Team 14 Project Proposal

Climate Change Prediction with Machine Learning

Description of project:

Purpose of this project aims to predict the yearly temperature change of a given city over a given time period. The output value should be numerically based on multiple extra factors like maximum temperature, minimum temperature, hPAAtSeaLevel, hPA, Humidity, Visibility, AverageWindSpeed, MaxSustainedWindSpeed, Fog and Precipitation

Team Members:

- Wooyoung Chung
- Mahavir Chandaliya
- Bhavana Gangula
- Jiahong Zhan

GitHub: *https://github.com/docmhvr/CMPE_257_PROJECT*

Dataset: https://en.tutiempo.net/climate

The datasets were obtained from tutiempo.net. We are using two datasets:

1) San Jose weather data containing the weather outcome of everyday from 2019 to 2021
2) Madrid weather data containing the weather outcome of everyday from 1991 to 1995

Description of the problem:

Our team found a climate dataset from 2019 to 2021 for San Jose and from 1991 to 1995 for Madrid. We want the training data to be able to predict future weather conditions. The specific process for the program will be based on the average annual temperature, annual average maximum temperature, average annual minimum temperature, total annual precipitation, annual average wind speed, number of days with rain, number of days with snow, number of days with storm,number of foggy days,number of days with tornado and number of days with hail as input, Then predict the certain weather condition for the next week, month or year. According to the prediction performance of the model, the prediction accuracy of the model is analyzed.

Potential methods:

Currently, we have pre-processing the dataset and made the dataset fit for a model. We are planning to conduct supervised learning to predict one of the weather conditions based on the patterns from other weather conditions of a specific timeline of Madrid and San Jose.

We are planning to do further data analysis by running PCA and perform normalization and scaling of the data to make it fit for running a ML model. Few models we will try to run on data will include Logistic regression, SVM, etc and after analysis of model performance and accuracy we will decide on the best model and optimize the model further for better prediction.

Preprocessing & Initial Findings:

We performed the following pre-processing steps on the data:

1) Data integration: combined the weather datasets of San Jose and Madrid
2) Data cleaning: remove missing data
3) Data reduction: remove unnecessary features
4) Data transformation: create new features from current ones and convert the unit of temperature

We performed the following visualizations on the dataset:

1) Line Plots
2) Bar Plots
3) Histograms
4) Heatmaps

We will start working on further pre-processing steps once we better understand the data and the interdependence between different columns in the dataset.

Based on the current results, we can see that we will need to do further data analysis before feeding the data to the model. The data needs to be scaled and normalized as well as there are a few outliers present in the data which need to be removed to give more accurate predictions.

In [66]:

```
import pandas as pd
import numpy as np
```

In [67]:

```
dfSanJose = pd.read_excel("724945-0.xlsx")
dfMadried = pd.read_excel("Madried.xlsx")
```

In [68]:

```
#Madrid Dataset
dfMadried
```

Out[68]:

	Y	M	D	T	TM	Tm	SLP	STP	H	PP	VV	V	VM	VG	FG	RA	SN	GR	TS	TR
0	1991	1	1	5.3	9.6	0.0	-	-	86	0	3.4	2.4	13	-	1	0	0	0	0	0
1	1991	1	2	2.6	6.4	0.0	-	-	88	0	3.7	4.1	11.1	-	1	0	0	0	0	0
2	1991	1	3	2.3	5.2	-1.0	-	-	87	0	2.6	2.0	9.4	-	1	0	0	0	0	0
3	1991	1	4	3.9	10.0	0.0	-	-	63	0	8.0	4.4	25.9	-	1	0	0	0	0	0
4	1991	1	5	2.9	10.4	-3.0	-	-	69	0	10.5	5.2	18.3	-	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1821	1995	12	27	9.7	11.0	6.5	1008.8	941.1	87	6.1	10.1	11.7	22.2	33.5	0	1	0	0	0	0
1822	1995	12	28	11.3	14.0	8.0	1012.8	945	78	0	12.4	14.6	22.2	40.7	0	1	0	0	0	0
1823	1995	12	29	9.0	10.2	7.6	1011.6	943.1	95	7.87	6.9	8.5	16.5	-	0	1	0	0	0	0
1824	1995	12	30	11.5	14.0	8.8	1001.6	935.1	91	21.08	10.3	17.8	29.4	53.5	0	1	0	0	0	0
1825	1995	12	31	11.3	14.0	8.0	1004.8	937.7	79	1.02	12.4	19.8	37	51.9	0	1	0	0	0	0

