Попередня обробка даних

Попередня підготовка даних не має встановленого переліку операцій; єдина мета полягає в тому, щоб дані після обробки були кориснішими, ніж до неї. На практиці існує три основні задачі, пов’язані з процесом попередньої обробки даних:

- Очищення даних
- Трансформація даних
- Збагачення даних

Попередня обробка даних

Основні завдання очищення даних наступні:

- Перейменування
- Сортування та переупорядкування
- Перетворення типів даних
- Обробка повторюваних даних
- Виправлення відсутніх або недійсних даних
- Фільтрація до потрібної підмножини даних

Попередня обробка даних

Розглянемо наступні дані про погоду:

date	datatype	station	attributes	value
2018-10-01	TAVG	GHCND:USW00014732	H,,S,	21.2
2018-10-01	TMAX	GHCND:USW00014732	,,W,2400	25.6
2018-10-01	TMIN	GHCND:USW00014732	,,W,2400	18.3
2018-10-02	TAVG	GHCND:USW00014732	H,,S,	22.7
2018-10-02	TMAX	GHCND:USW00014732	,,W,2400	26.1

Перейменування:
df.rename(columns={'value': 'temp_C', 'attributes': 'flags'},inplace=True)

date	datatype	station	flags	temp_C
2018-10-01	TAVG	GHCND:USW00014732	H,,S,	21.2
2018-10-01	TMAX	GHCND:USW00014732	,,W,2400	25.6
2018-10-01	TMIN	GHCND:USW00014732	,,W,2400	18.3
2018-10-02	TAVG	GHCND:USW00014732	H,,S,	22.7
2018-10-02	TMAX	GHCND:USW00014732	,,W,2400	26.1

```
df.rename(str.lower,
          axis='columns').columns
```

Попередня обробка даних

Сортування та переупорядкування:

df[df.datatype == 'TMAX'].sort_values(by='temp_C', ascending=False).head(5)

date	datatype	station	flags	temp_C
2018-10-07	TMAX	GHCND:USW00014732	,,W,2400	27.8
2018-10-10	TMAX	GHCND:USW00014732	,,W,2400	27.8
2018-10-11	TMAX	GHCND:USW00014732	,,W,2400	26.7
2018-10-04	TMAX	GHCND:USW00014732	,,W,2400	26.1
2018-10-02	TMAX	GHCND:USW00014732	,,W,2400	26.1

df[df.datatype == 'TMAX'].sort_values(by=['temp_C', 'date'], ascending=[False, True]).head(5)

date	datatype	station	flags	temp_C
2018-10-07	TMAX	GHCND:USW00014732	,,W,2400	27.8
2018-10-10	TMAX	GHCND:USW00014732	,,W,2400	27.8
2018-10-11	TMAX	GHCND:USW00014732	,,W,2400	26.7
2018-10-02	TMAX	GHCND:USW00014732	,,W,2400	26.1
2018-10-04	TMAX	GHCND:USW00014732	,,W,2400	26.1

Попередня обробка даних

df[df.datatype == 'TAVG'].nlargest(n=5, columns='temp_C’)

date	datatype	station	flags	temp_C
2018-10-10	TAVG	GHCND:USW00014732	H,,S,	23.8
2018-10-11	TAVG	GHCND:USW00014732	H,,S,	23.4
2018-10-07	TAVG	GHCND:USW00014732	H,,S,	22.8
2018-10-02	TAVG	GHCND:USW00014732	H,,S,	22.7
2018-10-03	TAVG	GHCND:USW00014732	H,,S,	21.8

df[df.datatype == 'TAVG'].nsmallest(n=5, columns='temp_C')

date	datatype	station	flags	temp_C
2018-10-26	TAVG	GHCND:USW00014732	H,,S,	7.3
2018-10-22	TAVG	GHCND:USW00014732	H,,S,	8.3
2018-10-25	TAVG	GHCND:USW00014732	H,,S,	8.8
2018-10-27	TAVG	GHCND:USW00014732	H,,S,	9.4
2018-10-18	TAVG	GHCND:USW00014732	H,,S,	9.6

