1. Brief description of the data set and a summary of its attributes

1.1. Data Source

The Dataset which I have chosen for this assignment is publicly available on Kaggle titled '**Weather in Szeged 2006-2016'.** The data contains Hourly/daily summary with temperature, pressure, wind speed and more for the city name Szeged in Hungary.

1.2. Objective

The basic objective of this data is to predict the Temperature of the area based on the various attributes/ features that are available within the dataset.

1.3. Attributes/ Features

Data available in the hourly response:

- Time – Time of Observation
- Summary – Brief summary of weather
- PrecipType – Type of precipitation
- Temperature
- apparentTemperature – Apparent temperature is the temperature equivalent perceived by humans, caused by the combined effects of air temperature, relative humidity, and wind speed.
- Humidity
- windSpeed
- windBearing
- visibility
- CloudCover
- Pressure

2. Initial plan for data exploration

The data available to us contains 10 attributes and 1 Target variable (Temperature). We have a total of 96453 observations. The plan for exploration includes:

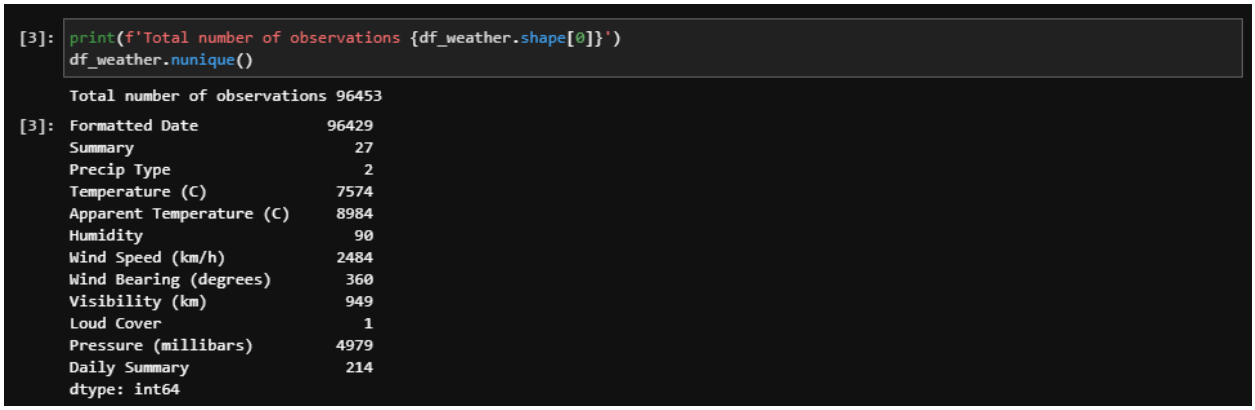
- To identify the underlying trends to find out whether these attributes describe our data in a meaningful trend.

- The tools that I have planned to use include statistical methods such as hypothesis testing, Correlation Test, scatter plots and histogram.
- Another interesting to investigate would be determine whether our target variable is normally distributed.
- During this stage, I will also take actions to identify the missing values and drop features which do not provide meaningful insights

3. Actions taken for data cleaning and feature engineering – Key findings

Key actions taken at this stage were:

- Identify the unique values under each attribute, thus enabling us to filter out non-value adding variables. This leads to the conclusion that probably attributes namely 'Formatted Date' and 'Loud Cover' do not add any meaningful information to analysis.



```
[3]: print(f'Total number of observations {df_weather.shape[0]}')
     df_weather.nunique()

     Total number of observations 96453

[3]: Formatted Date             96429
     Summary                      27
     Precip Type                   2
     Temperature (C)            7574
     Apparent Temperature (C)   8984
     Humidity                     90
     Wind Speed (km/h)          2484
     Wind Bearing (degrees)      360
     Visibility (km)             949
     Loud Cover                    1
     Pressure (millibars)       4979
     Daily Summary               214
     dtype: int64
```

- Identify the missing values by using isnull() function. I found out that for PrecipType attribute around 517 values were missing, although it is a very small number compared to the total amount of the data available with us. It would be prudent to either use major category to fill these values or otherwise drop them. I chose to do the former step and filled them with major category 'rain'.