1826 rows × 20 columns

In [69]:

```
#San Jose Dataset
dfSanJose
```

Out[69]:

	Y	M	D	T	TM	Tm	SLP	STP	H	PP	VV	V	VM	VG	FG	RA	SN	GR	TS	TR
0	2019	1	1	8.9	13.3	2.8	1021.0	1019.3	29	0.00	16.1	13.7	25.9	42.4	0	0	0	0	0	0
1	2019	1	2	6.4	13.9	0.6	1023.9	1022.2	43	0.00	16.1	6.3	11.1	-	0	0	0	0	0	0
2	2019	1	3	7.1	14.4	0.6	1023.7	1022	54	0.00	16.1	5.0	14.8	-	0	0	0	0	0	0
3	2019	1	4	7.9	16.7	1.1	1017.6	1015.9	62	0.00	16.1	3.3	16.5	-	0	0	0	0	0	0
4	2019	1	5	10.7	16.7	1.7	1008.9	1007.2	72	0.00	15.8	20.9	44.6	59.4	0	1	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1091	2021	12	27	8.9	11.7	3.3	1014.8	1013.1	81	0.00	14.2	11.1	24.1	37	0	1	0	0	0	0
1092	2021	12	28	7.2	11.7	5.6	1012.7	1011	76	6.60	16.1	7.4	18.3	-	0	1	0	0	0	0
1093	2021	12	29	9.6	12.8	5.6	1007.0	1005.5	80	0.00	15.8	17.6	33.5	50	0	1	0	0	0	0
1094	2021	12	30	8.5	12.8	5.0	1011.5	1010	85	6.86	14.0	5.6	14.8	-	0	1	0	0	0	0
1095	2021	12	31	8.2	12.8	2.8	1012.3	1010.7	74	0.00	16.1	10.7	24.1	-	0	0	0	0	0	0

1096 rows × 20 columns

In [70]:

```
#combining the two datasets
frames = [dfSanJose, dfMadried]
df = pd.concat(frames)
df
```

Out[70]:

	Y	M	D	T	TM	Tm	SLP	STP	H	PP	VV	V	VM	VG	FG	RA	SN	GR	TS	TR
0	2019	1	1	8.9	13.3	2.8	1021.0	1019.3	29	0.0	16.1	13.7	25.9	42.4	0	0	0	0	0	0
1	2019	1	2	6.4	13.9	0.6	1023.9	1022.2	43	0.0	16.1	6.3	11.1	-	0	0	0	0	0	0
2	2019	1	3	7.1	14.4	0.6	1023.7	1022	54	0.0	16.1	5.0	14.8	-	0	0	0	0	0	0
3	2019	1	4	7.9	16.7	1.1	1017.6	1015.9	62	0.0	16.1	3.3	16.5	-	0	0	0	0	0	0

4	2019	1	5	10.7	16.7	1.7	1008.9	1007.2	72	0.0	15.8	20.9	44.6	59.4	0	1	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1821	1995	12	27	9.7	11.0	6.5	1008.8	941.1	87	6.1	10.1	11.7	22.2	33.5	0	1	0	0	0	0
1822	1995	12	28	11.3	14.0	8.0	1012.8	945	78	0	12.4	14.6	22.2	40.7	0	1	0	0	0	0
1823	1995	12	29	9.0	10.2	7.6	1011.6	943.1	95	7.87	6.9	8.5	16.5	-	0	1	0	0	0	0
1824	1995	12	30	11.5	14.0	8.8	1001.6	935.1	91	21.08	10.3	17.8	29.4	53.5	0	1	0	0	0	0
1825	1995	12	31	11.3	14.0	8.0	1004.8	937.7	79	1.02	12.4	19.8	37	51.9	0	1	0	0	0	0

2922 rows × 20 columns

In [71]:

```
#renaming columns to be more legible
df = df.rename(columns={"Y":"Year","M":"Month","D":"Day","T":"Temp","TM":"MaxTemp","Tm":"MinTemp","SLP":
```

In [72]:

```
#Number of Missing Variables
np.sum(df=='-')
```

Out[72]:

```
Year                       0
Month                      0
Day                        0
Temp                       0
MaxTemp                    0
MinTemp                    0
hPAAtSeaLevel            925
hPA                      279
Humidity                   4
TotalRainfall              2
Visibility                 0
AverageWindSpeed           0
MaxSustainedWindSpeed      5
MaxWindSpeed            2285
Fog                        0
Rain                       0
Snow                       0
?                          0
Storm                      0
StormWithRain              0
dtype: int64
```

