weather_temperature_prediction

December 3, 2024

[2]:
```
import pandas as pd
weather = pd.read_csv("dataset.csv", index_col="datetime")
```

Here we have imported pandas to filter Ahmedabad's weather data set

[4]:
```
weather
```

[4]:

datetime	name	tempmax	tempmin	temp	humidity	precip	windgust	windspeed	visibility	uvindex
2022-12-01	Ahmedabad,India	30.0	17.0	22.9	51.6	0.0	30.2	18.4	4.0	8
2022-12-02	Ahmedabad,India	31.0	17.0	22.5	54.8	0.0	16.2	14.8	3.9	7
2022-12-03	Ahmedabad,India	30.0	16.0	22.5	59.1	0.0	12.6	11.2	3.8	6
2022-12-04	Ahmedabad,India	30.0	15.6	22.1	59.5	0.0	14.0	11.2	3.8	7
2022-12-05	Ahmedabad,India	30.0	16.0	23.0	64.3	0.0	18.0	14.8	3.7	7
…	…	…	…	…	…	…	…	…	…	…
2024-11-28	Ahmedabad,India	30.7	17.0	23.4	45.8	0.0	20.9	11.2	3.6	7
2024-11-29	Ahmedabad,India	28.0	18.0	22.5	41.7	0.0	27.4	33.5	4.0	7
2024-11-30	Ahmedabad,India	27.0	16.0	21.0	43.4	0.0	28.4	18.4	3.9	8
2024-12-01	Ahmedabad,India	28.0	15.0	21.3	43.2	0.0	29.5	18.4	3.4	8
2024-12-02	Ahmedabad,India	29.0	16.0	22.0	44.1	0.0	31.3	27.7	3.4	8

[733 rows x 10 columns]

[5]:
```
weather.index = pd.to_datetime(weather.index)
weather.index
```

[5]:
```
DatetimeIndex(['2022-12-01', '2022-12-02', '2022-12-03', '2022-12-04',
               '2022-12-05', '2022-12-06', '2022-12-07', '2022-12-08',
               '2022-12-09', '2022-12-10',
               …
               '2024-11-23', '2024-11-24', '2024-11-25', '2024-11-26',
               '2024-11-27', '2024-11-28', '2024-11-29', '2024-11-30',
               '2024-12-01', '2024-12-02'],
              dtype='datetime64[ns]', name='datetime', length=733, freq=None)
```

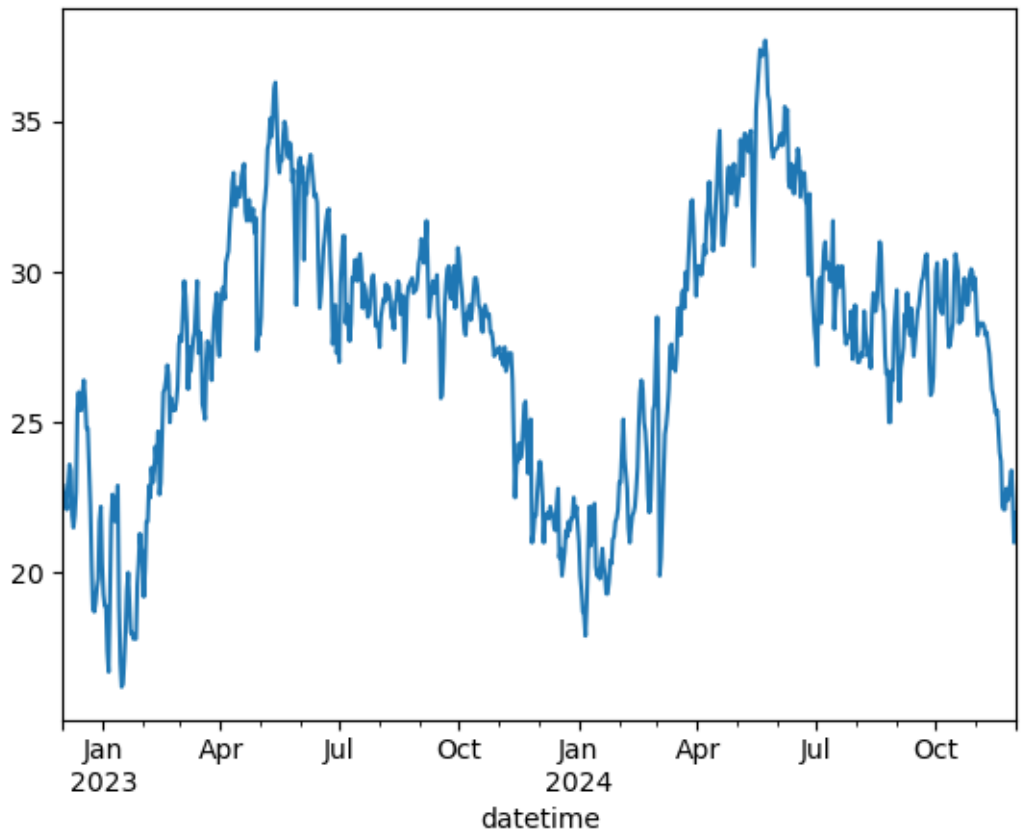
[6]:
```
weather.index.year.value_counts().sort_index()
```

