)	Location	Weather Type
0	14.0	73	9.5	82.0	partly cloudy	1010.82	2	Winter	3.5	inland	Rainy
1	39.0	96	8.5	71.0	partly cloudy	1011.43	7	Spring	10.0	inland	Cloudy
2	30.0	64	7.0	16.0	clear	1018.72	5	Spring	5.5	mountain	Sunny
3	38.0	83	1.5	82.0	clear	1026.25	7	Spring	1.0	coastal	Sunny
4	27.0	74	17.0	66.0	overcast	990.67	1	Winter	2.5	mountain	Rainy
...	...	...	...	...	...	...	...	...	...	...	...
13195	10.0	74	14.5	71.0	overcast	1003.15	1	Summer	1.0	mountain	Rainy

3. Melihat Struktur Data dari DataFrame

Melihat Struktur Data dari dataframe (df)

```
df.info()
```

✓ 0.4s Python

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13200 entries, 0 to 13199
Data columns (total 11 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Temperature          13200 non-null  float64
 1   Humidity             13200 non-null  int64  
 2   Wind Speed           13200 non-null  float64
 3   Precipitation (%)    13200 non-null  float64
 4   Cloud Cover          13200 non-null  object 
 5   Atmospheric Pressure 13200 non-null  float64
 6   UV Index             13200 non-null  int64  
 7   Season               13200 non-null  object 
 8   Visibility (km)      13200 non-null  float64
 9   Location             13200 non-null  object 
 10  Weather Type         13200 non-null  object 
dtypes: float64(5), int64(2), object(4)
memory usage: 1.1+ MB
```

Insight: struktur data dari dataframe, semuanya sudah benar. Tidak ada tipe data yang kurang tepat.

4. Mengecek Descriptive Statistics

Cek Descriptive Statistics

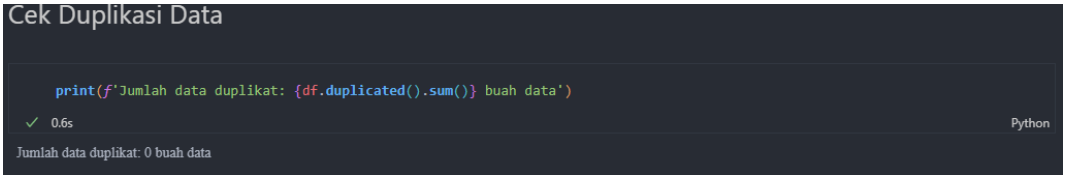
```
df.describe()
```

✓ 0.6s Python

	Temperature	Humidity	Wind Speed	Precipitation (%)	Atmospheric Pressure	UV Index	Visibility (km)
count	13200.000000	13200.000000	13200.000000	13200.000000	13200.000000	13200.000000	13200.000000
mean	19.127576	68.710833	9.832197	53.644394	1005.827896	4.005758	5.462917
std	17.386327	20.194248	6.908704	31.946541	37.199589	3.856600	3.371499
min	-25.000000	20.000000	0.000000	0.000000	800.120000	0.000000	0.000000
25%	4.000000	57.000000	5.000000	19.000000	994.800000	1.000000	3.000000
50%	21.000000	70.000000	9.000000	58.000000	1007.650000	3.000000	5.000000
75%	31.000000	84.000000	13.500000	82.000000	1016.772500	7.000000	7.500000
max	109.000000	109.000000	48.500000	109.000000	1199.210000	14.000000	20.000000

5. Mengecek Missing Values dalam Data

6. Mengecek Duplikasi Data



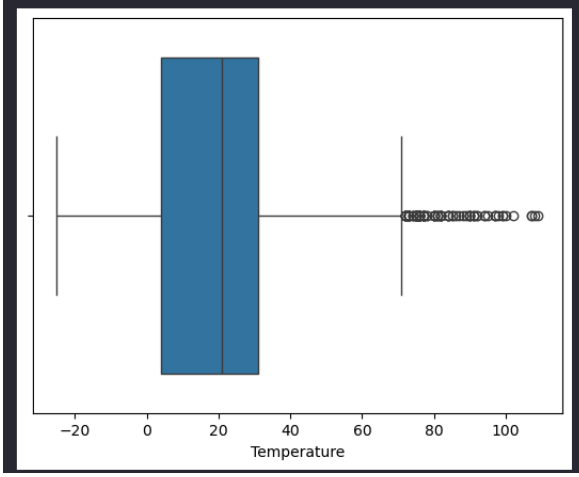
7. Mengecek Outlier dari Semua Kolom Numerik

Cek outlier kolom Temperature