In [73]:

```
#getting rid of null rows except for MaxWindSpeed
#Getting rid of MaxWindSpeed column because too many missing rows as well as not very useful
df = df.drop(["MaxWindSpeed"],axis=1)
df = df[df[:]!='-']
df = df.dropna(axis=0)
df
```

Out[73]:

	Year	Month	Day	Temp	MaxTemp	MinTemp	hPAAtSeaLevel	hPA	Humidity	TotalRainfall	Visibility	AverageW
0	2019	1	1	8.9	13.3	2.8	1021.0	1019.3	29	0.0	16.1	
1	2019	1	2	6.4	13.9	0.6	1023.9	1022.2	43	0.0	16.1	
2	2019	1	3	7.1	14.4	0.6	1023.7	1022	54	0.0	16.1	
3	2019	1	4	7.9	16.7	1.1	1017.6	1015.9	62	0.0	16.1	
4	2019	1	5	10.7	16.7	1.7	1008.9	1007.2	72	0.0	15.8	
...	...	...	...	...	...	...	...	...	...	...	...	
1821	1995	12	27	9.7	11.0	6.5	1008.8	941.1	87	6.1	10.1	
1822	1995	12	28	11.3	14.0	8.0	1012.8	945	78	0	12.4	
1823	1995	12	29	9.0	10.2	7.6	1011.6	943.1	95	7.87	6.9	
1824	1995	12	30	11.5	14.0	8.8	1001.6	935.1	91	21.08	10.3	
1825	1995	12	31	11.3	14.0	8.0	1004.8	937.7	79	1.02	12.4	

1989 rows × 19 columns

In [74]:

```
#dropping unneccessary columns
```

```
df = df.drop(["Year","Month","Day","?","StormWithRain","TotalRainfall","Storm"],axis=1)
#combining rain and snow as percipitation
df["Percipitation"] = df["Rain"]| df["Snow"]
df= df.drop(["Rain","Snow"],axis=1)
#Convert temp to F from C
df["Temp"] = df["Temp"]*9/5 + 32
df["MaxTemp"] = df["MaxTemp"]*9/5 + 32
df["MinTemp"] = df["MinTemp"]*9/5 + 32
```

In [75]:

```
df
```

Out[75]:

	Temp	MaxTemp	MinTemp	hPAAtSeaLevel	hPA	Humidity	Visibility	AverageWindSpeed	MaxSustainedWindSpee
0	48.02	55.94	37.04	1021.0	1019.3	29	16.1	13.7	25.
1	43.52	57.02	33.08	1023.9	1022.2	43	16.1	6.3	11.
2	44.78	57.92	33.08	1023.7	1022	54	16.1	5.0	14.
3	46.22	62.06	33.98	1017.6	1015.9	62	16.1	3.3	16.
4	51.26	62.06	35.06	1008.9	1007.2	72	15.8	20.9	44.
...	...	...	...	...	...	...	...	...	.
1821	49.46	51.80	43.70	1008.8	941.1	87	10.1	11.7	22.
1822	52.34	57.20	46.40	1012.8	945	78	12.4	14.6	22.
1823	48.20	50.36	45.68	1011.6	943.1	95	6.9	8.5	16.
1824	52.70	57.20	47.84	1001.6	935.1	91	10.3	17.8	29.
1825	52.34	57.20	46.40	1004.8	937.7	79	12.4	19.8	3

1989 rows × 11 columns

In [76]:

```
np.sum(df[:])
```

Out[76]:

```
Temp                    119954.34
MaxTemp                 144820.98
MinTemp                   97317.0
hPAAtSeaLevel           2022681.4
hPA                     1960627.2
Humidity                   121715
Visibility                27205.3
AverageWindSpeed          20296.1
MaxSustainedWindSpeed     46192.9
Fog                            75
Percipitation                 387
dtype: object
```

In [77]:

```
np.max(df[:])
```

```
C:\Users\MAHAVIR\anaconda3\lib\site-packages\numpy\core\fromnumeric.py:84: FutureWarning: In a future ve
rsion, DataFrame.max(axis=None) will return a scalar max over the entire DataFrame. To retain the old be
havior, use 'frame.max(axis=0)' or just 'frame.max()'
  return reduction(axis=axis, out=out, **passkwargs)
```

Out[77]:

```
Temp                      89.78
MaxTemp                  107.96
MinTemp                   78.98
hPAAtSeaLevel            1037.1
hPA                      1030.9
Humidity                     98
Visibility                 19.0
AverageWindSpeed           41.9
MaxSustainedWindSpeed      79.5
Fog                           1
Percipitation                 1
dtype: object
```

