Weather Forecast Predictor

Sultan Mahmud

Introduction

In this project we are using the dataset taken from Kaggle for making analysis with different data mining approaches on weather forecast. At first, we visualize the data set and try to find if there is any specific kind of noise as well as outliers. Then we preprocess the data and after that we have tried to fit our model with different classifier for getting a prediction result so that we can evaluate our model. A small glimpse of time series is also applied at the end of the project for getting or analyzing the attributes like temperature (c), summary, wind speed with apparent temperature. we have used matrix profile for getting the normalized values and after that applied discords and trying to find if there are any anomalies on that specific attributes. we have found anomalies in each attribute we were using during applying the matrix profile. In future we would like to build a model which could predict weather for a specific place based on previous data.

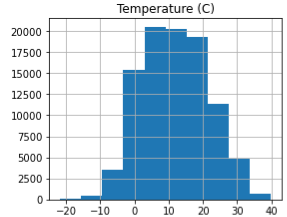
Dataset Description

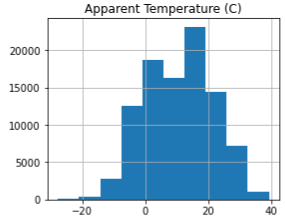
- This dataset contains historical daily weather data for Leeds, England from Jan 1st, 2006, to Dec 31st, 2016.
- This dataset was taken from Kaggle website. Below is the given link:

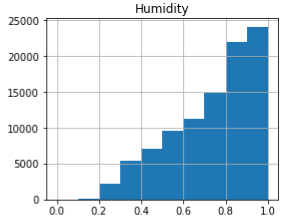
  https://www.kaggle.com/datasets/muthuj7/weather-dataset

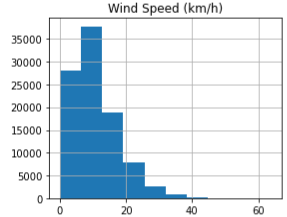
- 12 Attributes (Formatted Date, Summary, Precip Type, Temperature (C), Apparent Temperature (C), Humidity, Wind Speed (km/h), Wind Bearing (degrees), Visibility (km), Loud Cover, Pressure (millibars), Daily Summary)
- Row 96500 and 12 columns
- Instances (96500*12)