```
sns.boxplot(data=df, x=df['Temperature'])
```

✓ 0.2s

<Axes: xlabel='Temperature'>



Insight:

- Letak median lebih dekat ke Q3. Hal ini berarti distribusi cenderung condong ke kanan atau **right-skewed**
- Banyak outlier di sisi kanan (> 70 derajat celcius) yang cukup aneh jika diamati dengan konteks data, yaitu temperatur di dunia nyata. Hal ini bisa jadi karena error pencatatan atau data ekstrem yang harus ditangani
- Rentang suhu IQR berada di sekitar 0 - 35 derajat celcius, hal ini masih wajar

```
df['Temperature'].skew()
```

✓ 0.4s

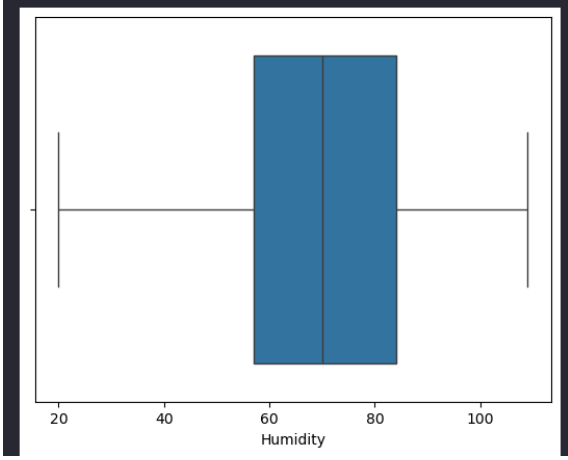
0.2217414467117672

Cek outlier kolom Humidity

```
sns.boxplot(data=df, x=df['Humidity'])
```

✓ 0.2s Python

<Axes: xlabel='Humidity'>



Insight:

- Median atau Q2 hampir terletak di tengah, artinya distribusi pada kolom Humidity ini hampir simetris
- Tidak ada outlier, artinya data nya stabil dan konsisten

```
df['Humidity'].skew()
```

✓ 0.3s Python

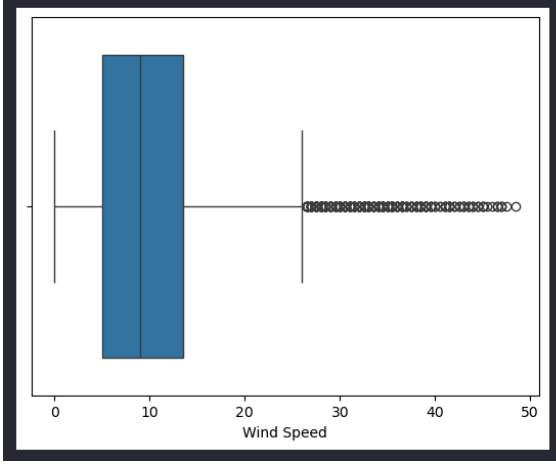
-0.40161426558981855

Cek outlier kolom Wind Speed

```
sns.boxplot(data=df, x=df['Wind Speed'])
```

✓ 0.2s Python

<Axes: xlabel='Wind Speed'>



Insight:

- Median terlihat berada di tengah-tengah (garis hitam di dalam box plot). Artinya distribusi ini simetris
- Banyak outlier saat wind speed > 25 km/jam. Ini ada kemungkinan terjadi badai atau data nya error
- Sebagian besar angin, berada di bawah 20 km/jam