In [78]:

```
df.to_csv("PreprocessedDataset")
```

In [79]:

```
data = pd.read_csv("PreprocessedDataset")
```

In [80]:

```
data.head(10)
```

Out[80]:

	Unnamed: 0	Temp	MaxTemp	MinTemp	hPAAtSeaLevel	hPA	Humidity	Visibility	AverageWindSpeed	MaxSustainedW
0	0	48.02	55.94	37.04	1021.0	1019.3	29	16.1	13.7	
1	1	43.52	57.02	33.08	1023.9	1022.2	43	16.1	6.3	
2	2	44.78	57.92	33.08	1023.7	1022.0	54	16.1	5.0	
3	3	46.22	62.06	33.98	1017.6	1015.9	62	16.1	3.3	
4	4	51.26	62.06	35.06	1008.9	1007.2	72	15.8	20.9	
5	5	49.64	55.94	42.98	1012.4	1010.7	86	13.7	17.4	
6	6	54.50	62.06	42.98	1016.7	1014.6	83	14.8	12.6	
7	7	56.48	62.06	48.92	1017.9	1016.1	79	16.1	9.1	
8	8	57.92	60.98	51.08	1015.9	1014.1	80	15.3	16.9	
9	9	52.52	62.06	42.98	1020.7	1019.2	87	13.4	5.4	

In [81]:

```
data.info()
```

```
RangeIndex: 1989 entries, 0 to 1988
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Unnamed: 0            1989 non-null   int64  
 1   Temp                  1989 non-null   float64
 2   MaxTemp               1989 non-null   float64
 3   MinTemp               1989 non-null   float64
 4   hPAAtSeaLevel         1989 non-null   float64
 5   hPA                   1989 non-null   float64
 6   Humidity              1989 non-null   int64  
 7   Visibility            1989 non-null   float64
 8   AverageWindSpeed      1989 non-null   float64
 9   MaxSustainedWindSpeed 1989 non-null   float64
 10  Fog                   1989 non-null   int64  
 11  Percipitation         1989 non-null   int64  
dtypes: float64(8), int64(4)
memory usage: 186.6 KB
```

In [82]:

```
data.value_counts()
```

Out[82]:

```
Unnamed: 0  Temp   MaxTemp  MinTemp  hPAAtSeaLevel  hPA     Humidity  Visibility  AverageWindSpeed  MaxS
ustainedWindSpeed  Fog  Percipitation
0           48.02  55.94    37.04    1021.0         1019.3  29        16.1        13.7              25.9
0     0                1
1137        43.52  62.60    29.12    1026.5         956.9   65        10.0        4.1               14.8
0     0                1
1151        46.76  60.80    37.40    1017.2         948.9   79        7.9         4.1               33.5
1     0                1
1150        51.08  60.44    42.44    1015.8         947.6   75        11.9        20.4              25.9
0     0                1
1149        50.36  55.40    46.40    1012.2         944.4   71        11.4        21.9              35.2
0     1                1

..
653         77.36  93.92    59.00    1015.4         1013.8  29        16.1        10.0              22.2
0     0                1
652         72.14  89.96    53.96    1018.3         1016.5  50        16.1        8.1               22.2
0     0                1
651         69.08  87.08    53.96    1017.7         1016.0  47        16.1        8.1               24.1
0     0                1
650         67.10  84.92    53.96    1019.0         1017.3  50        16.1        6.5               25.9
0     0                1
1825        52.34  57.20    46.40    1004.8         937.7   79        12.4        19.8              37.0
0     1                1
Length: 1989, dtype: int64
```