Попередня обробка даних

df.sample(5, random_state=0)

date	datatype	station	flags	temp_C
2018-10-01	TMIN	GHCND:USW00014732	,,W,2400	18.3
2018-10-11	TAVG	GHCND:USW00014732	H,,S,	23.4
2018-10-19	TMAX	GHCND:USW00014732	,,W,2400	17.2
2018-10-06	TMAX	GHCND:USW00014732	,,W,2400	21.1
2018-10-05	TMAX	GHCND:USW00014732	,,W,2400	22.8

df.sample(5, random_state=0).sort_index()

date	datatype	station	flags	temp_C
2018-10-01	TMIN	GHCND:USW00014732	,,W,2400	18.3
2018-10-05	TMAX	GHCND:USW00014732	,,W,2400	22.8
2018-10-06	TMAX	GHCND:USW00014732	,,W,2400	21.1
2018-10-11	TAVG	GHCND:USW00014732	H,,S,	23.4
2018-10-19	TMAX	GHCND:USW00014732	,,W,2400	17.2

Попередня обробка даних

df.sort_index(axis=1).head()

date	datatype	flags	station	temp_C
2018-10-01	TAVG	H,,S,	GHCND:USW00014732	21.2
2018-10-01	TMAX	,,W,2400	GHCND:USW00014732	25.6
2018-10-01	TMIN	,,W,2400	GHCND:USW00014732	18.3
2018-10-02	TAVG	H,,S,	GHCND:USW00014732	22.7
2018-10-02	TMAX	,,W,2400	GHCND:USW00014732	26.1

Методи sort_index() та sort_values() повертають новий DataFrame, коли потрібно саме замінити старий, то використовують параметр inplace=True

Попередня обробка даних

Перетворення типів даних:

df.dtypes

```
datatype     object
station      object
flags        object
temp_C      float64
dtype: object
```

df.tz_localize('EST’) – вказуємо часовий пояс

date	datatype	station	flags	temp_C
2018-10-01 00:00:00-05:00	TAVG	GHCND:USW00014732	H,,S,	21.2
2018-10-01 00:00:00-05:00	TMAX	GHCND:USW00014732	,,W,2400	25.6

Попередня обробка даних

df.tz_convert('UTC’)

date	datatype	station	flags	temp_C
2018-10-01 05:00:00+00:00	TAVG	GHCND:USW00014732	H,,S,	21.2
2018-10-01 05:00:00+00:00	TMAX	GHCND:USW00014732	,,W,2400	25.6

df.tz_convert('EET’)

date	datatype	station	flags	temp_C
2018-10-01 08:00:00+03:00	TAVG	GHCND:USW00014732	H,,S,	21.2
2018-10-01 08:00:00+03:00	TMAX	GHCND:USW00014732	,,W,2400	25.6

Попередня обробка даних

df.tz_localize(None).to_period('M’)

date	datatype	station	flags	temp_C
2018-10	TAVG	GHCND:USW00014732	H,,S,	21.2
2018-10	TMAX	GHCND:USW00014732	,,W,2400	25.6
2018-10	TMIN	GHCND:USW00014732	,,W,2400	18.3
2018-10	TAVG	GHCND:USW00014732	H,,S,	22.7
2018-10	TMAX	GHCND:USW00014732	,,W,2400	26.1

df_F = df.assign(date=pd.to_datetime(df.date),temp_F=(df.temp_C * 9/5) + 32)

	date	datatype	station	flags	temp_C	temp_F
0	2018-10-01	TAVG	GHCND:USW00014732	H,,S,	21.2	70.16
1	2018-10-01	TMAX	GHCND:USW00014732	,,W,2400	25.6	78.08
2	2018-10-01	TMIN	GHCND:USW00014732	,,W,2400	18.3	64.94
3	2018-10-02	TAVG	GHCND:USW00014732	H,,S,	22.7	72.86
4	2018-10-02	TMAX	GHCND:USW00014732	,,W,2400	26.1	78.98

```
date          datetime64[ns]
datatype              object
station               object
flags                 object
temp_C               float64
temp_F               float64
dtype: object
```

Попередня обробка даних

df = df.assign(temp_C_int=lambda x: x.temp_C.astype('int’))

date	datatype	station	flags	temp_C	temp_C_int
2018-10-01	TAVG	GHCND:USW00014732	H,,S,	21.2	21
2018-10-01	TMAX	GHCND:USW00014732	,,W,2400	25.6	25

В цілому, для перетворення даних в числові існує функція pd.to_numeric().

