Capstone 2

Model for Predicting Future Climate

By: Mehmet Ozer

Statement of the Problem

Global warming's dictionary definition is a gradual increase in the overall temperature of the earth's atmosphere and it is generally attributed to the greenhouse effect caused by increased levels of carbon dioxide, chlorofluorocarbons, and other pollutants. It has been a growing problem of whole humanity during the last couple decades.

The Paris agreement was signed by almost all United Nations countries by just a few exceptions. The major countries who did not sign the agreement were Turkey and Iran; however, just recently in 2020 the United States also opted out of the agreement because of the policies of the Republican Party and especially DOnald Trump who was the president at the time. It did not last long for the United States to stay out of the agreement. They rejoined in 2021. The purpose of the Paris agreement is to take necessary precautions worldwide to keep temperature rise under control. One of the major reasons of global warming is the greenhouse effect caused by gas emissions. The agreement focuses on reducing the gas emission but it does not force any country to meet a specific emission target.

According to United Nations Framework Convention on Climate Change the goals of Paris agreement are:

(a) Holding the increase in the global average temperature to well below 2 °C above pre-industrial levels and to pursue efforts to limit the temperature increase to 1.5 °C above pre-industrial levels, recognizing that this would significantly reduce the risks and impacts of climate change;

(b) Increasing the ability to adapt to the adverse impacts of climate change and foster climate resilience and low greenhouse gas emissions development, in a manner that does not threaten food production;

(c) Making finance flows consistent with a pathway towards low greenhouse gas emissions and climate-resilient development.

Concept of Capstone Project

The goal of this capstone project is to identify if the global warming phenomena is real and how much of change we are expecting to see during the next decades. Despite the discussions of if global warming exists this project fouses on also predicting the future climate for a chosen location in the world.

Business Case

An accurate prediction system can be extremely useful for a variety of reasons for the whole world. The fertile land and conditions of growing crops is getting limited every year because of global warming. Many sea products are facing extinction right now or in the near future. Without a good solid prediction it will be real difficult for the world to be prepared for what is coming. We may see too many business fields going bankrupt if they do not prepare themselves for the changing world climate. This may be very real for those companies that rely on crops, sea food and animal husbandry.

Summary of Capstone Project

The data used in this study were gathered from https://crudata.uea.ac.uk/ which is a website that stores invaluable information for many researchers who may be interested in climate studies. To find a meaningful location for this study I wanted to work with a location in the North Pole because I thought if there was any change it had to be more visible in the poles. There were more than one location's data available on the website; however, I chose Ilulissat because it was the only location which had the available data going back to 1824.

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	1824	-161	-196	-160	-59	15	57	81	42	7	-57	-90	-126
1	1825	-999	-999	-999	-999	-999	-999	54	-999	-999	-63	-136	-180
2	1826	-184	-999	-204	-234	-31	-6	-999	-999	6	-51	-60	-118
3	1827	-186	-74	-197	-105	-22	-999	-999	-999	-999	-999	-999	-999
4	1828	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999

Data Wrangling

The raw data was in wide format and the first few columns had too many missing values. I filled them using the rolling average of 6 months. For the first row I used the following 6 months mean value to fill them.

```
for i in df_ilu.columns:
    df_ilu[i] = df_ilu[i].fillna(df_ilu[i].rolling(6, min_periods=1).mean())
```

The original dataset did not habe column names in place so I had to put column names.

```
# Change column names
df_ilu.columns = ['Year', 'Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']
```

The data was in wide format but for time series analysis we need to have the dataset in long format. So I converted the wide format to long using Python's melt function.

```
# Convert wide to long
df_ilu_long = pd.melt(df_ilu, id_vars='Year', value_vars=('Jan',
                                                           'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul',
                                                           'Aug', 'Sep', 'Oct', 'Nov', 'Dec'))
```

The next thing I did was to put in a day into the time because they were in MM/YYYY format and I converted them to MM/DD/YYYY for analysis to be easier to recognize the months.

```
# Adding a random day to each month to have it mm/dd/yyyy format

df_ilu_long['Date'] = df_ilu_long['variable'].astype(str) + '/1/' + df_ilu_long['Year'].astype(str)

```

Then I converted this “Date” column into datetime object.

```
# Converting to datetime object
df_ilu_long['Date'] = df_ilu_long['Date'].astype('datetime64[ns]')
```

The temperature data in the daset was in Celcius but it was multiplied by 10. So instead of -10 we had -100 in the dataset. To fix this I created a loop to have the values back to their originals and I set this variable as the index.