In [83]:

```
data.columns
```

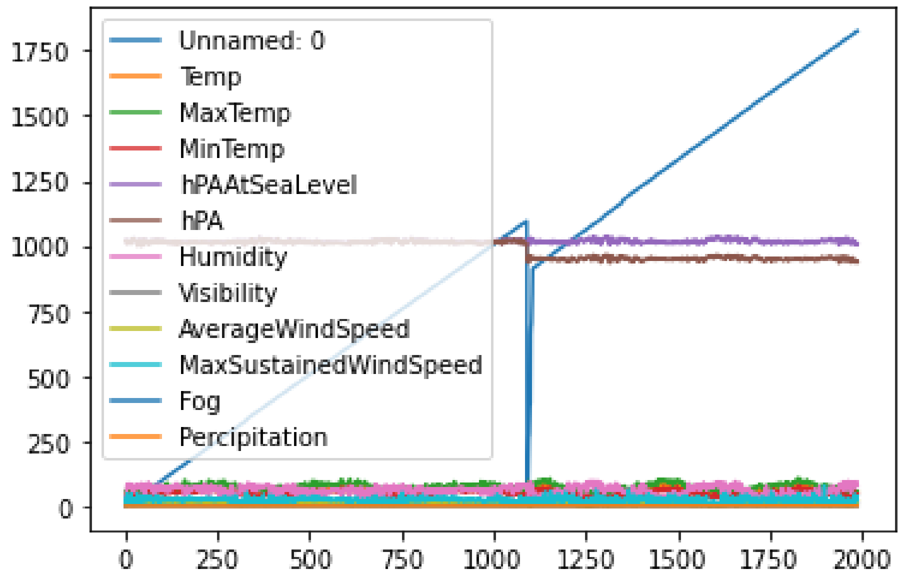
Out[83]:

```
Index(['Unnamed: 0', 'Temp', 'MaxTemp', 'MinTemp', 'hPAAtSeaLevel', 'hPA',
       'Humidity', 'Visibility', 'AverageWindSpeed', 'MaxSustainedWindSpeed',
       'Fog', 'Percipitation'],
      dtype='object')
```

In [84]:

```
data.plot()
```

Out[84]:

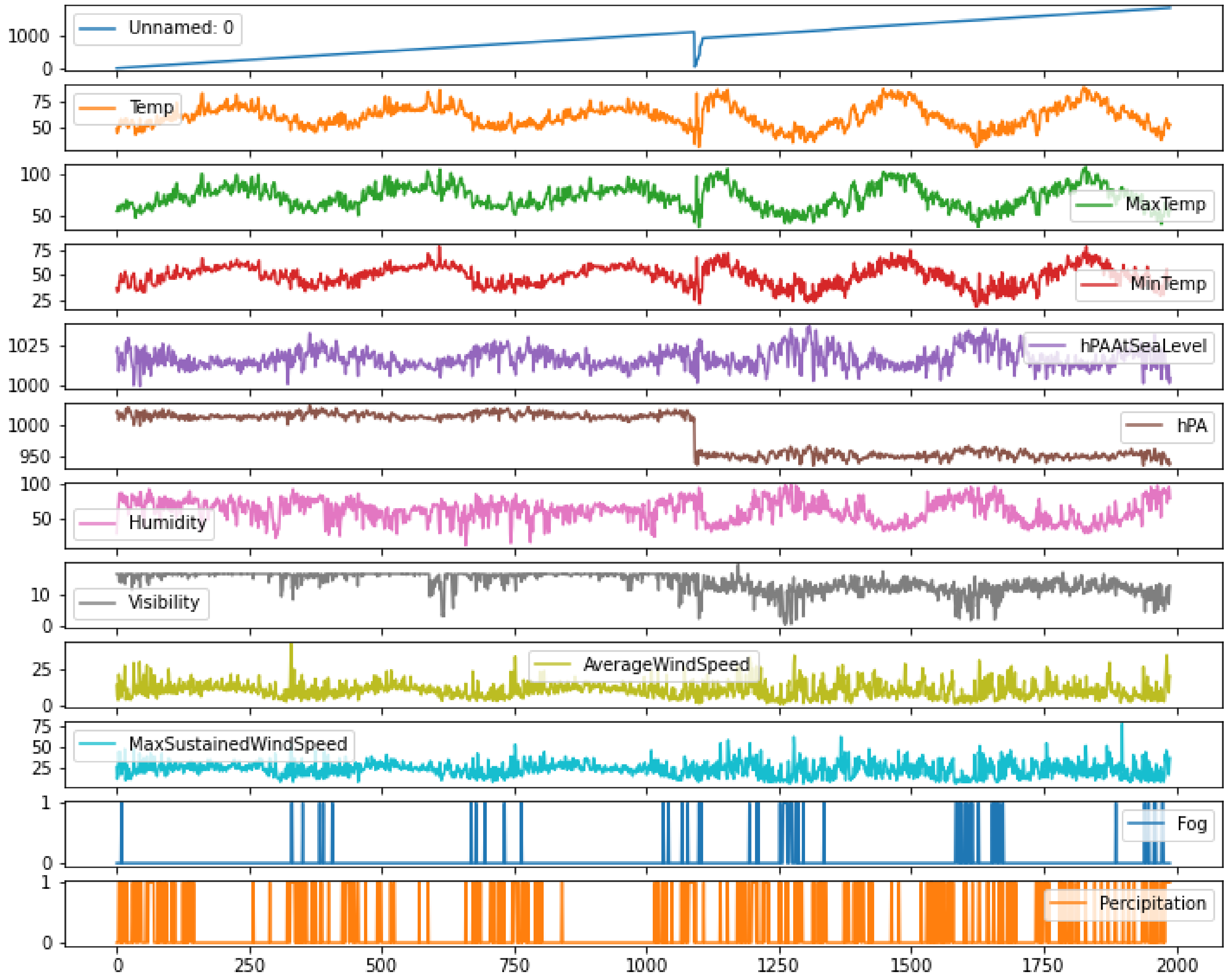


In [85]:

```
data.plot(subplots=True, figsize=(12,10))
```

Out[85]:

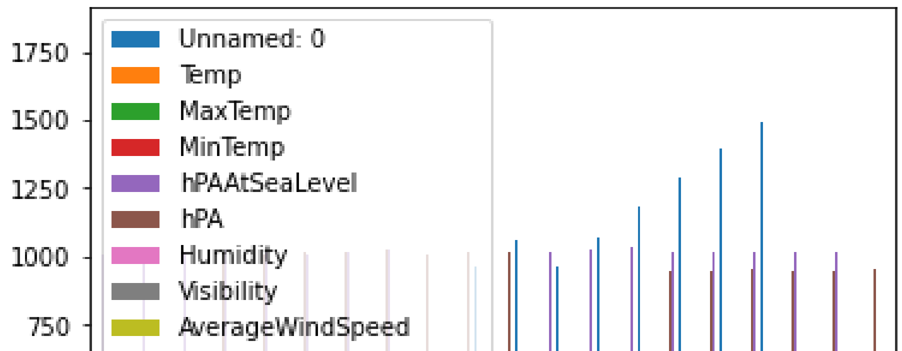
```
array([, , , ,
       , , , ,
       , , , ],
      dtype=object)
```

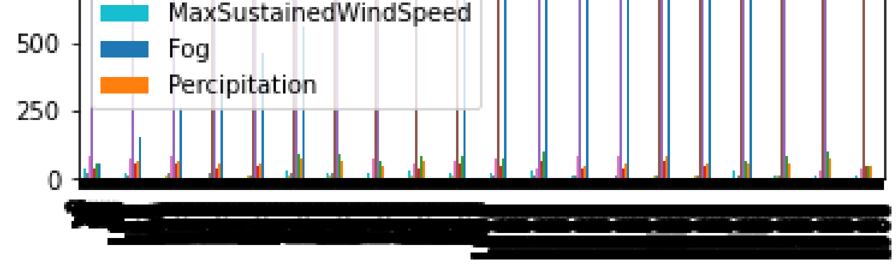


In [86]:

```
data.plot(kind="bar")
```

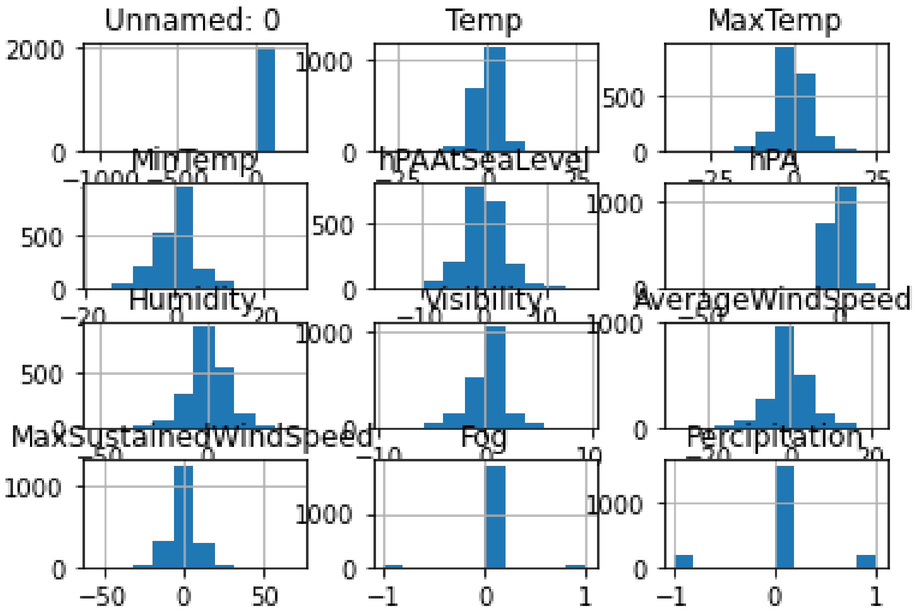
Out[86]:





In [88]:
```
data.diff().hist()
```

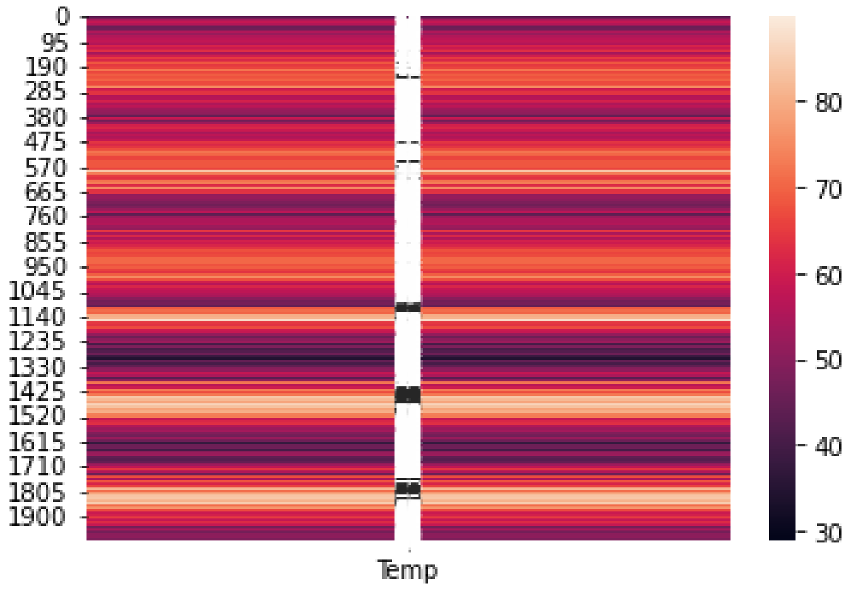
Out[88]:
```
array([[,
        ,
        ],
       [,
        ,
        ],
       [,
        ,
        ],
       [,
        ,
        ]], dtype=object)
```



In [89]:
```
import seaborn as sns
```

In [101…
```
sns.heatmap(data[['Temp']], annot=True)
```

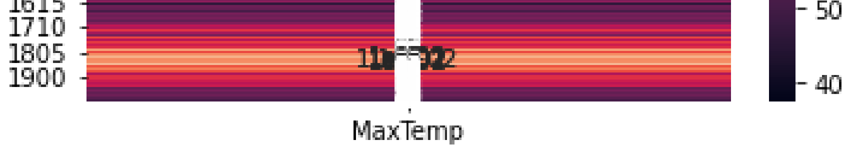
Out[101…



In [102…
```
sns.heatmap(data[['MaxTemp']], annot=True)
```

Out[102…

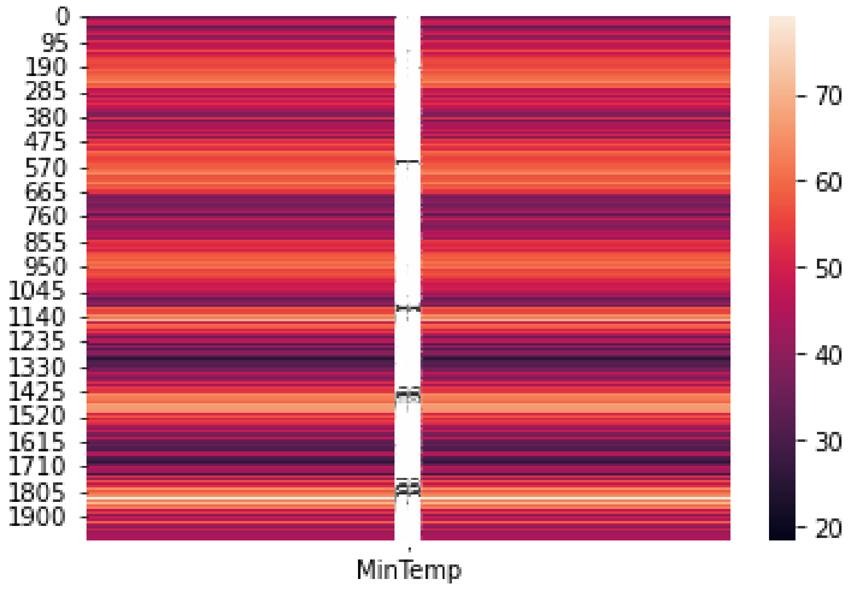
0
95
190
285
380
475
570
665
760
855
950
1045
1140
1235
1330
1425
1520
100
90
80
70
60