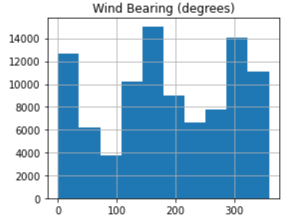
Data Visualization

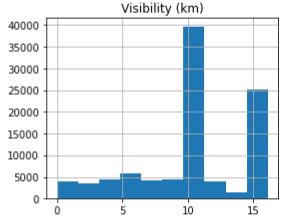


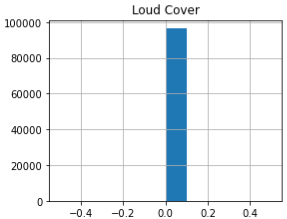


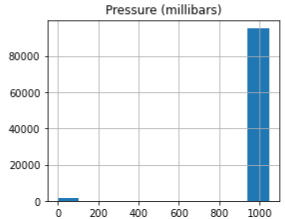


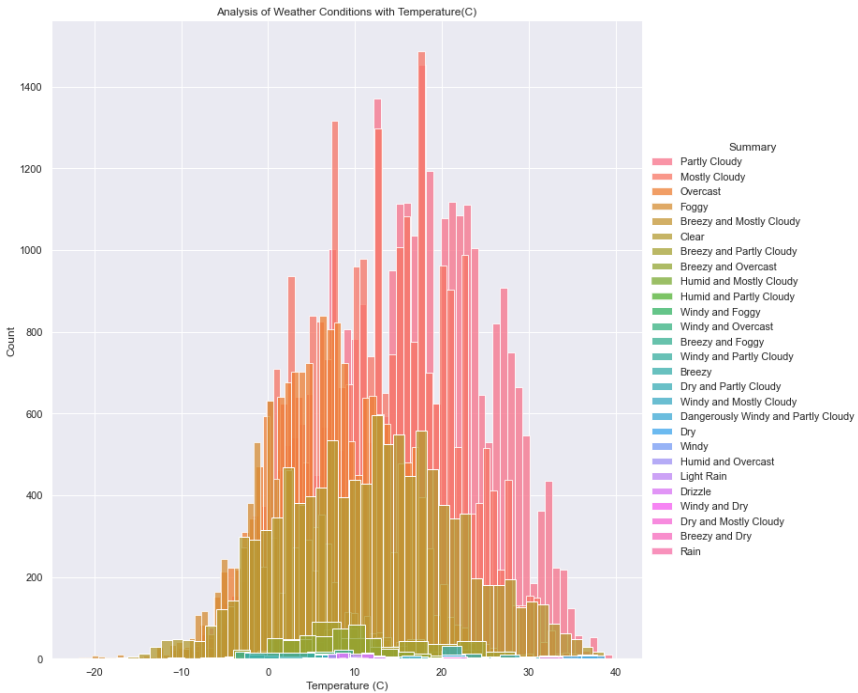




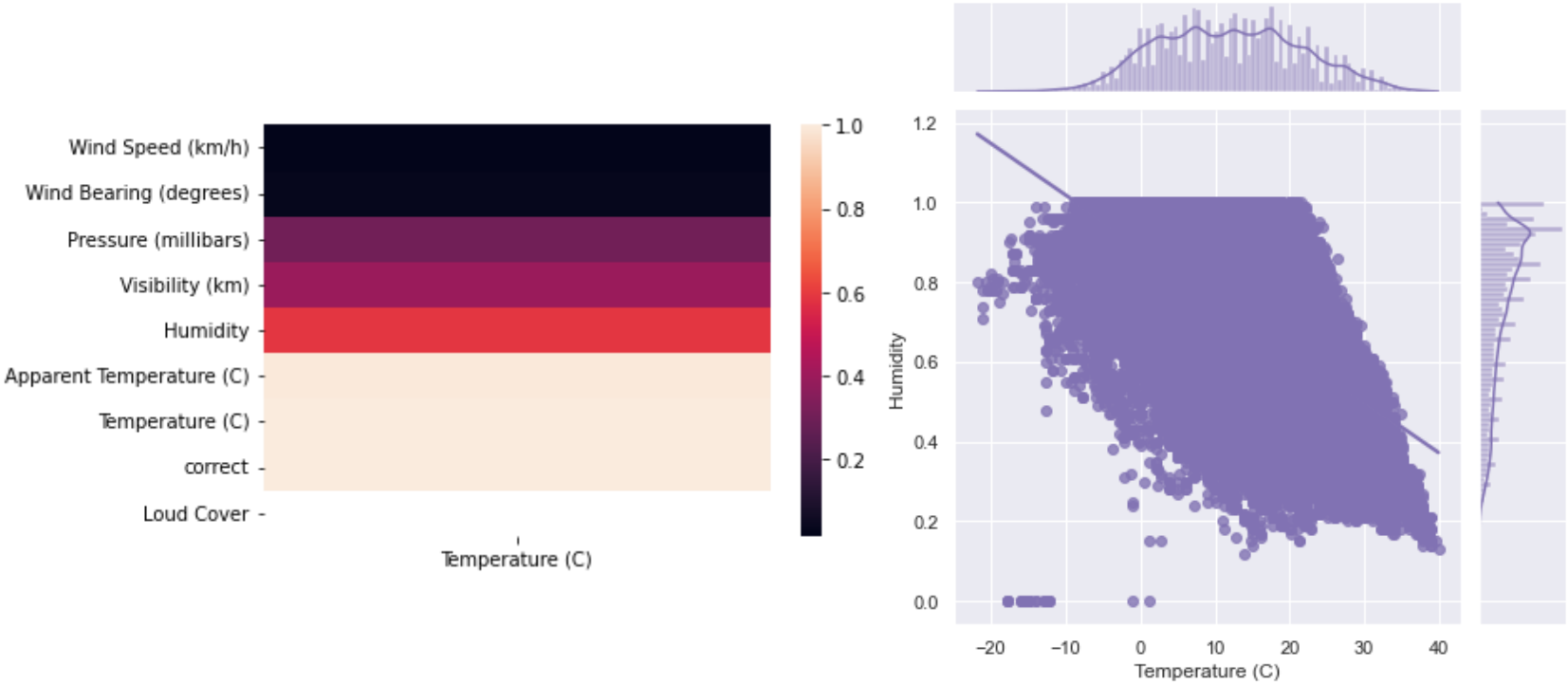




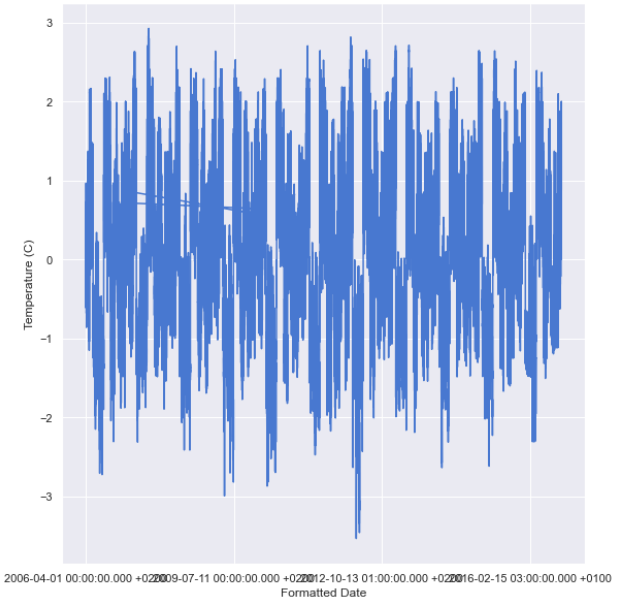


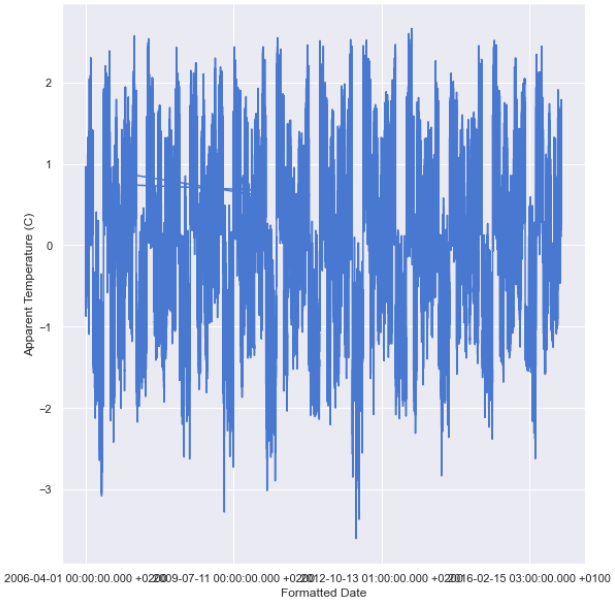


Data Visualization