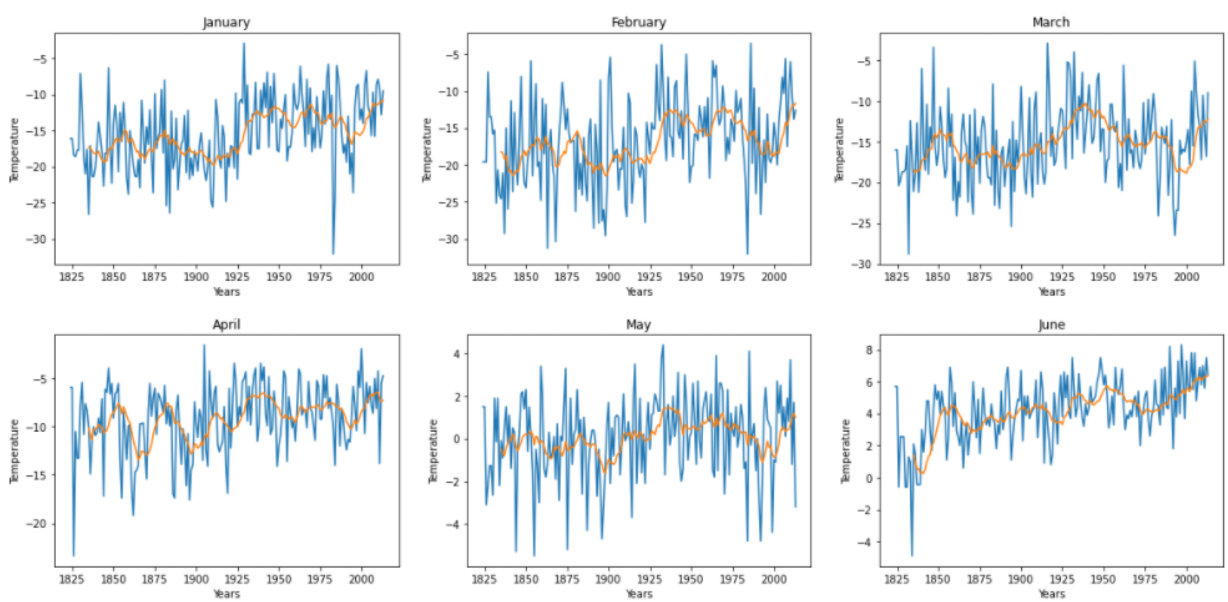
```
# Since the temperature is given as multiplied by 10 here i am
# dividing each temprature by 10 to get the actual value
for i in range(0, len(df_ilu_long)):
    df_ilu_long['Temp'][i] = df_ilu_long['Temp'][i]/10
```