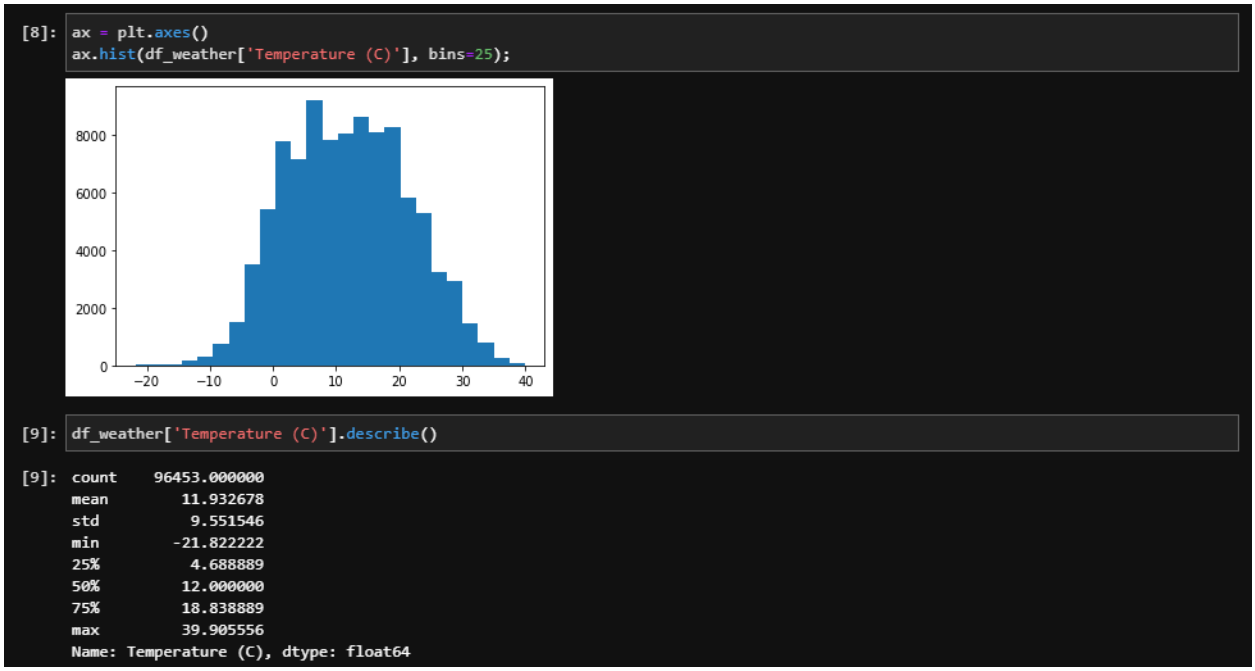
```
[5]: df_weather['Precip Type'].value_counts()

[5]: rain    85224
     snow    10712
     Name: Precip Type, dtype: int64

[6]: df_weather['Precip Type'] = df_weather['Precip Type'].fillna('rain')
```

- Further I tried to identify whether our target variable is skewed or not. For this purpose I used two methods, describe() function to identify the data distribution

and histogram plotting function of matplotlib. Both methods show a slight left skewed data. This will form the basis of first hypothesis test which we will discuss in the latter sections.



4. Formulation of Hypothesis

We formed in total 3 hypothesis based on the EDA that we performed in the previous step.

4.1. Hypothesis 1

Does precipitation (Rain/ Snow) play significant role in predicting temperature?

Null Hypothesis: There is no significant difference between the mean temperature on rainy and non-rainy days

Alternate Hypothesis: There is a significant difference between the mean temperature on rainy and non-rainy days

This will help us to decide whether to keep this feature or drop it.

4.2. Hypothesis 2

Whether the temperature is normally distributed?

Null Hypothesis: The temperature is normally distributed

Alternate Hypothesis: The temperature is not normally distributed

This will help us in deciding whether to log transform the target variable or not.

4.3. Hypothesis 3

Does mean temperature varies significantly for different summaries?

Null Hypothesis: There is no significant difference between the mean temperature among different summary group

Alternate Hypothesis: There is significant difference between the mean temperature among different summary group

This will again help us to verify whether this attribute provides any meaningful information gain or not.