```
df['Wind Speed'].skew()
```

✓ 0.3s Python

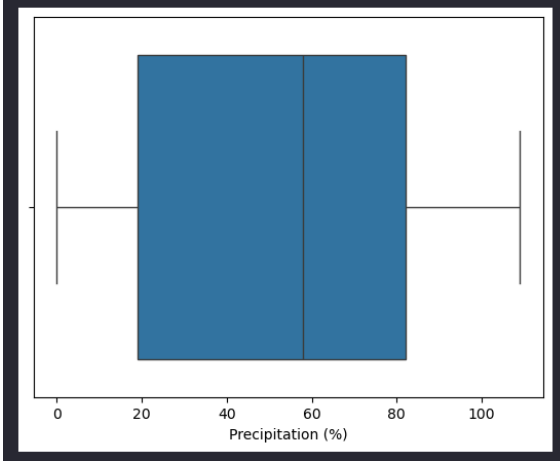
```
1.3602625756285232
```

Cek outlier kolom precipitation

```
sns.boxplot(data=df, x=df['Precipitation (%)'])
```

✓ 0.1s Python

```
<Axes: xlabel='Precipitation (%)'>
```



Insight:

- Median atau Q2 berdekatan dengan Q3
- Tidak ada outlier, artinya data curah hujan cukup stabil

```
df['Precipitation (%)'].skew()
```

✓ 0.3s Python

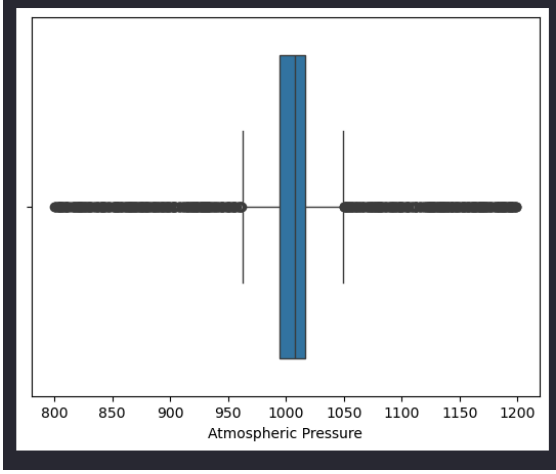
```
-0.15245706717664612
```

Cek outlier Atmoshperic Pressure

```
sns.boxplot(data=df, x=df['Atmospheric Pressure'])
```

✓ 0.2s Python

```
<Axes: xlabel='Atmospheric Pressure'>
```



Insight:

- Median berada hampir di tengah box, yang artinya distribusi ini hampir simetris atau seimbang
- Banyak outlier di sisi kiri dan kanan. Kemungkinan ini sensor error

```
df['Atmospheric Pressure'].skew()
```

✓ 0.4s Python

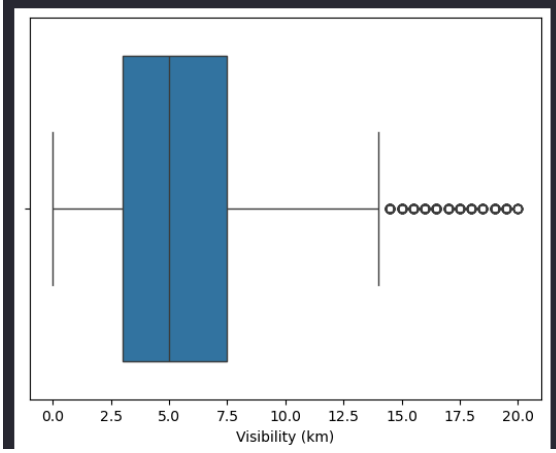
```
-0.2938986063675234
```

Cek outlier kolom Visibility

```
sns.boxplot(data=df, x=df['Visibility (km)'])
```

✓ 0.3s Python

```
<Axes: xlabel='Visibility (km)'>
```



Insight:

- Median hampir terletak di tengah-tengah, artinya distribusi ini hampir simetris
- Ada beberapa outlier ketika visibilitynya di atas (> 13) 13 km
- Mayoritas visibility nya berada di 2-8 km

```
df['Visibility (km)'].skew()
```

✓ 0.4s Python

```
1.2332751645049822
```

BAB III

Preprocessing

Setelah proses EDA selesai dilakukan, maka tahapan ini hanya perlu membersihkan atau meminimalisir outlier yang tampil pada tahap EDA sebelumnya. Dalam kasus ini, saya melakukan proses transformasi menggunakan pustaka *scikit-learn* yang mengambil modul *RobustScaler* untuk transformasi datanya. Berikut adalah hasil *preprocessingnya*:

Copy Data Asli