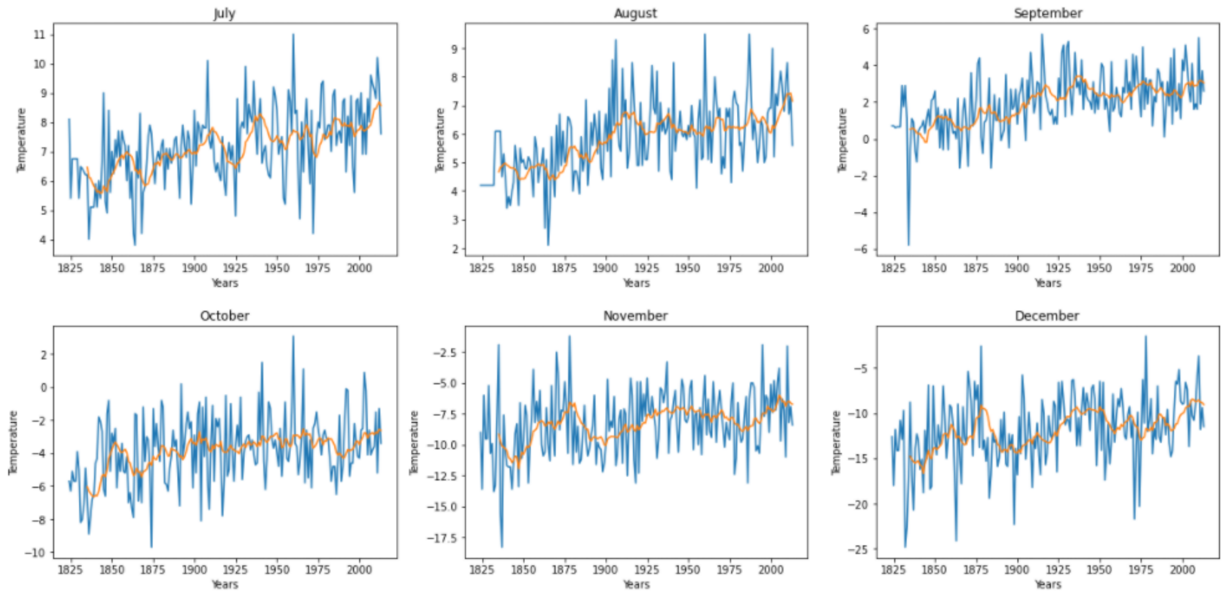
Exploratory Analysis and Exploratory Machine Learning

For the exploratory analysis first thing I did was looking at the monthly changes over time. For that purpose I used subplots of monts in a single plot. The code I wrote and the output are below.

```
fig, axs = plt.subplots(4, 3)

fig.set_figheight(20)
fig.set_figwidth(20)
fig.subplots_adjust(hspace=.3)


axs[0, 0].plot(df_ilu['Year'], jan)
axs[0, 0].set_title('January')
jan_moving_average = jan.rolling(window=12).mean()
axs[0, 0].plot(df_ilu['Year'], jan_moving_average)

axs[0, 1].plot(df_ilu['Year'], feb)
axs[0, 1].set_title('February')
feb_moving_average = feb.rolling(window=12).mean()
axs[0, 1].plot(df_ilu['Year'], feb_moving_average)

axs[0, 2].plot(df_ilu['Year'], mar)
axs[0, 2].set_title('March')
mar_moving_average = mar.rolling(window=12).mean()
axs[0, 2].plot(df_ilu['Year'], mar_moving_average)

axs[1, 0].plot(df_ilu['Year'], apr)
axs[1, 0].set_title('April')
apr_moving_average = apr.rolling(window=12).mean()
axs[1, 0].plot(df_ilu['Year'], apr_moving_average)

axs[1, 1].plot(df_ilu['Year'], may)
axs[1, 1].set_title('May')
may_moving_average = may.rolling(window=12).mean()
axs[1, 1].plot(df_ilu['Year'], may_moving_average)

axs[1, 2].plot(df_ilu['Year'], jun)
axs[1, 2].set_title('June')
jun_moving_average = jun.rolling(window=12).mean()
axs[1, 2].plot(df_ilu['Year'], jun_moving_average)

axs[2, 0].plot(df_ilu['Year'], jul)
axs[2, 0].set_title('July')
jul_moving_average = jul.rolling(window=12).mean()
axs[2, 0].plot(df_ilu['Year'], jul_moving_average)

axs[2, 1].plot(df_ilu['Year'], aug)
axs[2, 1].set_title('August')
aug_moving_average = aug.rolling(window=12).mean()
axs[2, 1].plot(df_ilu['Year'], aug_moving_average)

axs[2, 2].plot(df_ilu['Year'], sep)
axs[2, 2].set_title('September')
sep_moving_average = sep.rolling(window=12).mean()
axs[2, 2].plot(df_ilu['Year'], sep_moving_average)
```

```python
axs[3, 0].plot(df_ilu['Year'], oct)
axs[3, 0].set_title('October')
oct_moving_average = oct.rolling(window=12).mean()
axs[3, 0].plot(df_ilu['Year'], oct_moving_average)

axs[3, 1].plot(df_ilu['Year'], nov)
axs[3, 1].set_title('November')
nov_moving_average = nov.rolling(window=12).mean()
axs[3, 1].plot(df_ilu['Year'], nov_moving_average)

axs[3, 2].plot(df_ilu['Year'], dec)
axs[3, 2].set_title('December')
dec_moving_average = dec.rolling(window=12).mean()
axs[3, 2].plot(df_ilu['Year'], dec_moving_average)


for ax in axs.flat:
    ax.set(xlabel='Years', ylabel='Temperature')
```