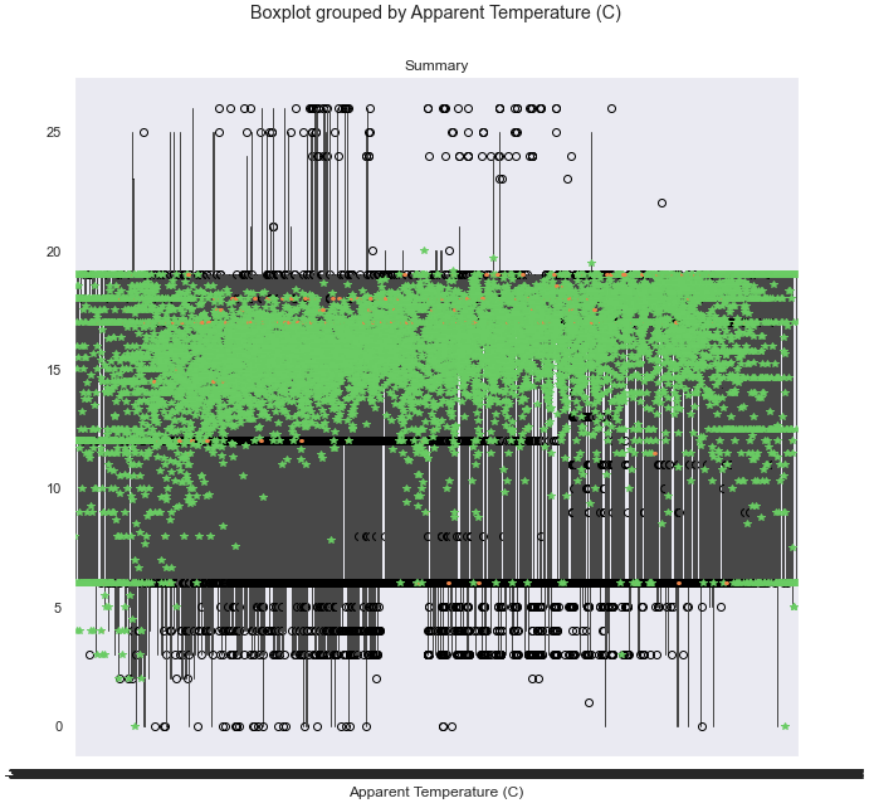


Data Visualization





Data Visualization



Data Preprocessing

```
df.isnull().sum()
```

```
Formatted Date                0
Summary                       0
Precip Type                 517
Temperature (C)               0
Apparent Temperature (C)      0
Humidity                      0
Wind Speed (km/h)             0
Wind Bearing (degrees)        0
Visibility (km)               0
Loud Cover                    0
Pressure (millibars)          0
Daily Summary                 0
dtype: int64
```

```
df.drop(['Daily Summary','Loud Cover'],axis=1,inplace=True)
```

```
le = LabelEncoder()
df['Precip Type']=le.fit_transform(df['Precip Type'])
df['Summary']=le.fit_transform(df['Summary'])
df.head(10)
```

