

Data Science Project 3

Weather Prediction using Machine Learning Algorithm

Table of contents

01

Business
Problems

02

Data
Understanding

03

Exploratory Data
Analysis

04

Data
Preprocessing

05

Data Modelling

06

Model Evaluation
& Conclusions

01

Business Problems

Short-Term Variability

Prediksi cuaca dalam beberapa hari ke depan seringkali melibatkan variasi cuaca yang cepat dan kompleks, sehingga memerlukan model yang dapat menangani perubahan cepat.

Dengan memanfaatkan Ridge Regression sebagai model analisis cuaca diharapkan dapat membantu meningkatkan akurasi prediksi cuaca dalam jangka pendek.

- Berdasarkan model yang digunakan tersebut, apakah ada perbedaan antara temperatur aktual dengan temperatur hasil prediksi?
- Bagaimana perkiraan temperatur prediksi yang dihasilkan oleh model tersebut?
- Variabel apa yang memiliki pengaruh terhadap suhu maksimum target?



[This Photo](#) by Unknown Author is licensed under [CC BY-ND](#)

02

Data Understanding

Dataset Information

Dataset diperoleh dari: National Centers for Environmental Information (NOAA).

- Data cuaca diambil dari **Oakland International Airport, CA US**. Pemilihan data ini di bandara karena bandara memiliki sensor suhu dan pengamatan cuaca yang baik.
- Rentang data pengukuran : 01 Januari 1960 hingga 05 Februari 2022.
- Terdapat 16859 baris data dan 35 kolom.

```
#Menampilkan beberapa karakteristik data menggunakan info()
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 16859 entries, 1960-01-01 to 2022-01-28
Data columns (total 35 columns):
 #   Column    Non-Null Count  Dtype  
--- 
 0   STATION   16859 non-null   object 
 1   NAME      16859 non-null   object 
 2   ACMH      5844 non-null   float64 
 3   ACSH      5844 non-null   float64 
 4   AWND      8051 non-null   float64 
 5   DAPR      8 non-null     float64 
 6   FMTM      2190 non-null   float64 
 7   FRGT      2 non-null     float64 
 8   MDPR      8 non-null     float64 
 9   PGTM      8512 non-null   float64 
 10  PRCP      16578 non-null  float64 
 11  SNOW      11380 non-null  float64 
 12  SNWD      11584 non-null  float64 
 13  TAVG      2037 non-null   float64 
 14  TMAX      16850 non-null  float64 
 15  TMIN      16849 non-null  float64 
 16  TSUN      1151 non-null   float64 
 17  WDF1      5844 non-null   float64 
 18  WDF2      8052 non-null   float64 
 19  WDF5      7965 non-null   float64 
 20  WDFG      4267 non-null   float64 
 21  WSF1      5844 non-null   float64 
 22  WSF2      8053 non-null   float64 
 23  WSF5      7965 non-null   float64 
 24  WSFG      4267 non-null   float64 
 25  WT01      3710 non-null   float64 
 26  WT02      333 non-null    float64 
 27  WT03      119 non-null    float64 
 28  WT04      4 non-null     float64 
 29  WT05      28 non-null    float64 
 30  WT07      2 non-null     float64 
 31  WT08      3197 non-null   float64 
 32  WT09      2 non-null     float64 
 33  WT16      1955 non-null   float64 
 34  WT18      3 non-null     float64 
dtypes: float64(33), object(2)
memory usage: 4.6+ MB
```

03

Exploratory Data Analysis

Data Preparation

1. Handling missing values

- Dari output yang diberikan, terlihat bahwa terdapat beberapa nilai yang hilang (missing values) pada dataset. Kolom (PCRP, SNOW, SNWD, TMAX, dan TMIN) memiliki jumlah missing value yang relatif kecil dibandingkan kolom-kolom lainnya.
- Berdasarkan Global Historical Climatology Network (GHCND) documentation. Terdapat 5 variabel utama yang menjadi fokus atau prioritas dalam analisis. Sehingga, kelima kolom tersebut dipilih untuk memenuhi kebutuhan analisis.

```
#Handling missing values pada dataset  
df.isna().sum().sort_values()
```

```
STATION      0  
NAME         0  
TMAX         9  
TMIN        10  
PRCP        281  
SNWD       5355  
SNOW        5479  
PGTM        8347  
WSF2        8806  
WDF2        8807  
AWND        8808  
WSF5        8894  
WDF5        8894  
WSF1       11015  
WDF1       11015  
ACSH       11015  
ACMH       11015  
WSFG        12592  
WDFG        12592  
WT01       13149  
WT08       13662  
FMTM       14669  
TAVG       14822  
WT16       14904  
TSUN       15708  
WT02       16526  
WT03       16740  
WT05       16831  
MDPR       16851  
DAPR       16851  
WT04       16855  
WT18       16856  
WT07       16857  
FRGT       16857  
WT09       16857  
dtype: int64
```

Mengambil 5 core variable atau main variable

```
#Taking 5 core variable atau main variable  
df1 = df[['PRCP', 'SNOW', 'SNWD', 'TMAX', 'TMIN']].copy()
```