In [103…

```
sns.heatmap(data[['MinTemp']], annot=True)
```

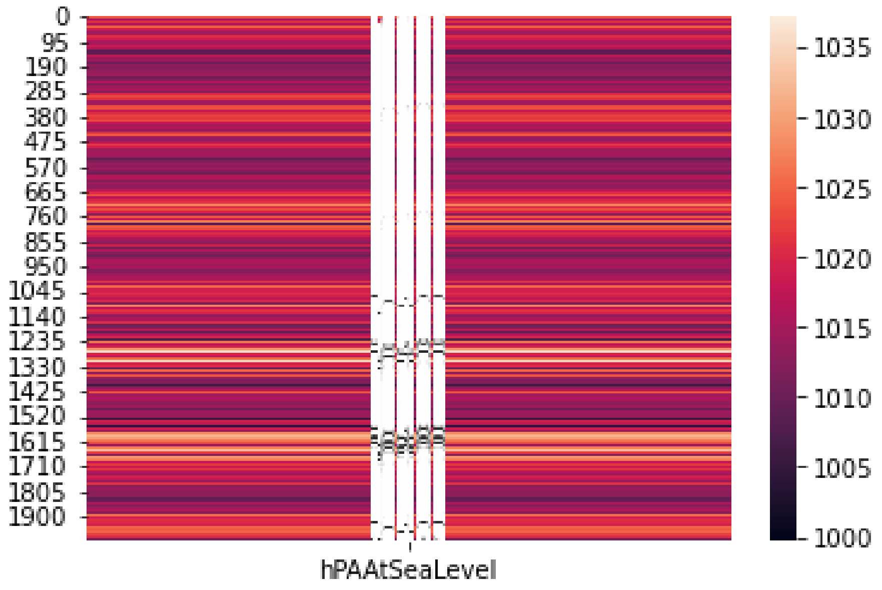
Out[103…



In [104…

```
sns.heatmap(data[['hPAAtSeaLevel']], annot=True)
```

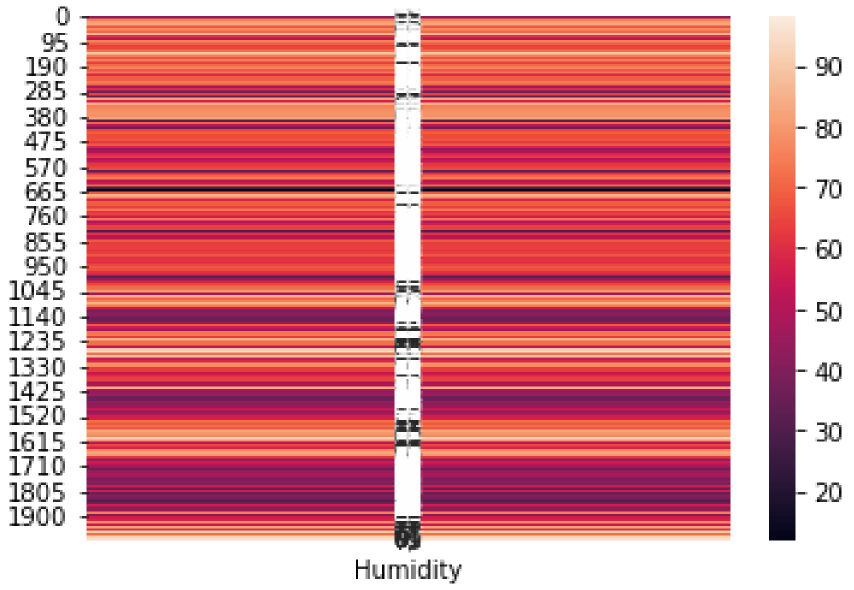
Out[104…



In [105…

```
sns.heatmap(data[['Humidity']], annot=True)
```

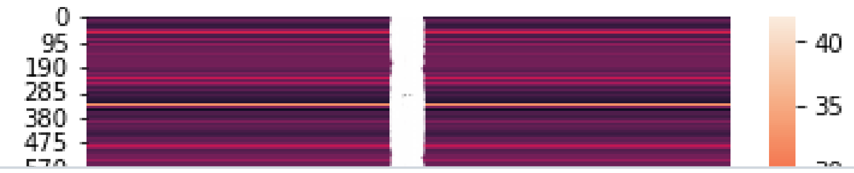
Out[105…



In [106…

```
sns.heatmap(data[['AverageWindSpeed']], annot=True)
```

Out[106…


0
95
190
285
380
475
40
35