Anim Yusufbhai Malvat
110024150
malvat@uwindsor.ca
School of Computer Science

Dhwani Gurjar
110022182
gurjar2@uwindsor.ca
School of Computer Science

Siddhi Patel
110009085
patel2e2@uwindsor.ca
School of Computer Science

Satyapalsinh Solanki
110022029
solanki@uwindsor.ca
School of Computer Science

Project Title: Data mining: Data prediction of increase in gas emissions in Ontario

Course: COMP 8157 – Advanced Database Topics

Data source: Canada's official national greenhouse gas inventory

https://www.canada.ca/en/environment-climate-change/services/climate-change/greenhouse-gas-emissions/inventory.html

GitHub: https://github.com/malvat/DataMiningGasEmissions

Milestone II

As we mentioned in milestone I, we have successfully collected the data and processed it according to our need. Now It is time to model our prediction.

NOTE: we have used **Python Jupyter Notebook** for our project, as it provides aesthetic visuals for presentation purposes and works as an IDE at the same time.

Before we begin, quick recap to what we are doing and what we have done so far:

- Collect data
- Process the data according to the need
- Visualize the dataset to find trends
- Choose a model to map and predict
- Train parameters to improve model
- Predict results and write a report

Choosing a model for analysis

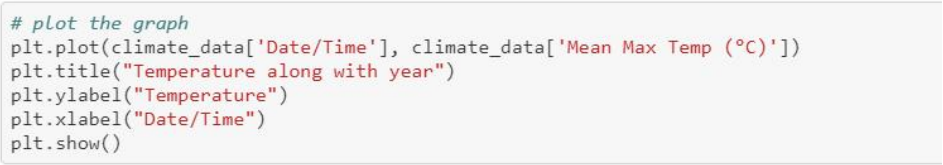
So, we now must choose what model we are going to use, there are quite a few options and we have investigated all of them to find the best for our application.

- KNN (k nearest neighbors) classification
- Linear regression model
- Shrinkage
- Sparsity
- Classification

From the basic internet research, we have observed that KNN and classification are two models that will not be helpful to our need and therefore, we are not going to include them in our project. But for the rest of the models, we will try to create the model.

Before we begin creating models

First, let us retrieve the temperature data, after doing the same procedure for the temperature data, we can now visualize the data to see some trends.



```
# plot the graph
plt.plot(climate_data['Date/Time'], climate_data['Mean Max Temp (°C)'])
plt.title("Temperature along with year")
plt.ylabel("Temperature")
plt.xlabel("Date/Time")
plt.show()
```

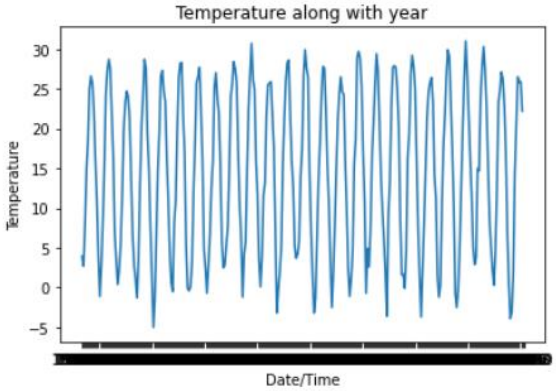
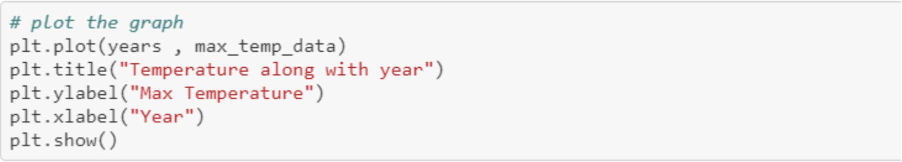


Figure 1 Change in temperature

After visualization, we can see that there is not a huge increase or decrease in temperature neither any trend can be seen. To make things clear, we thought to visualize only maximum temperature in a year, results were found as Figure 2. Maximum temperature.



```
# plot the graph
plt.plot(years , max_temp_data)
plt.title("Temperature along with year")
plt.ylabel("Max Temperature")
plt.xlabel("Year")
plt.show()
```