Penjelasan setiap kolomnya:

PRCP (Precipitation)	Pengukuran curah hujan (inci)
SNOW (Snowfall)	Pengukuran salju yang turun (inci)
SNWD (Snow Depth)	Pengukuran ketebalan salju di permukaan tanah (inci)
TMAX (Maximum Temperature)	Pencatatan suhu tertinggi yang dicapai (Fahrenheit)
TMIN (Minimum Temperature)	Pencatatan suhu terendah yang dicapai (Fahrenheit)

2. Mengubah nama kolom

```
df1.head()
```

	PRCP	SNOW	SNWD	TMAX	TMIN
DATE					
1960-01-01	0.0	0.0	0.0	49.0	30.0
1960-01-02	0.0	0.0	0.0	49.0	29.0
1960-01-03	0.0	0.0	0.0	54.0	35.0
1960-01-04	0.0	0.0	0.0	54.0	36.0
1960-01-05	0.0	0.0	0.0	55.0	33.0



```
#Change the column name
```

```
df1.columns =['precipitation', 'snowfall', 'snow_depth', 'temp_max', 'temp_min']
```

```
df1
```

	precipitation	snowfall	snow_depth	temp_max	temp_min
DATE					
1960-01-01	0.0	0.0	0.0	49.0	30.0
1960-01-02	0.0	0.0	0.0	49.0	29.0
1960-01-03	0.0	0.0	0.0	54.0	35.0
1960-01-04	0.0	0.0	0.0	54.0	36.0
1960-01-05	0.0	0.0	0.0	55.0	33.0
...
2022-01-24	0.0	NaN	NaN	60.0	39.0
2022-01-25	0.0	NaN	NaN	57.0	43.0
2022-01-26	0.0	NaN	NaN	57.0	41.0
2022-01-27	0.0	NaN	NaN	67.0	39.0
2022-01-28	0.0	NaN	NaN	64.0	39.0

16859 rows × 5 columns

Hal ini dilakukan untuk memberikan deskripsi lebih jelas.

3. Mengecek missing values

```
#Calculate the number of missing values (NaN or null values)
df1.isna().sum()
```

```
precipitation      281
snowfall          5479
snow_depth        5355
temp_max           9
temp_min          10
dtype: int64
```

Terdapat missing value untuk kelima kolom tersebut.

A. Kolom 'snowfall'

```
#Count the unique values in the 'snowfall' column of the DataFrame df1
df1['snowfall'].value_counts()
```

```
0.0    11379
1.0      1
Name: snowfall, dtype: int64
```

Kolom 'snowfall' menunjukkan sebagian besar data memiliki nilai 'snowfall' sebesar 0.0 dan hanya terdapat satu kejadian dengan nilai 'snowfall' sebesar 1. Oleh karena itu, **dilakukan penghapusan pada kolom 'snowfall'**.

B. Kolom 'snow_dept'

```
#Count the unique values in the 'snow_dept' column of the DataFrame df1
df1['snow_depth'].value_counts()
```

Kolom 'snow_depth' menunjukkan bahwa semua nilai pada kolom 'snow_depth' adalah 0.0 dan tidak ada unique value lain. Oleh karena itu, **kolom 'snow_depth' dihapus**.

C. Kolom 'precipitation'

```
#Count the number of missing values
df1['precipitation'].isna().sum()
```

```
281
```

```
#Fill missing value (NaN or null) in the 'precipitation' column with 0
df1['precipitation'] = df1['precipitation'].fillna(0)
```

Terdapat 281 nilai yang hilang (NaN) dalam kolom 'precipitation'. Sehingga, **nilai yang hilang diisi dengan nilai 0**.

D. Kolom 'temp_min'

```
#Count the number of missing values  
df1['temp_min'].isna().sum()  
  
10
```

Terdapat 10 nilai yang hilang (NaN) dalam kolom 'temp_min'.

E. Kolom 'temp_max'

```
#Count the number of missing values  
df1['temp_max'].isna().sum()  
  
9
```

Terdapat 9 nilai yang hilang (NaN) dalam kolom 'temp_max'.

Mengisi missing value untuk kedua kolom tersebut

Metode ffill digunakan karena 'temp_max' dan 'temp_min' berhubungan dengan data time-series dimana nilai-nilai seringkali berkorelasi secara temporal.

```
#Fill missing values (NaN) using the forward-fill method  
df1 = df1.fillna(method='ffill')
```

```
df1.isna().sum()
```

```
precipitation      0  
temp_max           0  
temp_min           0  
dtype: int64
```

Tidak terdapat missing value lagi pada ketiga kolom tersebut.