Попередня обробка даних

Можливо також перетворити дані на категоріальний (тобто якісний) тип даних.

df.assign(station=df.station.astype('category'),datatype=df.datatype.astype('category’))

```
datatype      category
station       category
flags           object
temp_C         float64
temp_C_int       int32
dtype: object
```

df.describe(include='category')

	datatype	station
count	93	93
unique	3	1
top	TAVG	GHCND:USW00014732
freq	31	93

df.describe()

	temp_C	temp_C_int
count	93.000000	93.000000
mean	15.408602	14.956989
std	6.133703	6.084396
min	5.600000	5.000000
25%	10.200000	10.000000
50%	14.400000	14.000000
75%	21.100000	21.000000
max	27.800000	27.000000

Попередня обробка даних

Трансформація даних зосереджується на зміні структури даних, щоб полегшити подальший аналіз; це зазвичай передбачає зміну того, які дані йдуть в рядках, а які – в стовпцях. Більшість даних мають широкий або довгий формат; кожен з цих форматів має свої переваги, і важливо знати, який з них знадобиться для аналізу.

Широкий формат:

Дата	Температура	Вологість	Вітер
2021-10-11	7	70%	1
2021-10-12	8	80%	2

Довгий формат:

Дата	Параметр	Значення
2021-10-11	Температура	7
2021-10-11	Вологість	70%
2021-10-11	Вітер	1
2021-10-12	Температура	8
2021-10-12	Вологість	80%
2021-10-12	Вітер	2

Попередня обробка даних

Транспонувати дані можна за допомогою методу T:
df.head().T

date	2018-10-01	2018-10-01	2018-10-01	2018-10-02	2018-10-02
datatype	TAVG	TMAX	TMIN	TAVG	TMAX
station	GHCND:USW00014732	GHCND:USW00014732	GHCND:USW00014732	GHCND:USW00014732	GHCND:USW00014732
flags	H,,S,	,,W,2400	,,W,2400	H,,S,	,,W,2400
temp_C	21.2	25.6	18.3	22.7	26.1
temp_F	70.16	78.08	64.94	72.86	78.98

wide_df = df.pivot(columns='datatype', values=['temp_C', 'temp_F'])

	temp_C			temp_F		
datatype	TAVG	TMAX	TMIN	TAVG	TMAX	TMIN
date						
2018-10-01	21.2	25.6	18.3	70.16	78.08	64.94
2018-10-02	22.7	26.1	19.4	72.86	78.98	66.92
2018-10-03	21.8	25.0	18.9	71.24	77.00	66.02
2018-10-04	21.3	26.1	17.8	70.34	78.98	64.04
2018-10-05	20.3	22.8	16.1	68.54	73.04	60.98

Попередня обробка даних

wide_df = df.pivot(columns='datatype')

	station			flags			temp_C			temp_F		
datatype	TAVG	TMAX	TMIN	TAVG	TMAX	TMIN	TAVG	TMAX	TMIN	TAVG	TMAX	TMIN
date												
2018-10-01	GHCND:USW00014732	GHCND:USW00014732	GHCND:USW00014732	H,,S,	,,W,2400	,,W,2400	21.2	25.6	18.3	70.16	78.08	64.94
2018-10-02	GHCND:USW00014732	GHCND:USW00014732	GHCND:USW00014732	H,,S,	,,W,2400	,,W,2400	22.7	26.1	19.4	72.86	78.98	66.92
2018-10-03	GHCND:USW00014732	GHCND:USW00014732	GHCND:USW00014732	H,,S,	,,W,2400	,,W,2400	21.8	25.0	18.9	71.24	77.00	66.02
2018-10-04	GHCND:USW00014732	GHCND:USW00014732	GHCND:USW00014732	H,,S,	,,W,2400	,,W,2400	21.3	26.1	17.8	70.34	78.98	64.04
2018-10-05	GHCND:USW00014732	GHCND:USW00014732	GHCND:USW00014732	H,,S,	,,W,2400	,,W,2400	20.3	22.8	16.1	68.54	73.04	60.98