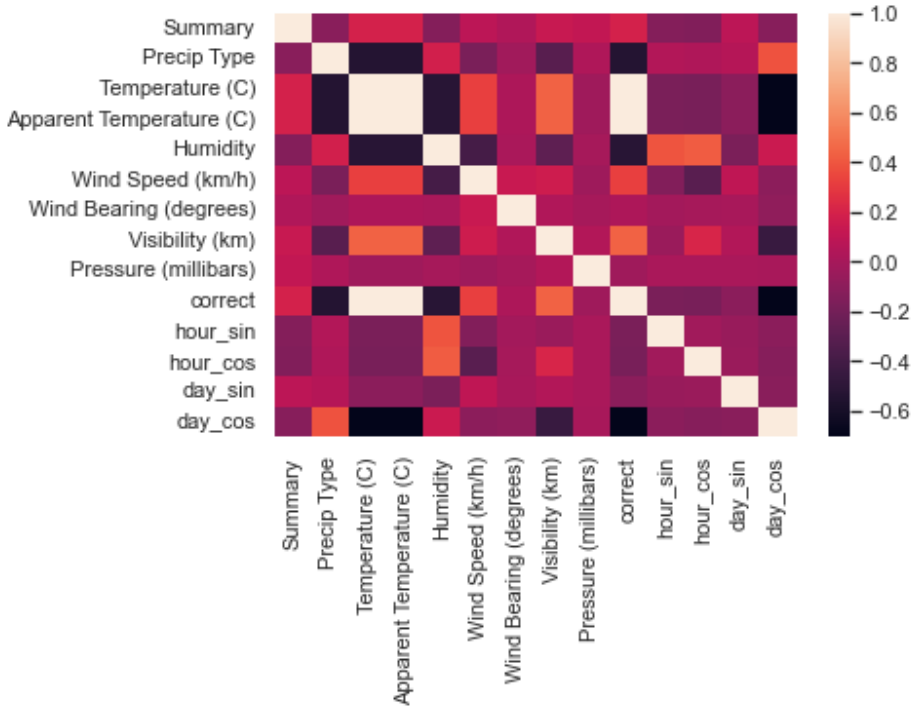
```
scaler = StandardScaler()
df[df.columns[2:]] = scaler.fit_transform(df[df.columns[2:]])
df.head()
```

```
(                           Summary  Precip Type  Temperature (C)  \
Summary                    1.000000    -0.118078         0.191193
Precip Type               -0.118078     1.000000        -0.541422
Temperature (C)            0.191193    -0.541422         1.000000
Apparent Temperature (C)   0.191193    -0.541422         1.000000
Humidity                  -0.137957     0.175191        -0.518759
Wind Speed (km/h)          0.084177    -0.172403         0.317136
Wind Bearing (degrees)     0.038864    -0.024229         0.027619
Visibility (km)            0.129828    -0.317517         0.436935
Pressure (millibars)       0.111859     0.030302        -0.028021
correct                    0.191193    -0.541422         1.000000
hour_sin                  -0.136505     0.043074        -0.173999
hour_cos                  -0.150465     0.031393        -0.180936
day_sin                    0.081952     0.058150        -0.109402
day_cos                   -0.118803     0.384120        -0.704746

                          Apparent Temperature (C)  Humidity  ...   correct  \
Summary                                   0.191193 -0.137957  ...  0.191193
Precip Type                              -0.541422  0.175191  ... -0.541422
Temperature (C)                           1.000000 -0.518759  ...  1.000000
Apparent Temperature (C)                  1.000000 -0.518759  ...  1.000000
Humidity                                 -0.518759  1.000000  ... -0.518759
Wind Speed (km/h)                         0.317136 -0.395751  ...  0.317136
Wind Bearing (degrees)                    0.027619  0.012985  ...  0.027619
Visibility (km)                           0.436935 -0.287946  ...  0.436935
Pressure (millibars)                     -0.028021  0.000318  ... -0.028021
correct                                   1.000000 -0.518759  ...  1.000000
hour_sin                                 -0.173999  0.392036  ... -0.173999
hour_cos                                 -0.180936  0.414462  ... -0.180936
day_sin                                  -0.109402 -0.166902  ... -0.109402
day_cos                                  -0.704746  0.146627  ... -0.704746

                           hour_sin  hour_cos   day_sin   day_cos
Summary                   -0.136505 -0.150465  0.081952 -0.118803
Precip Type                0.043074  0.031393  0.058150  0.384120
Temperature (C)           -0.173999 -0.180936 -0.109402 -0.704746
Apparent Temperature (C)  -0.173999 -0.180936 -0.109402 -0.704746
Humidity                   0.392036  0.414462 -0.166902  0.146627
Wind Speed (km/h)         -0.145343 -0.307446  0.099526 -0.104144
Wind Bearing (degrees)    -0.017779  0.000656  0.006656 -0.091987
Visibility (km)           -0.047366  0.210791  0.045477 -0.450109
Pressure (millibars)       0.010953  0.009489  0.013230  0.007603
correct                   -0.173999 -0.180936 -0.109402 -0.704746
hour_sin                   1.000000 -0.022358 -0.056623 -0.108930
hour_cos                  -0.022358  1.000000 -0.051136 -0.130914
day_sin                   -0.056623 -0.051136  1.000000 -0.118039
day_cos                   -0.108930 -0.130914 -0.118039  1.000000
```

Data Preprocessing

```
def discretize_date(current_date, t):
    current_date = current_date[:-10]
    cdate = datetime.strptime(current_date, '%Y-%m-%d %H:%M:%S')

    if t == 'hour_sin':
        return np.sin(2 * np.pi * cdate.hour/24.0)
    if t == 'hour_cos':
        return np.cos(2 * np.pi * cdate.hour/24.0)
    if t == 'day_sin':
        return np.sin(2 * np.pi * cdate.timetuple().tm_yday/365.0)
    if t == 'day_cos':
        return np.cos(2 * np.pi * cdate.timetuple().tm_yday/365.0)
```