4. Mengubah index pada DataFrame

```
df1.index
```

```
Index(['1960-01-01', '1960-01-02', '1960-01-03', '1960-01-04', '1960-01-05',
       '1960-01-06', '1960-01-07', '1960-01-08', '1960-01-09', '1960-01-10',
       ...
       '2022-01-19', '2022-01-20', '2022-01-21', '2022-01-22', '2022-01-23',
       '2022-01-24', '2022-01-25', '2022-01-26', '2022-01-27', '2022-01-28'],
      dtype='object', name='DATE', length=16859)
```



```
#Converting the index of the DataFrame 'df1' to the datetime data type.  
df1.index = pd.to_datetime(df1.index)
```

```
df1.index
```

```
DatetimeIndex(['1960-01-01', '1960-01-02', '1960-01-03', '1960-01-04',
                  '1960-01-05', '1960-01-06', '1960-01-07', '1960-01-08',
                  '1960-01-09', '1960-01-10',
                  ...
                  '2022-01-19', '2022-01-20', '2022-01-21', '2022-01-22',
                  '2022-01-23', '2022-01-24', '2022-01-25', '2022-01-26',
                  '2022-01-27', '2022-01-28'],
                 dtype='datetime64[ns]', name='DATE', length=16859, freq=None)
```

5. Mengecek kolom dalam dataset

- Pada dokumentasi GHCND (Global Historical Climatology Network - Daily), terdapat catatan yang menyebutkan "Note: 9's in a field (e.g., 9999) indicate missing data or data that has not been received."
- Ini berarti jika suatu kolom dalam dataset memiliki nilai 9 (contoh: 9999), maka hal ini menunjukkan bahwa ketidadaan data atau ketidaktersediaan data.

```
#Make sure that columns don't have 9999
#Check if x equals 9999
df1.apply(lambda x: (x==9999).sum())
```

```
precipitation      0
temp_max          0
temp_min          0
dtype: int64
```

Hasilnya menunjukkan bahwa tidak ada nilai 9999 dalam kolom-kolom tersebut.

Informasi terbaru dataset

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 16859 entries, 1960-01-01 to 2022-01-28
Data columns (total 3 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   precipitation    16859 non-null   float64
 1   temp_max         16859 non-null   float64
 2   temp_min         16859 non-null   float64
dtypes: float64(3)
memory usage: 526.8 KB
```

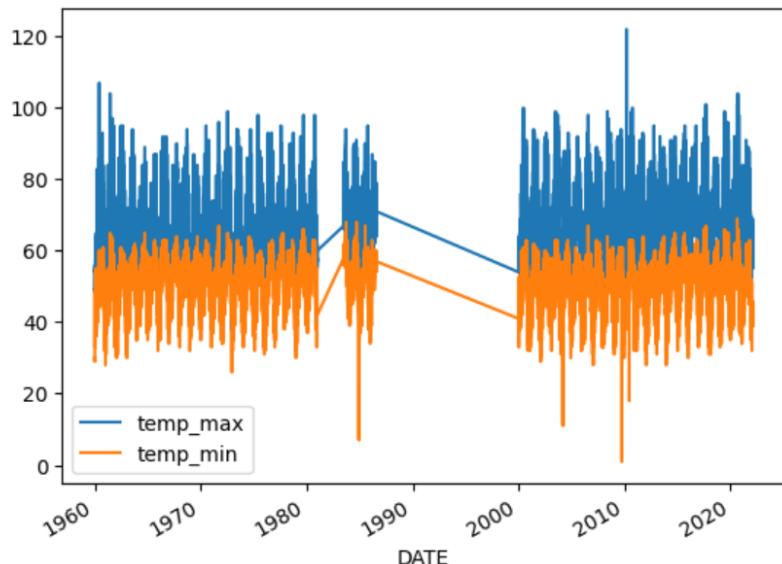
Tidak terdapat missing values.

Data Visualization

1. Line plot 'temp_max' dan 'temp_min'

```
#Create a line plot for 'temp_max' and 'temp_min'  
df1[['temp_max','temp_min']].plot()
```

```
<Axes: xlabel='DATE'>
```



- Pada grafik tersebut terdapat beberapa gaps, adanya gaps di sekitar tahun 1983 hingga 1985 dan 1990 hingga 1999.
- Hal ini mungkin terjadi karena perubahan pada stasiun cuaca di bandara tersebut.

```
#Cheking which years of data are missing  
df1.index.year.value_counts().sort_index()
```