[6]:
```
datetime
2022     31
2023    365
2024    337
Name: count, dtype: int64
```

We had plot temperature graph and will define a function to predict the future temperature

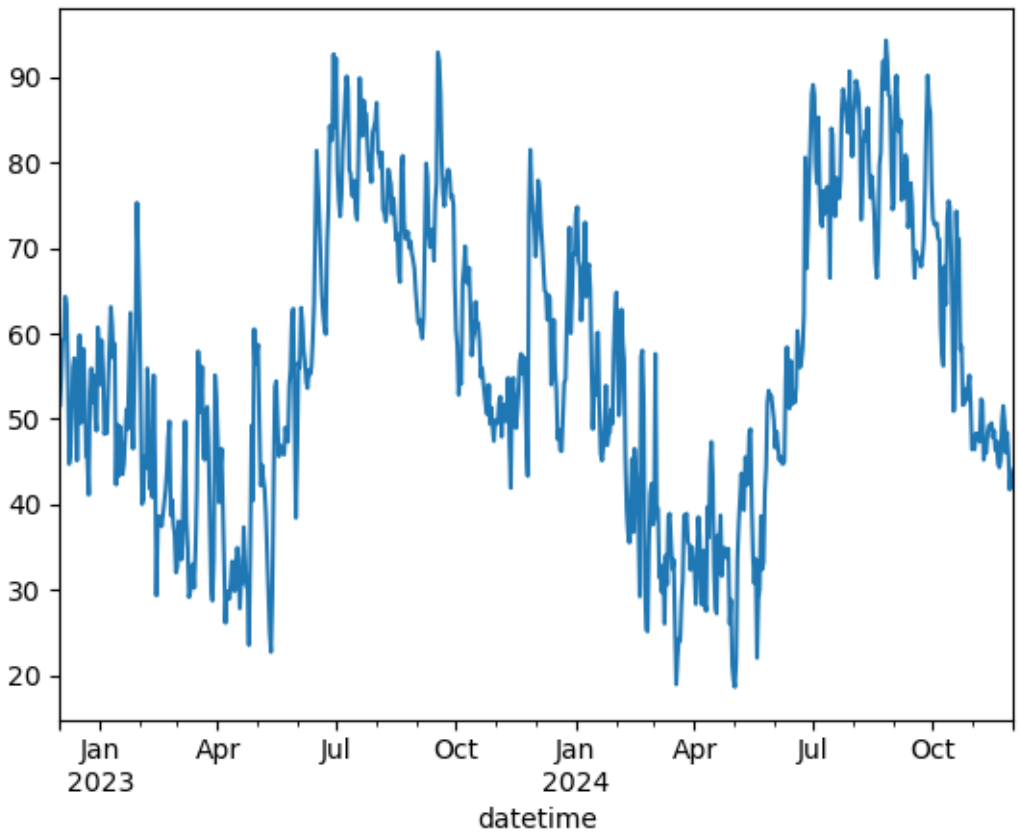
[8]:
```
weather["temp"].plot()
```

[8]:
```
<Axes: xlabel='datetime'>
```



[9]:
```
weather["humidity"].plot()
```

[9]:
```
<Axes: xlabel='datetime'>
```



[10]:
```
weather["Prediction-temperature"] =  weather.shift(-1)["temp"]
```

[11]:
```
weather
```

[11]:

datetime	name	tempmax	tempmin	temp	humidity	precip	\
2022-12-01	Ahmedabad,India	30.0	17.0	22.9	51.6	0.0	
2022-12-02	Ahmedabad,India	31.0	17.0	22.5	54.8	0.0	
2022-12-03	Ahmedabad,India	30.0	16.0	22.5	59.1	0.0	
2022-12-04	Ahmedabad,India	30.0	15.6	22.1	59.5	0.0	
2022-12-05	Ahmedabad,India	30.0	16.0	23.0	64.3	0.0	
…	…	…	…	…	…	…	
2024-11-28	Ahmedabad,India	30.7	17.0	23.4	45.8	0.0	
2024-11-29	Ahmedabad,India	28.0	18.0	22.5	41.7	0.0	
2024-11-30	Ahmedabad,India	27.0	16.0	21.0	43.4	0.0	
2024-12-01	Ahmedabad,India	28.0	15.0	21.3	43.2	0.0	
2024-12-02	Ahmedabad,India	29.0	16.0	22.0	44.1	0.0	