5. Hypothesis Testing – Method and Results

At the onset I would like to clarify that I will test only the 1st and 2nd hypothesis, as the 3rd hypothesis testing will require the application of the ANOVA Statistics which is out of the scope of current assignment.

5.1. Testing Hypothesis 1

For testing whether precipitation plays any significant role in determining the mean temperatures, we can employ two methods, either we can define our own function to calculate z statistics and p value or otherwise we can use CompareMeans.ztest_ind() method within statsmodels module. Both will return same output. With pvalue ~ 0; thus, we can retain this feature.

Further we can also use ttest_ind method under scipy module by setting the variable 'equal_var' to false. Although the name suggests it is implementing t test, but since the sample sizes are more than 30, it will compute z statistics automatically. This is evident if you check the results for all three approaches below.

```
[43]: from scipy.stats import ttest_ind

[44]: ttest_ind(df_weather.loc[df_weather['Precip Type'] == 'rain', 'Temperature (C)'],
                df_weather.loc[df_weather['Precip Type'] == 'snow', 'Temperature (C)'],
                equal_var = False
               )

[44]: Ttest_indResult(statistic=416.37958029244624, pvalue=0.0)
```

```
[10]: M_mean = df_weather.loc[df_weather['Precip Type'] == 'rain', 'Temperature (C)'].mean()
      F_mean = df_weather.loc[df_weather['Precip Type'] == 'snow', 'Temperature (C)'].mean()
      M_std = df_weather.loc[df_weather['Precip Type'] == 'rain', 'Temperature (C)'].std()
      F_std = df_weather.loc[df_weather['Precip Type'] == 'snow', 'Temperature (C)'].std()
      no_of_M = df_weather.loc[df_weather['Precip Type'] == 'rain', 'Temperature (C)'].count()
      no_of_F = df_weather.loc[df_weather['Precip Type'] == 'snow', 'Temperature (C)'].count()

[16]: from scipy.stats import norm

[17]: def twoSampZ(X1, X2, mudiff, sd1, sd2, n1, n2):
          pooledSE = np.sqrt(sd1**2/n1 + sd2**2/n2)
          z = ((X1 - X2) - mudiff)/pooledSE
          pval = 2*(1 - norm.cdf(abs(z)))
          return round(z,3), pval
      z,p= twoSampZ(M_mean,F_mean,0,M_std,F_std,no_of_M,no_of_F)
      print(z, p)

      416.38 0.0

[41]: import statsmodels.stats.weightstats as ws

      col1 = ws.DescrStatsW(df_weather.loc[df_weather['Precip Type'] == 'rain', 'Temperature (C)'])
      col2 = ws.DescrStatsW(df_weather.loc[df_weather['Precip Type'] == 'snow', 'Temperature (C)'])

      cm_obj = ws.CompareMeans(col1, col2)

      zstat, z_pval = cm_obj.ztest_ind(usevar='unequal')

      print(zstat.round(3), z_pval.round(3))

      416.38 0.0
```

5.2. Testing Hypothesis 2

For testing hypothesis 2 we can utilize the normaltest function within the scipy module. This function essentially checks whether our input variable is normally distributed or not and returns pvalue as an output. The testing confirms that the temperature variable skewness is within limits and therefore can be proceeded without log transformation.

```
[45]: from scipy.stats.mstats import normaltest

[46]: normaltest(df_weather['Temperature (C)'].values)

[46]: NormaltestResult(statistic=2781.31464301078, pvalue=0.0)
```

6. Conclusion and further steps

6.1. We can see that hypothesis testing has helped us identify whether our variables are meeting our hypotheses derived during the EDA stage.

6.2. Further, we can measure the correlation between the features and between the features and target variable to determine which features provide good measure of the trends or variance in target variable.

6.3. In next step we will one hot code the categorical variables based on the results of hypothesis testing and drop all those features which fail to add any meaningful information to the model.

6.4. Finally, we will treat all variables for skewness and scale them to prepare them for modelling.

Dataset Refrence –

https://www.kaggle.com/budincsevity/szeged-weather