```
1960    366  
1961    365  
1962    365  
1963    365  
1964    366  
1965    365  
1966    365  
1967    365  
1968    366  
1969    365  
1970    365  
1971    365  
1972    366  
1973    365  
1974    365  
1975    365  
1976    366  
1977    365  
1978    365  
1979    365  
1980    366  
1983    184  
1984    366  
1985    365  
1986    212  
2000    365  
2001    365
```

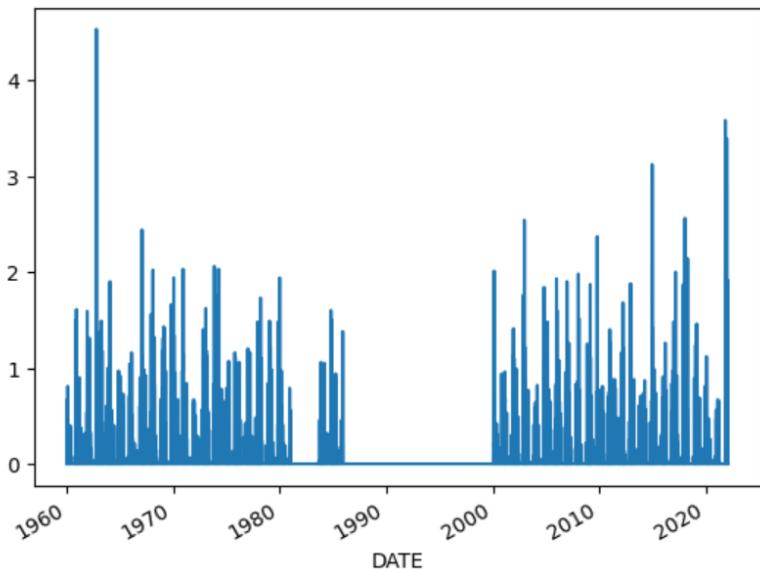
...

Terdapat tahun yang hilang pada data tersebut yaitu
1981 hingga 1982 dan **1987 hingga 1999**.

2. Line plot ‘precipitation’

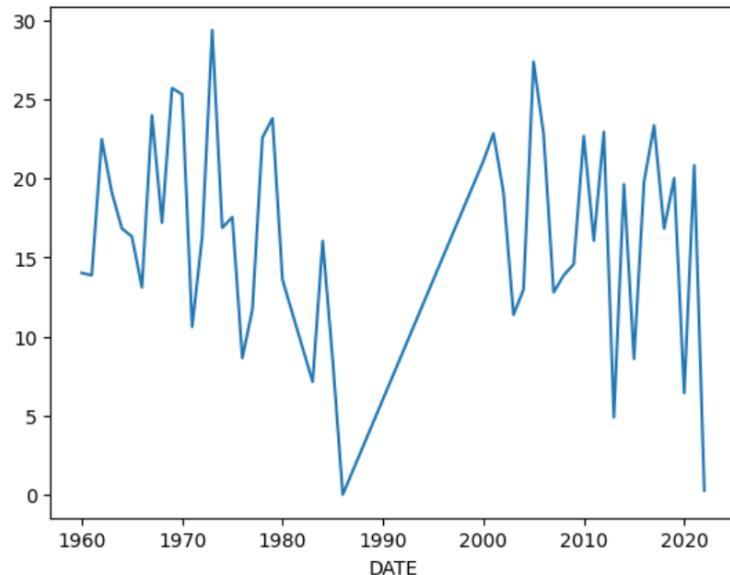
```
#Plot out the precipitation column  
df1['precipitation'].plot()
```

```
<Axes: xlabel='DATE'>
```



```
df1.groupby(df1.index.year).apply(lambda x: x["precipitation"].sum()).plot()
```

```
<Axes: xlabel='DATE'>
```



04

Data Pre-Processing

1. Penambahan fitur baru ‘target’

- Penambahan fitur baru ‘target’ ini dilakukan untuk memprediksi temperatur pada waktu berikutnya berdasarkan variabel prediktor waktu sebelumnya.
- Hal ini dilakukan untuk mengevaluasi sejauh mana fitur-fitur lain dapat memberikan informasi untuk memprediksi target.

```
df1['target'] = df1.shift(-1)['temp_max']
```

- Terdapat nilai yang hilang (missing value) di akhir data karena semua nilai di setiap kolom digeser ke atas satu baris.
- Sehingga baris terakhir akan menyebabkan null values. Oleh karena itu, **dilakukan penghapusan baris pada tanggal terakhir (2022-01-28) tersebut.**

DATE	precipitation	temp_max	temp_min	target
1960-01-01	0.0	49.0	30.0	49.0
1960-01-02	0.0	49.0	29.0	54.0
1960-01-03	0.0	54.0	35.0	54.0
1960-01-04	0.0	54.0	36.0	55.0
1960-01-05	0.0	55.0	33.0	53.0
...
2022-01-24	0.0	60.0	39.0	57.0
2022-01-25	0.0	57.0	43.0	57.0
2022-01-26	0.0	57.0	41.0	67.0
2022-01-27	0.0	67.0	39.0	64.0
2022-01-28	0.0	64.0	39.0	NaN

16859 rows × 4 columns

Metode Ridge Regression

- **Metode Ridge Regression** digunakan untuk mengatasi masalah overfitting dalam model regresi.
- Ridge regression menambahkan istilah regulasi ke fungsi tujuan regresi untuk mencegah koefisien model menjadi terlalu besar.

```
#Use ridge regression which is a type of regression that minimizes overfitting
reg = Ridge(alpha=.1)
```

- Penggunaan alpha menunjukkan seberapa besar efek regulasi yang diberikan pada model. **Semakin besar nilai alpha, maka semakin kuat regulasi yang diterapkan.**
- Parameter alpha (α) digunakan dalam teknik regulasi untuk membantu mencegah overfitting dalam memberikan hukuman atau penalizing terhadap nilai absolut dari koefisien model regresi.

2. Variabel prediktor

	Variabel prediktor	Variabel target
DATE	precipitation temp_max temp_min	target
1960-01-01	0.0 49.0 30.0	49.0
1960-01-02	0.0 49.0 29.0	54.0
1960-01-03	0.0 54.0 35.0	54.0
1960-01-04	0.0 54.0 36.0	55.0
1960-01-05	0.0 55.0 33.0	53.0
...
2022-01-23	0.0 60.0 41.0	60.0
2022-01-24	0.0 60.0 39.0	57.0
2022-01-25	0.0 57.0 43.0	57.0
2022-01-26	0.0 57.0 41.0	67.0
2022-01-27	0.0 67.0 39.0	64.0

16858 rows × 4 columns

- **Variabel predictor** adalah variabel ‘precipitation’, ‘temp_max’, dan ‘temp_min’ saat ini.
- **Variabel target** adalah variabel ‘temp_max’ hari berikutnya.

3. Membagi dataset menjadi train dan test set