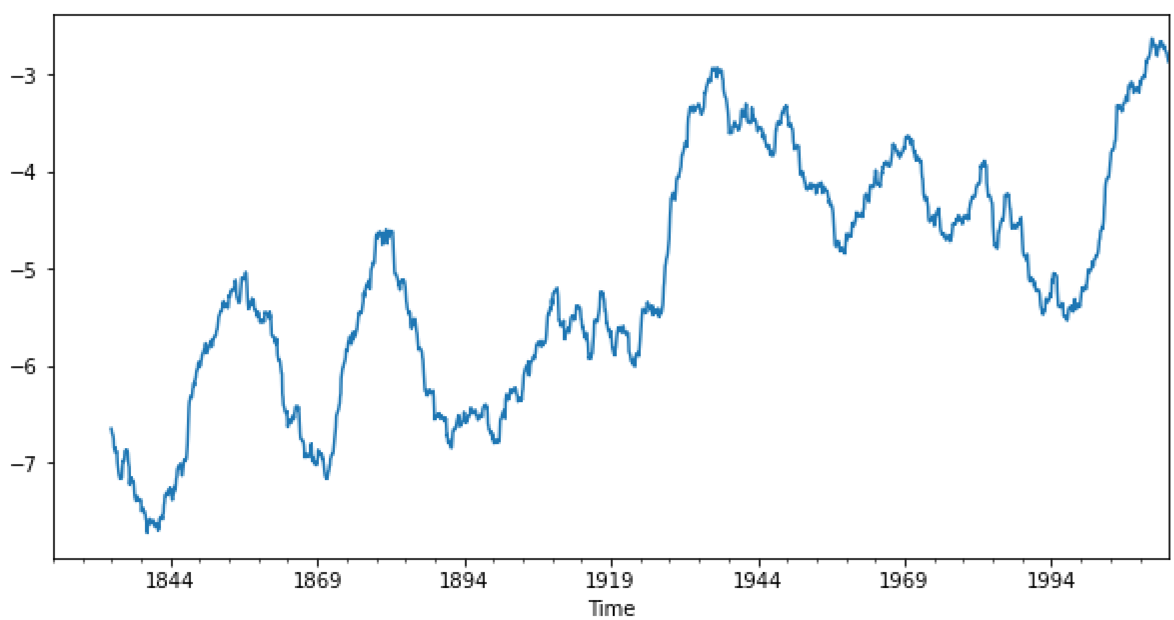




After plotting for each month I also looked that moving average and moving median of the temperature data.