Попередня обробка даних

Для переходу від широкого формату до довгого використовується метод melt() або stack().

datatype	TAVG	TMAX	TMIN
date			
2018-10-01	21.2	25.6	18.3
2018-10-02	22.7	26.1	19.4
2018-10-03	21.8	25.0	18.9
2018-10-04	21.3	26.1	17.8
2018-10-05	20.3	22.8	16.1

```
long_df = wide_df.melt(value_vars=['TMAX', 'TMIN',
'TAVG'],value_name='temp_C')
```

	datatype	temp_C
0	TMAX	25.6
1	TMAX	26.1
2	TMAX	25.0

Попередня обробка даних

wide_df.stack().to_frame('temp_C’)

		temp_C
date	**datatype**	
2018-10-01	**TAVG**	21.2
	TMAX	25.6
	TMIN	18.3
2018-10-02	**TAVG**	22.7
	TMAX	26.1

Попередня обробка даних

Виявлення проблем в даних.

Візьмемо наступні дані про погоду за рік:

	date	station	PRCP	SNOW	SNWD	TMAX	TMIN	TOBS	WESF	inclement_weather
0	2018-01-01T00:00:00	?	0.0	0.0	-inf	5505.0	-40.0	NaN	NaN	NaN
1	2018-01-01T00:00:00	?	0.0	0.0	-inf	5505.0	-40.0	NaN	NaN	NaN
2	2018-01-01T00:00:00	?	0.0	0.0	-inf	5505.0	-40.0	NaN	NaN	NaN
3	2018-01-02T00:00:00	GHCND:USC00280907	0.0	0.0	-inf	-8.3	-16.1	-12.2	NaN	False
4	2018-01-03T00:00:00	GHCND:USC00280907	0.0	0.0	-inf	-4.4	-13.9	-13.3	NaN	False

PRCP – опади в міліметрах

SNWD – глибина снігу в міліметрах

TOBS – значення температури під час спостереження

WESF – висота води в міліметрах, що відповідає глибині снігу

Inclement_weather – наявність некомфортних погодних умов

Попередня обробка даних

df.describe()

	PRCP	SNOW	SNWD	TMAX	TMIN	TOBS	WESF
count	765.000000	577.000000	577.0	765.000000	765.000000	398.000000	11.000000
mean	5.360392	4.202773	NaN	2649.175294	-15.914379	8.632161	16.290909
std	10.002138	25.086077	NaN	2744.156281	24.242849	9.815054	9.489832
min	0.000000	0.000000	-inf	-11.700000	-40.000000	-16.100000	1.800000
25%	0.000000	0.000000	NaN	13.300000	-40.000000	0.150000	8.600000
50%	0.000000	0.000000	NaN	32.800000	-11.100000	8.300000	19.300000
75%	5.800000	0.000000	NaN	5505.000000	6.700000	18.300000	24.900000
max	61.700000	229.000000	inf	5505.000000	23.900000	26.100000	28.700000

Попередня обробка даних

df.info()

```
RangeIndex: 765 entries, 0 to 764
Data columns (total 10 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   date               765 non-null    object
 1   station            765 non-null    object
 2   PRCP               765 non-null    float64
 3   SNOW               577 non-null    float64
 4   SNWD               577 non-null    float64
 5   TMAX               765 non-null    float64
 6   TMIN               765 non-null    float64
 7   TOBS               398 non-null    float64
 8   WESF               11 non-null     float64
 9   inclement_weather  408 non-null    object
dtypes: float64(7), object(3)
memory usage: 59.9+ KB
```

df[df.inclement_weather.isna()].shape[0]

357

df[df.SNWD.isin([-np.inf, np.inf])].shape[0]