```
#Create new DataFrame 'train' from the beginning up to December 31, 2020.  
train = df1.loc[:'2020-12-31']
```

train

	precipitation	temp_max	temp_min	target
DATE				
1960-01-01	0.00	49.0	30.0	49.0
1960-01-02	0.00	49.0	29.0	54.0
1960-01-03	0.00	54.0	35.0	54.0
1960-01-04	0.00	54.0	36.0	55.0
1960-01-05	0.00	55.0	33.0	53.0
...
2020-12-27	0.00	63.0	44.0	61.0
2020-12-28	0.10	61.0	42.0	60.0
2020-12-29	0.00	60.0	39.0	56.0
2020-12-30	0.07	56.0	36.0	62.0
2020-12-31	0.06	62.0	44.0	60.0

16467 rows × 4 columns

```
#Create new DataFrame 'test' from the January 1, 2021 to the end of the DataFrame.  
test = df1.loc['2021-01-01':]
```

test

	precipitation	temp_max	temp_min	target
DATE				
2021-01-01	0.00	60.0	40.0	57.0
2021-01-02	0.14	57.0	51.0	56.0
2021-01-03	0.00	56.0	49.0	62.0
2021-01-04	0.36	62.0	46.0	59.0
2021-01-05	0.00	59.0	42.0	59.0
...
2022-01-23	0.00	60.0	41.0	60.0
2022-01-24	0.00	60.0	39.0	57.0
2022-01-25	0.00	57.0	43.0	57.0
2022-01-26	0.00	57.0	41.0	67.0
2022-01-27	0.00	67.0	39.0	64.0

391 rows × 4 columns

05

Data Modelling

```
#Used to train (fit) a regression model using the training data.
```

```
reg.fit(train[predictors], train['target']) Pelatihan model regresi menggunakan data 'train'
```

```
    ▾ Ridge
```

```
Ridge(alpha=0.1)
```

Model dilatih untuk membuat prediksi pada data uji 'test'

```
predictions = reg.predict(test[predictors]) Hasil prediksi disimpan dalam variabel 'predictions'
```

```
#Calculate the mean absolute error (MAE) between the actual target values (test['target']) and the predicted values (predictions).  
mean_absolute_error(test['target'],predictions)
```

```
3.4111699434528306
```

- **MAE** adalah metrik evaluasi yang mengukur rata-rata dari selisih absolut antara nilai aktual dan nilai prediksi.
- Semakin rendah nilai MAE, semakin baik kinerja model dalam membuat prediksi yang akurat.
- Nilai Mean Absolute Error (MAE) sebesar 3.4 menunjukkan menunjukkan rata-rata kesalahan absolut antara nilai aktual dan nilai prediksi pada data uji.

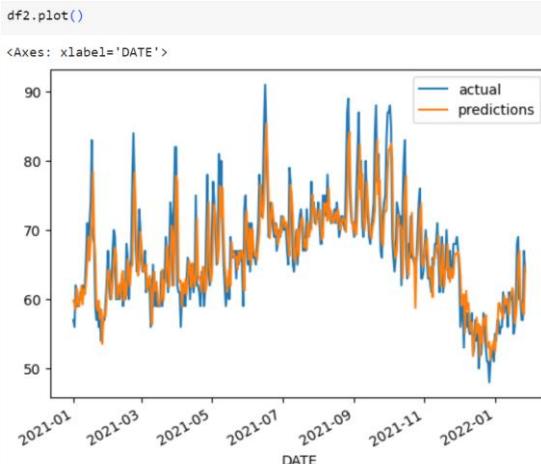
06

Model Evaluation

```
df2 = pd.concat([test['target'], pd.Series(predictions, index=test.index)], axis=1)
df2.columns = ['actual', 'predictions']
```

	actual	predictions
DATE		
2021-01-01	57.0	59.806024
2021-01-02	56.0	59.310181
2021-01-03	62.0	58.538685
2021-01-04	59.0	61.531814
2021-01-05	59.0	59.444266
...
2022-01-23	60.0	59.985714
2022-01-24	57.0	59.626333
2022-01-25	57.0	58.181680
2022-01-26	67.0	57.822299
2022-01-27	64.0	64.674302

391 rows × 2 columns



- Pada grafik tersebut menunjukkan bahwa nilai puncak temperatur yang terjadi dalam keadaan aktual jauh lebih tinggi dibandingkan dengan nilai yang diprediksi oleh model.
- Model prediksi tidak mencapai tingkat rendah temperatur aktual. Ini mungkin terjadi karena model tidak dapat mengidentifikasi atau memprediksi penurunan temperatur yang tajam.
- Prediksi yang dihasilkan oleh model cenderung lebih konservatif dimana model cenderung memberikan perkiraan yang sedikit lebih rendah untuk nilai maksimum dan sedikit lebih tinggi untuk nilai minimum temperatur.

Koefisien regresi yang dihasilkan oleh model regresi:

```
reg.coef_  
  
array([-2.20730384,  0.72113834,  0.17969047])  
  
predictors = ['precipitation', 'temp_max', 'temp_min']
```

- **Koefisien negatif:** variabel tersebut akan berkontribusi pada **penurunan** nilai target.
- **Koefisien positif:** variabel akan berkontribusi pada **peningkatan** nilai target.

Precipitation (precipitation)

Koefisien: -2.21

Nilai koefisien negatif menunjukkan bahwa curah hujan memiliki korelasi negatif pada temperature. Artinya, jika terjadi hujan, model mengidikasikan bahwa temperatur besok kemungkinan akan rendah.

Maximum Temperature (temp_max)

Koefisien: 0.72

Nilai koefisien positif yang cukup besar menunjukkan bahwa suhu maksimum hari sebelumnya memiliki pengaruh pada suhu maksimum besok.

Artinya, suhu maksimum sebelumnya adalah faktor utama yang memengaruhi suhu maksimum besok.

Minimum Temperature (temp_min)

Koefisien: 0.18

Nilai koefisien positif yang lebih kecil menunjukkan bahwa suhu minimum hari sebelumnya juga memiliki korelasi positif pada suhu maksimum besok, meskipun pengaruhnya lebih kecil dibandingkan dengan suhu maksimum.

1. Create a function to make predictions

Membuat prediksi menggunakan model regresi pada data ‘test’ dan kemudian menghitung mean absolute error (MAE) dari prediksi tersebut.

```
def create_predictions(predictors, df1, reg):
    train = df1.loc[:'2020-12-31']
    test = df1.loc['2021-01-01':]
    reg.fit(train[predictors], train['target'])
    predictions = reg.predict(test[predictors])
    error = mean_absolute_error(test['target'], predictions)
    df2 = pd.concat([test['target'], pd.Series(predictions, index=test.index)], axis=1)
    df2.columns = ['actual', 'predictions']
    return error, df2
```



2. Create a rolling average

Berapa nilai rata-rata suhu maksimum bulan ini dengan mempertimbangkan suhu bulan sebelumnya?

Nilai di setiap baris ‘month_max’ adalah rata-rata suhu maksimum harian dalam rentang waktu 30 hari.

```
df1['month_max'] = df1['temp_max'].rolling(30).mean()
```

Nilai di setiap baris ‘month_day_max’ adalah rasio antara rata-rata suhu maksimum harian dalam rentang waktu 30 hari dan suhu maksimum harian aktual. Rasio ini memberikan informasi tentang seberapa tinggi dengan rata-rata suhu maksimum dalam beberapa hari terakhir.

```
#Find some interesting ratios  
df1['month_day_max'] = df1['month_max']/df1['temp_max']
```

Rasio ini memberikan informasi tentang perbandingan antara suhu maksimum dan minimum harian.

```
#Ratio between the maximum temperature and the minimum temperature  
df1['max_min'] = df1['temp_max']/df1['temp_min']
```

	precipitation	temp_max	temp_min	target	month_max	month_day_max	max_min	monthly_avg	day_of_year_avg	Variabel prediktor
DATE										
1960-02-07	0.06	62.0	55.0	60.0	57.233333	0.923118	1.127273	60.714286	62.000000	
1960-02-08	0.51	60.0	50.0	55.0	57.533333	0.958889	1.200000	60.625000	60.000000	
1960-02-09	0.36	55.0	48.0	60.0	57.533333	1.046061	1.145833	60.000000	55.000000	
1960-02-10	0.07	60.0	44.0	61.0	57.766667	0.962778	1.363636	60.000000	60.000000	
1960-02-11	0.00	61.0	40.0	61.0	58.033333	0.951366	1.525000	60.090909	61.000000	
...
2022-01-16	0.00	61.0	46.0	60.0	55.266667	0.906011	1.326087	56.759915	57.543478	
2022-01-17	0.00	60.0	44.0	55.0	55.466667	0.924444	1.363636	56.762208	57.413043	
2022-01-18	0.00	55.0	42.0	56.0	55.433333	1.007879	1.309524	56.760962	57.695652	

- Jika nilai 'month_day_max' > 1 menunjukkan bahwa suhu harian aktual cenderung lebih tinggi daripada rata-rata dalam beberapa hari terakhir. Sedangkan jika < 1 menunjukkan kecenderungan sebaliknya.
- Jika nilai 'max_min' lebih tinggi menunjukkan bahwa perbedaan antara suhu maksimum dan minimum harian pada hari tersebut lebih besar. dan jika nilai 'max_min' lebih rendah menunjukkan bahwa perbedaan antara suhu maksimum dan minimum harian pada hari tersebut lebih kecil.