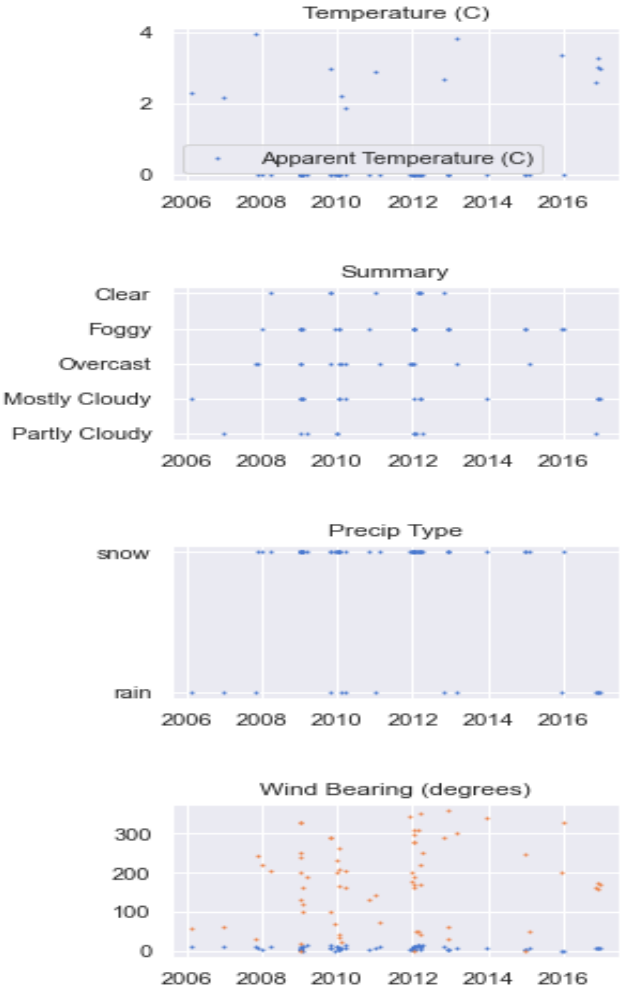
```
date_types = ['hour_sin', 'hour_cos', 'day_sin', 'day_cos']
for dt in date_types:
    df[dt] = df['Formatted Date'].apply(lambda x : discretize_date(x, dt))
df.drop(['Formatted Date'],axis=1,inplace=True)
```



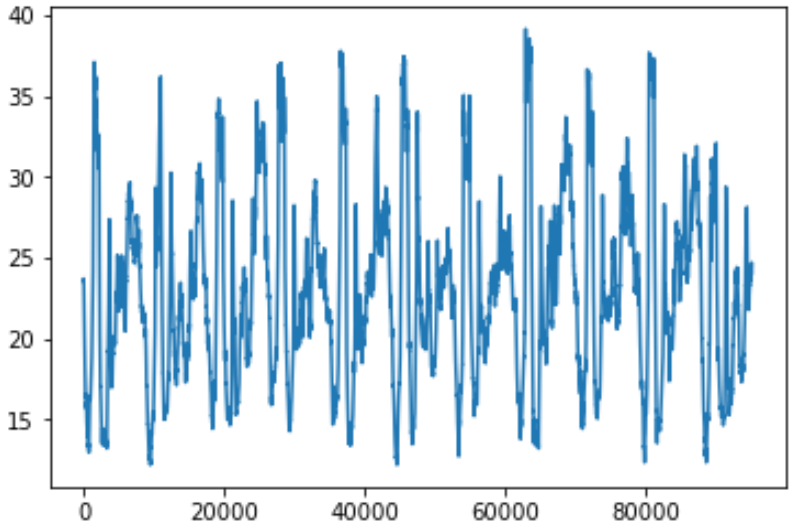
Classification

Classifier	Accuracy
Random Forest	0.62798
KNN	0.53226
Logistic Regression	0.49798
Decision Tree	0.69636