577

Попередня обробка даних

```
contain_nulls = df[df.SNOW.isna() | df.SNWD.isna() | df.TOBS.isna() | df.WESF.isna() | df.inclement_weather.isna()]
contain_nulls.head()
```

	date	station	PRCP	SNOW	SNWD	TMAX	TMIN	TOBS	WESF	inclement_weather
0	2018-01-01T00:00:00	?	0.0	0.0	-inf	5505.0	-40.0	NaN	NaN	NaN
1	2018-01-01T00:00:00	?	0.0	0.0	-inf	5505.0	-40.0	NaN	NaN	NaN
2	2018-01-01T00:00:00	?	0.0	0.0	-inf	5505.0	-40.0	NaN	NaN	NaN
3	2018-01-02T00:00:00	GHCND:USC00280907	0.0	0.0	-inf	-8.3	-16.1	-12.2	NaN	False
4	2018-01-03T00:00:00	GHCND:USC00280907	0.0	0.0	-inf	-4.4	-13.9	-13.3	NaN	False

```
contain_nulls.shape[0]
```

765

Попередня обробка даних

```
pd.DataFrame({'np.inf Snow Depth':df[df.SNWD == np.inf].SNOW.describe(),'-np.inf Snow Depth':df[df.SNWD == -np.inf].SNOW.describe()}).T
```

	count	mean	std	min	25%	50%	75%	max
np.inf Snow Depth	24.0	101.041667	74.498018	13.0	25.0	120.5	152.0	229.0
-np.inf Snow Depth	553.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0

```
df.describe(include='object')
```

	date	station	inclement_weather
count	765	765	408
unique	324	2	2
top	2018-07-05T00:00:00	GHCND:USC00280907	False
freq	8	398	384

Попередня обробка даних

Виявимо, чи наявні однакові рядки:

Рахує тільки перші рядки:

```
df[df.duplicated()].shape[0]
```

284

Всі рядки:

```
df[df.duplicated(keep=False)].shape[0]
```

482

За заданими стовпцями:

```
df[df.duplicated(['date', 'station'])].shape[0]
```

284

Попередня обробка даних

df[df.duplicated()].head()

	date	station	PRCP	SNOW	SNWD	TMAX	TMIN	TOBS	WESF	inclement_weather
1	2018-01-01T00:00:00	?	0.0	0.0	-inf	5505.0	-40.0	NaN	NaN	NaN
2	2018-01-01T00:00:00	?	0.0	0.0	-inf	5505.0	-40.0	NaN	NaN	NaN
5	2018-01-03T00:00:00	GHCND:USC00280907	0.0	0.0	-inf	-4.4	-13.9	-13.3	NaN	False
6	2018-01-03T00:00:00	GHCND:USC00280907	0.0	0.0	-inf	-4.4	-13.9	-13.3	NaN	False
8	2018-01-04T00:00:00	?	20.6	229.0	inf	5505.0	-40.0	NaN	19.3	True

Попередня обробка даних

Виправлення помилок.

Виділяємо дані для невідомої станції в окремий об’єкт

```
wesf_df = df[df.station ==
'?'].drop_duplicates('date').set_index('date').WESF
```

```
date
2018-01-01T00:00:00     NaN
2018-01-04T00:00:00    19.3
2018-01-05T00:00:00     NaN
2018-01-07T00:00:00     NaN
2018-01-08T00:00:00     NaN
```

```
df.sort_values('station', ascending=False, inplace=True)
```

Попередня обробка даних

Видаляємо продубльовані дати:

```
df_new = df.drop_duplicates('date’)
df_new.shape[0]
```

324

Видаляємо стовпець з назвою станції:

```
df_new.drop(columns='station').set_index('date').sort_index()
```

date	PRCP	SNOW	SNWD	TMAX	TMIN	TOBS	WESF	inclement_weather
2018-01-01T00:00:00	0.0	0.0	-inf	5505.0	-40.0	NaN	NaN	NaN
2018-01-02T00:00:00	0.0	0.0	-inf	-8.3	-16.1	-12.2	NaN	False
2018-01-03T00:00:00	0.0	0.0	-inf	-4.4	-13.9	-13.3	NaN	False
2018-01-04T00:00:00	20.6	229.0	inf	5505.0	-40.0	NaN	19.3	True
2018-01-05T00:00:00	0.3	NaN	NaN	5505.0	-40.0	NaN	NaN	NaN