```
error, df2 = create_predictions (predictors, df1, reg)
```

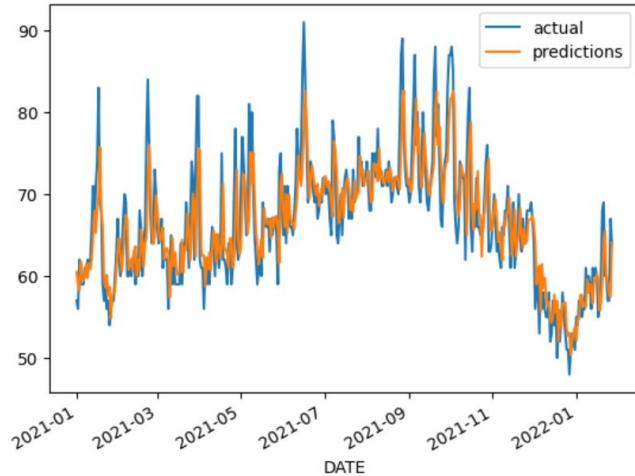
```
error
```

```
3.360129746207605
```

- Sebelumnya, MAE adalah 3.4, dan setelah dilakukan beberapa penambahan fitur atau perubahan lainnya pada model, MAE berubah menjadi 3.36.
- MAE sekitar 3.36 derajat Fahrenheit, yang berarti rata-rata kesalahan prediksi suhu maksimum harian adalah sekitar 3.36 derajat Fahrenheit.

```
df2.plot()
```

```
<Axes: xlabel='DATE'>
```



3. Adding in monthly and daily averages

Penambahan fitur ini diharapkan dapat meningkatkan kemampuan model untuk menangkap pola suhu untuk berbagai skala waktu.

```
# Add more predictors - Monthly Average for 'temp_max'  
df1["monthly_avg"] = df1.groupby(df1.index.month, group_keys=False)[['temp_max']].apply(lambda x: x.expanding(1).mean())
```

Rata-rata suhu maksimum 'temp_max' setiap bulan.

```
df1['day_of_year_avg'] = df1[['temp_max']].groupby(df1.index.day_of_year, group_keys=False).apply(lambda x: x.expanding(1).mean())
```

Mengelompokkan data suhu maksimum berdasarkan hari dalam setahun.

```
predictors = ['precipitation', 'temp_max', 'temp_min', 'month_max', 'month_day_max', 'max_min', 'monthly_avg', 'day_of_year_avg']  
  
error, df2 = create_predictions(predictors, df1, reg)  
  
error  
  
3.3176926587659956
```

Perbedaan antara nilai yang diprediksi oleh model dan nilai aktual yang diamati telah sedikit berkurang. Error telah sedikit berkurang dari 3.36 menjadi 3.31, itu berarti model regresi yang digunakan untuk memprediksi suhu maksimum besok cenderung lebih dekat atau akurat daripada sebelumnya.

4. Running Model Diagnostics

`reg.coef_` adalah array yang berisi koefisien (bobot) yang diterapkan pada setiap prediktor dalam model.

```
reg.coef_  
  
array([-0.90804013,  0.40146278,  0.03114081,  0.33672074,  
      -15.97061869,  0.0493612 ,  0.14343362,  0.08006707])
```

Precipitation

Koefisien: **-0.91**

Koefisien negatif menunjukkan peningkatan curah hujan cenderung dikaitkan dengan penurunan suhu maksimum (`temp_max`).

Artinya, jika curah meningkat suhu maksimum besok kemungkinan akan lebih rendah.

Temperature Maksimum (`temp_max`)

Koefisien: **0.40**

Koefisien positif menunjukkan bahwa suhu maksimum hari ini (`temp_max`) memiliki korelasi positif terhadap prediksi suhu maksimum besok.

Artinya, peningkatan suhu maksimum hari ini mungkin terjadi peningkatan suhu maksimum besok.

Temperature Minimum (`temp_min`)

Koefisien: **0.03**

Koefisien positif kecil menunjukkan bahwa suhu minimum hari ini (`temp_min`) juga memiliki pengaruh positif terhadap prediksi suhu maksimum besok.

Month_max

Koefisien: 0.34

Koefisien positif menunjukkan bahwa suhu bulanan memiliki korelasi positif terhadap suhu maksimum besok.

Month_day_max

Koefisien: -15.97

Koefisien negatif yang besar menunjukkan bahwa rasio antara suhu maksimum bulanan dan suhu maksimum harian memiliki korelasi negatif yang signifikan pada suhu maksimum besok.

Artinya, perubahan besar dalam suhu maksimum dapat menyebabkan penurunan drastis pada suhu maksimum besok.

Max_min

Koefisien: 0.05

Koefisien positif kecil menunjukkan bahwa rasio antara suhu maksimum dengan suhu minimum memiliki pengaruh positif terhadap suhu maksimum besok.

Monthly_avg

Koefisien: 0.14

Koefisien positif menunjukkan bahwa rata-rata suhu maksimum bulanan memiliki pengaruh positif terhadap prediksi suhu maksimum besok.

Day_of_year_avg

Koefisien: 0.08

Koefisien positif menunjukkan bahwa rata-rata suhu maksimum harian memiliki pengaruh positif terhadap suhu maksimum besok.

Korelasi antara kolom target yang mewakili suhu maksimum besok dengan kolom prediktor.