```
df_new = df.copy()
df_new
```

✓ 0.9s Python

	Temperature	Humidity	Wind Speed	Precipitation (%)	Cloud Cover	Atmospheric Pressure	UV Index	Season	Visibility (km)	Location	Weather Type
0	14.0	73	9.5	82.0	partly cloudy	1010.82	2	Winter	3.5	inland	Rainy
1	39.0	96	8.5	71.0	partly cloudy	1011.43	7	Spring	10.0	inland	Cloudy
2	30.0	64	7.0	16.0	clear	1018.72	5	Spring	5.5	mountain	Sunny
3	38.0	83	1.5	82.0	clear	1026.25	7	Spring	1.0	coastal	Sunny
4	27.0	74	17.0	66.0	overcast	990.67	1	Winter	2.5	mountain	Rainy
...	...	...	...	...	...	...	...	...	...	...	...
13195	10.0	74	14.5	71.0	overcast	1003.15	1	Summer	1.0	mountain	Rainy
13196	-1.0	76	3.5	23.0	cloudy	1067.23	1	Winter	6.0	coastal	Snowy
13197	30.0	77	5.5	28.0	overcast	1012.69	3	Autumn	9.0	coastal	Cloudy
13198	3.0	76	10.0	94.0	overcast	984.27	0	Winter	2.0	inland	Snowy
13199	-5.0	38	0.0	92.0	overcast	1015.37	5	Autumn	10.0	mountain	Rainy

13200 rows × 11 columns

Transformasi Data

```
scaler = RobustScaler()
```

✓ 0.5s Python

```
cols_to_scale = df_new.select_dtypes(include=['number']).columns
cols_to_scale
```

✓ 0.6s Python

```
Index(['Temperature', 'Humidity', 'Wind Speed', 'Precipitation (%)',
       'Atmospheric Pressure', 'UV Index', 'Visibility (km)'],
      dtype='object')
```

```
df_new[cols_to_scale] = scaler.fit_transform(df_new[cols_to_scale])
df_new
```

✓ 0.1s Python

	Temperature	Humidity	Wind Speed	Precipitation (%)	Cloud Cover	Atmospheric Pressure	UV Index	Season	Visibility (km)	Location	Weather Type
0	-0.259259	0.111111	0.058824	0.380952	partly cloudy	0.144271	-0.166667	Winter	-0.333333	inland	Rainy
1	0.666667	0.962963	-0.058824	0.206349	partly cloudy	0.172033	0.666667	Spring	1.111111	inland	Cloudy
2	0.333333	-0.222222	-0.235294	-0.666667	clear	0.503812	0.333333	Spring	0.111111	mountain	Sunny
3	0.629630	0.481481	-0.882353	0.380952	clear	0.846513	0.666667	Spring	-0.888889	coastal	Sunny
4	0.222222	0.148148	0.941176	0.126984	overcast	-0.772784	-0.333333	Winter	-0.555556	mountain	Rainy
...	...	...	...	...	...	...	...	...	...	...	...
13195	-0.407407	0.148148	0.647059	0.206349	overcast	-0.204801	-0.333333	Summer	-0.888889	mountain	Rainy
13196	-0.814815	0.222222	-0.647059	-0.555556	cloudy	2.711571	-0.333333	Winter	0.222222	coastal	Snowy
13197	0.333333	0.259259	-0.411765	-0.476190	overcast	0.229378	0.000000	Autumn	0.888889	coastal	Cloudy
13198	-0.666667	0.222222	0.117647	0.571429	overcast	-1.064057	-0.500000	Winter	-0.666667	inland	Snowy
13199	-0.962963	-1.185185	-1.058824	0.539683	overcast	0.351348	0.333333	Autumn	1.111111	mountain	Rainy

13200 rows × 11 columns