Попередня обробка даних

df_new = df_new.assign(WESF=lambda x: x.WESF.combine_first(wesf_df))

date	PRCP	SNOW	SNWD	TMAX	TMIN	TOBS	WESF	inclement_weather
2018-01-01T00:00:00	0.0	0.0	-inf	5505.0	-40.0	NaN	NaN	NaN
2018-01-02T00:00:00	0.0	0.0	-inf	-8.3	-16.1	-12.2	NaN	False
2018-01-03T00:00:00	0.0	0.0	-inf	-4.4	-13.9	-13.3	NaN	False
2018-01-04T00:00:00	20.6	229.0	inf	5505.0	-40.0	NaN	19.3	True
2018-01-05T00:00:00	0.3	NaN	NaN	5505.0	-40.0	NaN	NaN	NaN

df_new.shape
(324,8)
df_ new.dropna().shape
(4,8)
df_new.dropna(how='all').shape
(324,8)

Попередня обробка даних

df_new.dropna(axis='columns',thresh=df_new.shape[0] * .75).columns

```
Index(['PRCP', 'SNOW', 'SNWD', 'TMAX', 'TMIN', 'TOBS', 'inclement_weather'], dtype='object')
```

Оскільки відсутніх значень забагато краще замість відкидання заповнити їх:

df_new.loc[:,'WESF'].fillna(0, inplace=True)

date	PRCP	SNOW	SNWD	TMAX	TMIN	TOBS	WESF	inclement_weather
2018-01-01T00:00:00	0.0	0.0	-inf	5505.0	-40.0	NaN	0.0	NaN
2018-01-02T00:00:00	0.0	0.0	-inf	-8.3	-16.1	-12.2	0.0	False
2018-01-03T00:00:00	0.0	0.0	-inf	-4.4	-13.9	-13.3	0.0	False
2018-01-04T00:00:00	20.6	229.0	inf	5505.0	-40.0	NaN	19.3	True
2018-01-05T00:00:00	14.2	127.0	inf	-4.4	-13.9	-13.9	0.0	True

Попередня обробка даних

Замінюємо надвисокі та наднизькі значення температури на відсутні значення:
df_new.assign(TMAX=lambda x: x.TMAX.replace(5505, np.nan),TMIN=lambda x: x.TMIN.replace(-40, np.nan))

date	PRCP	SNOW	SNWD	TMAX	TMIN	TOBS	WESF	inclement_weather
2018-01-01T00:00:00	0.0	0.0	-inf	NaN	NaN	NaN	0.0	NaN
2018-01-02T00:00:00	0.0	0.0	-inf	-8.3	-16.1	-12.2	0.0	False
2018-01-03T00:00:00	0.0	0.0	-inf	-4.4	-13.9	-13.3	0.0	False
2018-01-04T00:00:00	20.6	229.0	inf	NaN	NaN	NaN	19.3	True
2018-01-05T00:00:00	14.2	127.0	inf	-4.4	-13.9	-13.9	0.0	True

Попередня обробка даних

Заповнюємо ці значення сусідніми:
df_new.assign(TMAX=lambda x: x.TMAX.fillna(method='ffill'),TMIN=lambda x: x.TMIN.fillna(method='ffill'))

date	PRCP	SNOW	SNWD	TMAX	TMIN	TOBS	WESF	inclement_weather
2018-01-01T00:00:00	0.0	0.0	-inf	NaN	NaN	NaN	0.0	NaN
2018-01-02T00:00:00	0.0	0.0	-inf	-8.3	-16.1	-12.2	0.0	False
2018-01-03T00:00:00	0.0	0.0	-inf	-4.4	-13.9	-13.3	0.0	False
2018-01-04T00:00:00	20.6	229.0	inf	-4.4	-13.9	NaN	19.3	True
2018-01-05T00:00:00	14.2	127.0	inf	-4.4	-13.9	-13.9	0.0	True