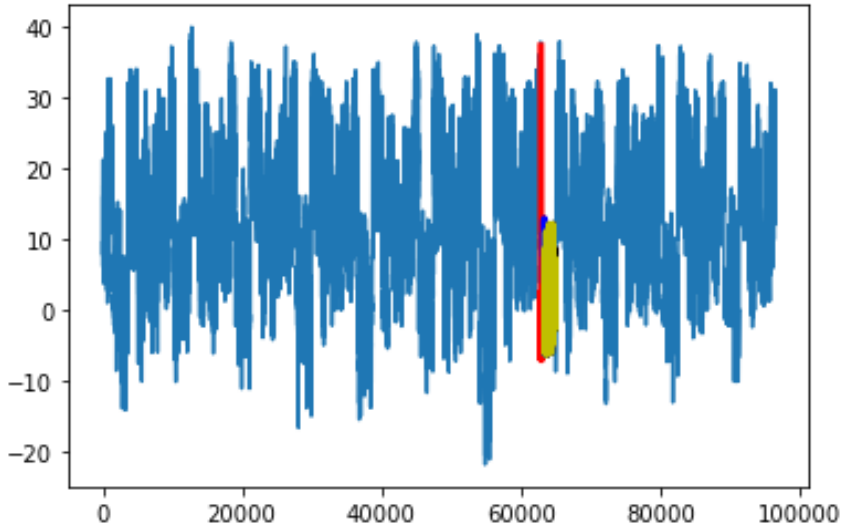
Time series


Temperature (C)
4
2
0
Apparent Temperature (C)
2006
2008
2010
2012
2014
2016
Summary
Clear
Foggy
Overcast
Mostly Cloudy
Partly Cloudy
2006
2008
2010
2012
2014
2016
Precip Type
snow
rain
2006
2008
2010
2012
2014
2016
Wind Bearing (degrees)
300
200
100
0
2006
2008
2010
2012
2014
2016