```
df1.corr()['target']
```

```
precipitation      -0.205413
temp_max           0.821650
temp_min           0.596016
target              1.000000
month_max          0.686842
month_day_max     -0.421537
max_min             0.045228
monthly_avg        0.689805
day_of_year_avg    0.712334
Name: target, dtype: float64
```

Besar perbedaan suhu aktual dan prediksi

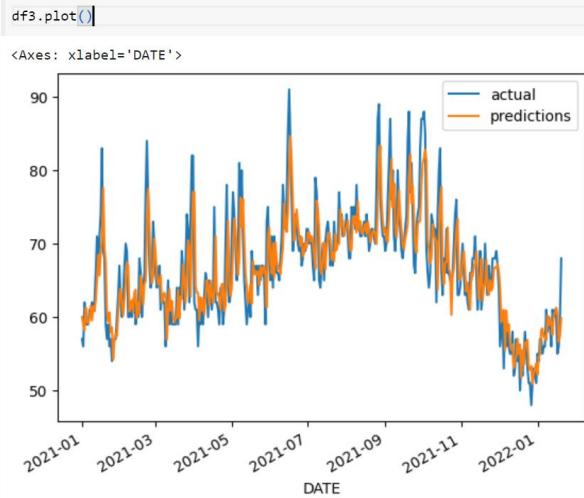
```
df2['diff'] = (df2['actual'] - df2['predictions']).abs()
```

```
df2.sort_values('diff', ascending=False).head()
```

	actual	predictions	diff	
DATE				
2021-01-17	83.0	68.585837	14.414163	
2021-05-07	81.0	67.774432	13.225568	
2021-04-01	62.0	75.178360	13.178360	
2021-02-22	84.0	71.168844	12.831156	
2021-10-16	66.0	78.823197	12.823197	

- **Korelasi positif** yang tinggi (mendekati 1) yaitu ‘temp_max’, ‘month_max’, ‘monthly_avg’, dan ‘day_of_year_avg’ menunjukkan bahwa variabel-variabel tersebut memiliki hubungan positif yang kuat dengan suhu maksimum besok.
- **Korelasi negatif** ‘precipitation’ dan ‘month_day_max’ menunjukkan hubungan negative dengan suhu maksimum besok.

5. Predict weather for the whole next week



- Sebelumnya, MAE adalah 3.4, dan setelah dilakukan beberapa penambahan fitur atau perubahan lainnya pada model, MAE berubah menjadi 3.31
- MAE sekitar 3.31 derajat Fahrenheit, yang berarti rata-rata kesalahan prediksi suhu maksimum harian adalah sekitar 3.31 derajat Fahrenheit.

Korelasi antara kolom target yang mewakili suhu maksimum besok dengan kolom prediktor.

```
df1.corr()['target_week']
```

precipitation	-0.157508
temp_max	0.540127
temp_min	0.565775
target	0.543002
month_max	0.629014
month_day_max	-0.100895
max_min	-0.004257
monthly_avg	0.677243
day_of_year_avg	0.663093
target_week	1.000000
week_max	0.622067
week_day_max	-0.018579
week_max_min	-0.004257
week_avg	0.677234
day_of_year1_avg	0.663046

Name: target_week, dtype: float64

Besar perbedaan suhu aktual dan prediksi

```
df3['diff'] = (df3['actual'] - df3['predictions']).abs()  
  
df3.sort_values('diff', ascending=False).head()
```

	actual	predictions	diff	
DATE				
2021-01-17	83.0	68.813351	14.186649	
2021-04-01	62.0	75.631245	13.631245	
2021-05-07	81.0	68.036892	12.963108	
2021-02-21	77.0	64.549758	12.450242	
2021-02-22	84.0	71.602975	12.397025	

KESIMPULAN

Ridge Regression digunakan untuk melatih model berdasarkan data historis cuaca.

- Temperatur dalam keadaan aktual jauh lebih tinggi dibandingkan dengan nilai yang diprediksi oleh model.
- Temperatur prediksi yang dihasilkan oleh model cenderung lebih konservatif dimana model cenderung memberikan perkiraan yang lebih sedikit rendah untuk nilai maksimum dan sedikit lebih tinggi untuk nilai minimum temperatur.
- Variabel precipitation berkorelasi negatif dengan temperatur sedangkan variabel lainnya berkorelasi positif dengan suhu maksimum besok.

Inne Andarini Herdianti S.Si



Data Science Enthusiast

A bachelor of Science degree in Physics was obtained from the Bandung Institute of Technology. I'm eager to dive into the world of data science. Please check out some of the projects I've worked on my GitHub or LinkedIn. I'm excited about the opportunity to bring my skills and enthusiasm for Data Science!

Contact:



<https://www.linkedin.com/in/inneandarini/>



inneandarinii@gmail.com



<https://github.com/inneandarinii>

Thanks!

CREDITS: This presentation template was created by [Slidesgo](#), and includes icons by [Flaticon](#), and infographics & images by [Freepik](#)