Попередня обробка даних

Замінюємо нескінченні значення на великі числа для того, щоб до цих даних можна було застосовувати алгоритми машинного навчання:

df_new.assign(SNWD=lambda x: np.nan_to_num(x.SNWD)).head()

date	PRCP	SNOW	SNWD	TMAX	TMIN	TOBS	WESF	inclement_weather
2018-01-01T00:00:00	0.0	0.0	-1.797693e+308	NaN	NaN	NaN	0.0	NaN
2018-01-02T00:00:00	0.0	0.0	-1.797693e+308	-8.3	-16.1	-12.2	0.0	False
2018-01-03T00:00:00	0.0	0.0	-1.797693e+308	-4.4	-13.9	-13.3	0.0	False
2018-01-04T00:00:00	20.6	229.0	1.797693e+308	-4.4	-13.9	NaN	19.3	True
2018-01-05T00:00:00	14.2	127.0	1.797693e+308	-4.4	-13.9	-13.9	0.0	True

Попередня обробка даних

Або обмежити значення:

df_new.assign(SNWD=lambda x: x.SNWD.clip(0, x.SNOW)).head()

date	PRCP	SNOW	SNWD	TMAX	TMIN	TOBS	WESF	inclement_weather
2018-01-01T00:00:00	0.0	0.0	0.0	-8.3	-16.1	NaN	0.0	NaN
2018-01-02T00:00:00	0.0	0.0	0.0	-8.3	-16.1	-12.2	0.0	False
2018-01-03T00:00:00	0.0	0.0	0.0	-4.4	-13.9	-13.3	0.0	False
2018-01-04T00:00:00	20.6	229.0	229.0	-4.4	-13.9	NaN	19.3	True
2018-01-05T00:00:00	14.2	127.0	127.0	-4.4	-13.9	-13.9	0.0	True

Попередня обробка даних

df_new.assign(TMAX=lambda x: x.TMAX.fillna(x.TMAX.median()),TMIN=lambda x: x.TMIN.fillna(x.TMIN.median()),TOBS=lambda x: x.TOBS.fillna((x.TMAX + x.TMIN) / 2))

date	PRCP	SNOW	SNWD	TMAX	TMIN	TOBS	WESF	inclement_weather
2018-01-01	0.0	0.0	0.0	14.4	5.6	10.0	0.0	NaN
2018-01-02	0.0	0.0	0.0	-8.3	-16.1	-12.2	0.0	False
2018-01-03	0.0	0.0	0.0	-4.4	-13.9	-13.3	0.0	False
2018-01-04	20.6	229.0	229.0	14.4	5.6	10.0	19.3	True
2018-01-05	14.2	127.0	127.0	-4.4	-13.9	-13.9	0.0	True

Попередня обробка даних

df_new.apply(lambda x: x.fillna(x.rolling(7, min_periods=0).median())).head(8)

date	PRCP	SNOW	SNWD	TMAX	TMIN	TOBS	WESF	inclement_weather
2018-01-01	0.0	0.0	0.0	NaN	NaN	NaN	0.0	NaN
2018-01-02	0.0	0.0	0.0	-8.30	-16.1	-12.20	0.0	False
2018-01-03	0.0	0.0	0.0	-4.40	-13.9	-13.30	0.0	False
2018-01-04	20.6	229.0	229.0	-6.35	-15.0	-12.75	19.3	True
2018-01-05	14.2	127.0	127.0	-4.40	-13.9	-13.90	0.0	True
2018-01-06	0.0	0.0	0.0	-10.00	-15.6	-15.00	0.0	False
2018-01-07	0.0	0.0	0.0	-11.70	-17.2	-16.10	0.0	False
2018-01-08	0.0	0.0	0.0	-7.80	-16.7	-8.30	0.0	False

Попередня обробка даних

Збагачення даних відбувається через або об’єднання нових даних з вихідними даними (додаючи нові рядки або стовпці), або використання вихідних даних для створення нових даних. Основні способи збагачення даних:

- Додавання нових стовпців: використання функціональних перетворень над даними з наявних стовпців для створення нових значень.
- Групування: перетворення неперервних даних або дискретних даних з багатьма різними значеннями в сегменти. Таким чином стовпець стає дискретним, що дозволяє контролювати кількість можливих значень у стовпці.
- Агрегація: згортання даних та їх узагальнення.
- Повторна вибірка: агрегування даних часових рядів через певні проміжки часу.