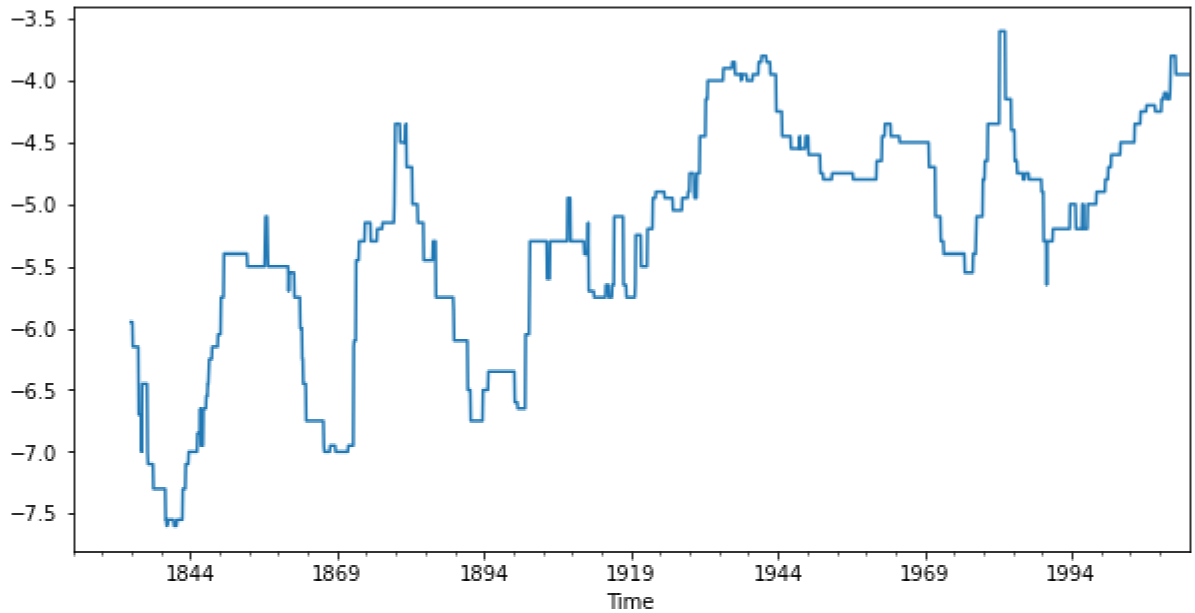
```
# Creting and plotting the rolling mean and median
df_ilu_long.Temp.rolling(window=120).mean().plot(figsize=(10, 5))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f70b3b3640>
```



```
1  df_ilu_long.Temp.rolling(window=120).median().plot(figsize=(10, 5))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f70b480490>
```



Modeling

Several time series analysis techniques were used to create the modeling. For them to be used first the train_test_split function from the Sklearn package was used to split the data into train and test sets. Test size was determined as 20% of the dataset. First model used was the base model which produced RMSE score of 5.76. Then I used Arima to come to a conclusion. Arima produced 3.65. To determine the p,d,q values in Arima I used ACF and PACF along with the code block below.

```python
# Finding the best p,d,q values for the analysis
p=d=q=range(0, 5)
pdq = list(itertools.product(p,d,q))
```

```python
for param in pdq:
    try:
        temp_model = ARIMA(train, order=param)
        temp_model_fit = temp_model.fit()
        print(param, temp_model_fit.aic)
    except:
        continue
```

The results of Arima produced the following AIC and RMSE scores:

```python
temp_model = ARIMA(train, order=(3,1,4))
temp_model_fit = temp_model.fit()
print('AIC = ',temp_model_fit.aic)
temp_forecast = temp_model_fit.forecast(steps=457)[0]
print('ARIMA RMSE = ',np.sqrt(mean_squared_error(test, temp_forecast)))
```

```
AIC =  9411.378020150167
ARIMA RMSE =  3.6538495814733234
```

Next I used Exponential Smoothing Halt Winters test

```python
model = ExponentialSmoothing (train, trend='add', seasonal='add', seasonal_periods=12)
fitted = model.fit()
#.fittedvalues
#forecast = model_fit.forecast(steps=457)[0]
forecast = fitted.forecast(steps=457)

```

```python
]: print('Halt Winters RMSE = ', np.sqrt(mean_squared_error(test, forecast)))
```

```
Halt Winters RMSE =  3.4193666994031324
```

Lastly, I used Facebook Prophet to end the analysis. FB Prophet produced the best RMSE score which was 3.321 and it became the technique to use for prediction of future temperature.