Time series

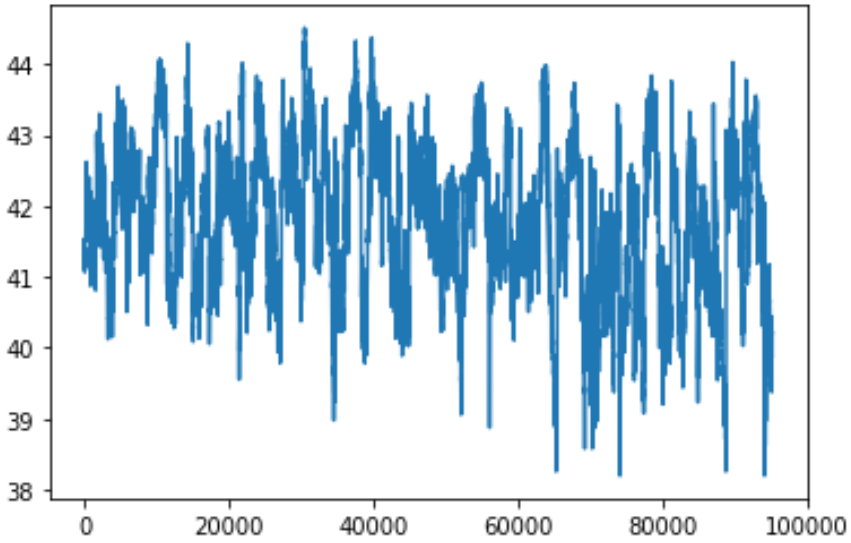


Matrix Profiling of Temperature C

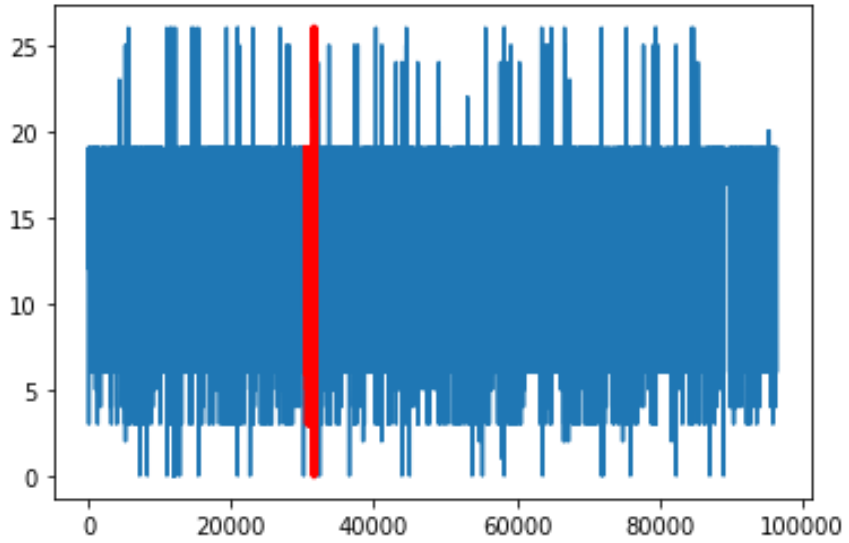


Anomaly Detection of Temperature C

Time series

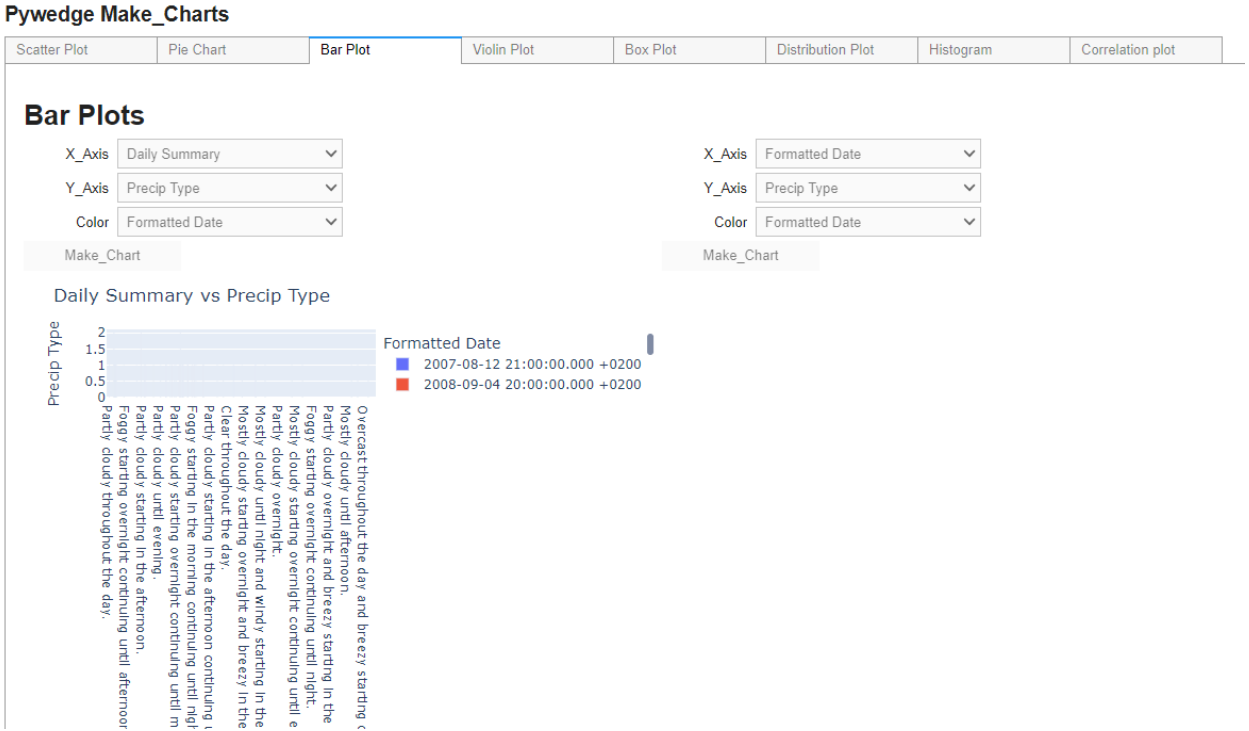


Matrix Profiling of Summary



Anomaly Detection of Summary

Pywedge


Pywedge Make_Charts
Scatter Plot
Pie Chart
Bar Plot
Violin Plot
Box Plot
Distribution Plot
Histogram
Correlation plot
Bar Plots
X_Axis
Daily Summary
Y_Axis
Precip Type
Color
Formatted Date
Make_Chart
X_Axis
Formatted Date
Y_Axis
Precip Type
Color
Formatted Date
Make_Chart
Daily Summary vs Precip Type
Precip Type
2
1.5
1
0.5
0
Formatted Date
2007-08-12 21:00:00.000 +0200
2008-09-04 20:00:00.000 +0200
Partly cloudy throughout the day.
Foggy starting overnight continuing until afternoon
Partly cloudy until evening.
Partly cloudy starting in the afternoon.
Partly cloudy starting overnight continuing until m
Partly cloudy starting in the afternoon continuing
Foggy starting in the morning continuing until nigh
Partly cloudy starting overnight continuing until m
Clear throughout the day.
Mostly cloudy starting overnight and breezy in the
Mostly cloudy until night and windy starting in the
Partly cloudy overnight.
Mostly cloudy starting overnight continuing until e
Foggy starting overnight continuing until night.
Partly cloudy overnight and breezy starting in the
Mostly cloudy until afternoon.
Overcast throughout the day and breezy starting

Thank You