```
df.describe()
```

✓ 0.8s Python

	Temperature	Humidity	Wind Speed	Precipitation (%)	Atmospheric Pressure	UV Index	Visibility (km)
count	13200.000000	13200.000000	13200.000000	13200.000000	13200.000000	13200.000000	13200.000000
mean	19.127576	68.710833	9.832197	53.644394	1005.827896	4.005758	5.462917
std	17.386327	20.194248	6.908704	31.946541	37.199589	3.856600	3.371499
min	-25.000000	20.000000	0.000000	0.000000	800.120000	0.000000	0.000000
25%	4.000000	57.000000	5.000000	19.000000	994.800000	1.000000	3.000000
50%	21.000000	70.000000	9.000000	58.000000	1007.650000	3.000000	5.000000
75%	31.000000	84.000000	13.500000	82.000000	1016.772500	7.000000	7.500000
max	109.000000	109.000000	48.500000	109.000000	1199.210000	14.000000	20.000000

Insight: Descriptive statistics dari data weather_classification_data sebelum dipreprocessing

```
df_new.describe()
```

✓ 0.9s Python

	Temperature	Humidity	Wind Speed	Precipitation (%)	Atmospheric Pressure	UV Index	Visibility (km)
count	13200.000000	13200.000000	13200.000000	13200.000000	13200.000000	13200.000000	13200.000000
mean	-0.069349	-0.047747	0.097906	-0.069137	-0.082927	0.167626	0.102870
std	0.643938	0.747935	0.812789	0.507088	1.693007	0.642767	0.749222
min	-1.703704	-1.851852	-1.058824	-0.920635	-9.444988	-0.500000	-1.111111
25%	-0.629630	-0.481481	-0.470588	-0.619048	-0.584822	-0.333333	-0.444444
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	0.370370	0.518519	0.529412	0.380952	0.415178	0.666667	0.555556
max	3.259259	1.444444	4.647059	0.809524	8.718170	1.833333	3.333333

Insight: Descriptive statistics dari data weather_classification_data sesudah dipreprocessing

BAB IV

Hasil Akhir

Berikut adalah screenshot tampilan hasil akhir setelah datanya di *preprocessing*:

```
df_new.head()
```

✓ 0.7s Python

	Temperature	Humidity	Wind Speed	Precipitation (%)	Cloud Cover	Atmospheric Pressure	UV Index	Season	Visibility (km)	Location	Weather Type
0	-0.259259	0.111111	0.058824	0.380952	partly cloudy	0.144271	-0.166667	Winter	-0.333333	inland	Rainy
1	0.666667	0.962963	-0.058824	0.206349	partly cloudy	0.172033	0.666667	Spring	1.111111	inland	Cloudy
2	0.333333	-0.222222	-0.235294	-0.666667	clear	0.503812	0.333333	Spring	0.111111	mountain	Sunny
3	0.629630	0.481481	-0.882353	0.380952	clear	0.846513	0.666667	Spring	-0.888889	coastal	Sunny
4	0.222222	0.148148	0.941176	0.126984	overcast	-0.772784	-0.333333	Winter	-0.555556	mountain	Rainy

```
df_new.tail()
```

✓ 0.8s Python

	Temperature	Humidity	Wind Speed	Precipitation (%)	Cloud Cover	Atmospheric Pressure	UV Index	Season	Visibility (km)	Location	Weather Type
13195	-0.407407	0.148148	0.647059	0.206349	overcast	-0.204801	-0.333333	Summer	-0.888889	mountain	Rainy
13196	-0.814815	0.222222	-0.647059	-0.555556	cloudy	2.711571	-0.333333	Winter	0.222222	coastal	Snowy
13197	0.333333	0.259259	-0.411765	-0.476190	overcast	0.229378	0.000000	Autumn	0.888889	coastal	Cloudy
13198	-0.666667	0.222222	0.117647	0.571429	overcast	-1.064057	-0.500000	Winter	-0.666667	inland	Snowy
13199	-0.962963	-1.185185	-1.058824	0.539683	overcast	0.351348	0.333333	Autumn	1.111111	mountain	Rainy

Link Program dan Data

1. Link program: https://github.com/Farhan-Nawwafal/pembelajaran-mesin/blob/master/tugas/pertemuan-3/preprocessing-pm.ipynb
2. Link data: https://www.kaggle.com/datasets/nikhil7280/weather-type-classification