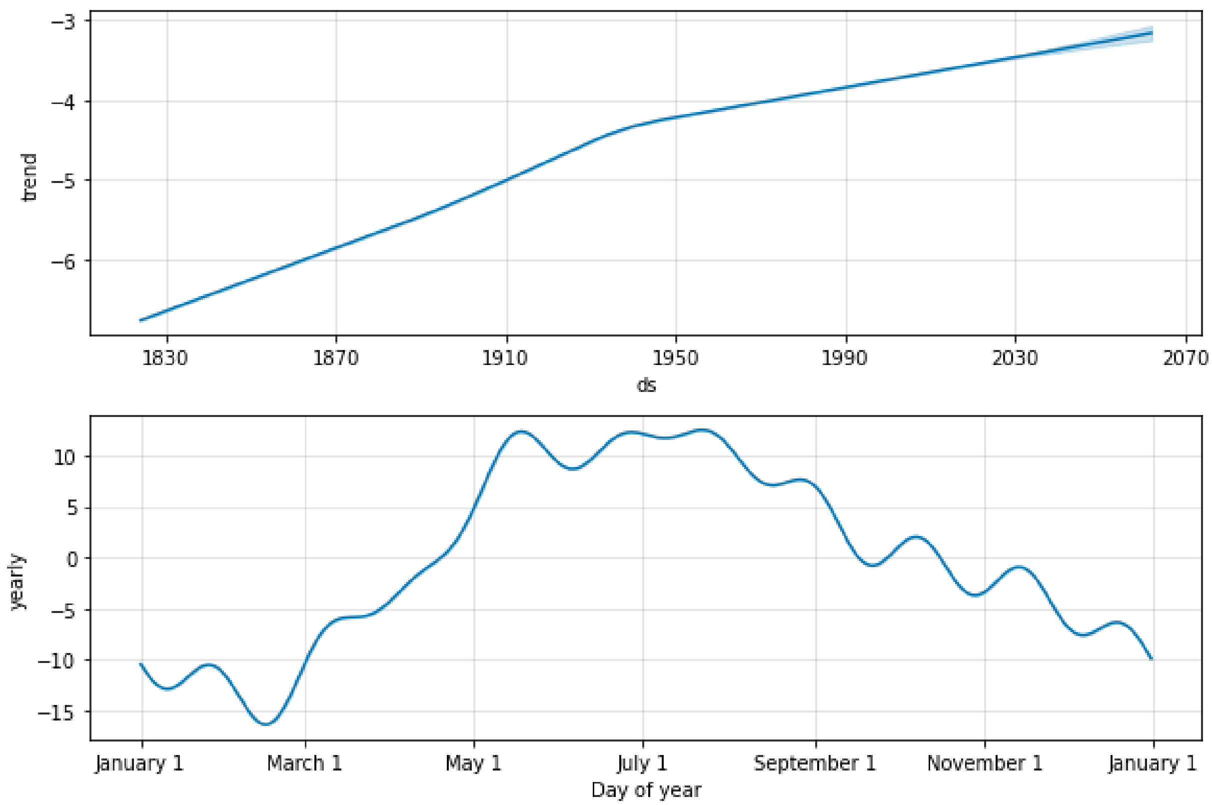
```python
# Fitting the model with the train data
model=Prophet(seasonality_mode='additive', seasonality_prior_scale=0.1,
              changepoint_prior_scale=0.03,
              daily_seasonality=False, weekly_seasonality=False,
              holidays=None )
model.fit(train)

# Making predictions
future = model.make_future_dataframe(periods=457+120, freq='MS')
forecast = model.predict(future)
```

Conclusion

I created the predictions using the whole dataset by the following code block.

```python
# Making predictions with the whole data
model=Prophet(seasonality_mode='additive', seasonality_prior_scale=0.1,
              changepoint_prior_scale=0.03,   daily_seasonality=False,
              weekly_seasonality=False,  holidays=None )
model.fit(data)
future = model.make_future_dataframe(periods=457+120, freq='MS')
forecast = model.predict(future)
```

```
1 model.plot_components(forecast)
```



As seen on the graph and predicted value for Jan 2062 from the table the mean temperature of Ilulissat for January 1824 was -17.32 C and for January 2062 I am predicting with this model that it will be -13.5 C. This is almost 4 degrees of increase. We can conclude by saying that global warming will be affecting the world in an enormous way during the upcoming decades.