datetime	windgust	windspeed	visibility	uvindex	Prediction-temperature

```
2022-12-01      30.2       18.4         4.0        8                    22.5
2022-12-02      16.2       14.8         3.9        7                    22.5
2022-12-03      12.6       11.2         3.8        6                    22.1
2022-12-04      14.0       11.2         3.8        7                    23.0
2022-12-05      18.0       14.8         3.7        7                    23.6
…               …          …            …          …                    …
2024-11-28      20.9       11.2         3.6        7                    22.5
2024-11-29      27.4       33.5         4.0        7                    21.0
2024-11-30      28.4       18.4         3.9        8                    21.3
2024-12-01      29.5       18.4         3.4        8                    22.0
2024-12-02      31.3       27.7         3.4        8                     NaN

[733 rows x 11 columns]
```

[12]:
```
weather = weather.ffill()
weather
```

[12]:
```
                      name  tempmax  tempmin  temp  humidity  precip  \
datetime
2022-12-01  Ahmedabad,India     30.0     17.0  22.9      51.6     0.0
2022-12-02  Ahmedabad,India     31.0     17.0  22.5      54.8     0.0
2022-12-03  Ahmedabad,India     30.0     16.0  22.5      59.1     0.0
2022-12-04  Ahmedabad,India     30.0     15.6  22.1      59.5     0.0
2022-12-05  Ahmedabad,India     30.0     16.0  23.0      64.3     0.0
…                       …       …        …     …         …       …
2024-11-28  Ahmedabad,India     30.7     17.0  23.4      45.8     0.0
2024-11-29  Ahmedabad,India     28.0     18.0  22.5      41.7     0.0
2024-11-30  Ahmedabad,India     27.0     16.0  21.0      43.4     0.0
2024-12-01  Ahmedabad,India     28.0     15.0  21.3      43.2     0.0
2024-12-02  Ahmedabad,India     29.0     16.0  22.0      44.1     0.0

            windgust  windspeed  visibility  uvindex  Prediction-temperature
datetime
2022-12-01      30.2       18.4         4.0        8                    22.5
2022-12-02      16.2       14.8         3.9        7                    22.5
2022-12-03      12.6       11.2         3.8        6                    22.1
2022-12-04      14.0       11.2         3.8        7                    23.0
2022-12-05      18.0       14.8         3.7        7                    23.6
…               …          …            …          …                    …
2024-11-28      20.9       11.2         3.6        7                    22.5
2024-11-29      27.4       33.5         4.0        7                    21.0
2024-11-30      28.4       18.4         3.9        8                    21.3
2024-12-01      29.5       18.4         3.4        8                    22.0
2024-12-02      31.3       27.7         3.4        8                    22.0

[733 rows x 11 columns]
```

We had imported Ridge regression model from sklearn it is very similar to linear regression model and initialized it.

[14]:
```
from sklearn.linear_model import Ridge
rr = Ridge(alpha=.1)
```

Predictors columns are created to predict the temperature

[16]:
```
predictors = weather.columns[~weather.columns.
 ↪isin(["name","Prediction-temperature"
                                                ])]
```

[17]:
```
predictors
```

[17]:
```
Index(['tempmax', 'tempmin', 'temp', 'humidity', 'precip', 'windgust',
       'windspeed', 'visibility', 'uvindex'],
      dtype='object')
```

We are defining function called backtest which is going to take weather data frame,Ridge model,predictors

[19]:
```
def backtest(weather,model,predictors,start=60,step=10):
    all_prediction = []

    for i in range(start,weather.shape[0],step):
        train = weather.iloc[:i,:]
        test = weather.iloc[i:(i+step),:]

        model.fit(train[predictors],train["Prediction-temperature"])

        preds = model.predict(test[predictors])

        preds = pd.Series(preds,index=test.index)
        combined = pd.concat([test["Prediction-temperature"],preds],axis=1)

        combined.columns = ["actual","prediction"]
        combined["diff"] = (combined["prediction"] - combined["actual"]).abs()

        all_prediction.append(combined)
    return pd.concat(all_prediction)
```

[20]:
```
predictions = backtest(weather, rr, predictors)
```

[21]:
```
predictions
```

[21]:

datetime	actual	prediction	diff
2023-01-30	20.6	21.103984	0.503984

```
2023-01-31    19.2   20.483182  1.283182
2023-02-01    20.1   20.290868  0.190868
2023-02-02    21.7   20.588340  1.111660
2023-02-03    21.7   21.237743  0.462257
…              …         …        …
2024-11-28    22.5   23.422898  0.922898
2024-11-29    21.0   22.393851  1.393851
2024-11-30    21.3   21.045840  0.254160
2024-12-01    22.0   21.494535  0.505465
2024-12-02    22.0   22.071395  0.071395

[673 rows x 3 columns]
```