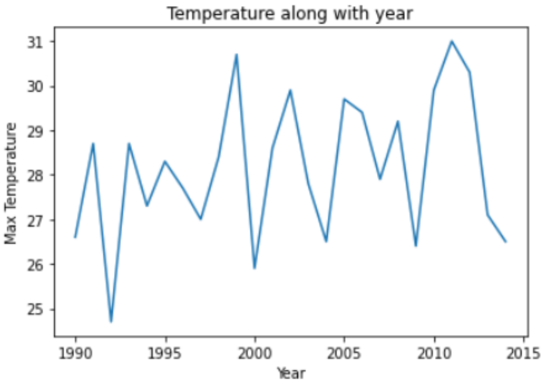
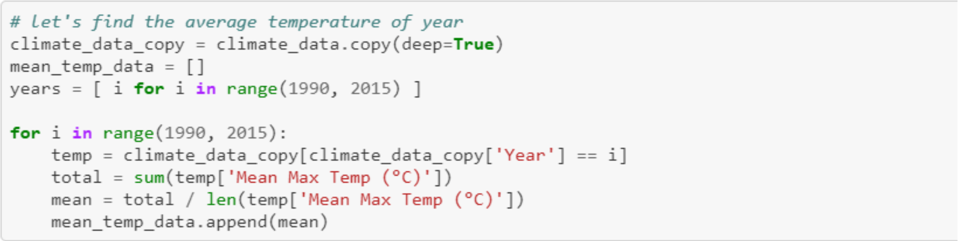


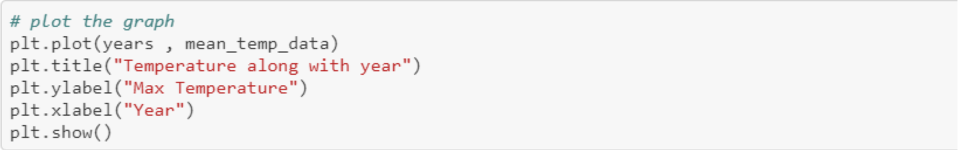
Figure 2 Maximum temperature

After visualizing maximum temperature, we can observe a slight increase in temperature through all these years, let us try to find the average temperature through out the year, and see the trend.



```
# let's find the average temperature of year
climate_data_copy = climate_data.copy(deep=True)
mean_temp_data = []
years = [ i for i in range(1990, 2015) ]

for i in range(1990, 2015):
    temp = climate_data_copy[climate_data_copy['Year'] == i]
    total = sum(temp['Mean Max Temp (°C)'])
    mean = total / len(temp['Mean Max Temp (°C)'])
    mean_temp_data.append(mean)
```



```
# plot the graph
plt.plot(years , mean_temp_data)
plt.title("Temperature along with year")
plt.ylabel("Max Temperature")
plt.xlabel("Year")
plt.show()
```

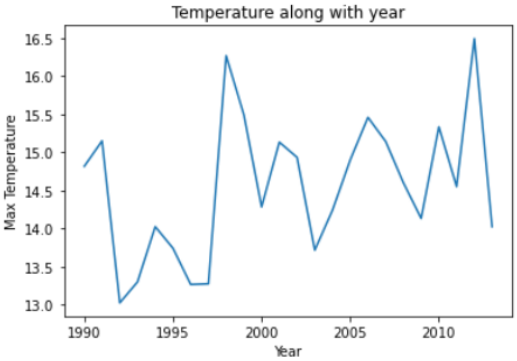


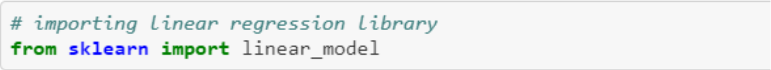
Figure 3 Average temperature

We can see there's fluctuation in temperature, but there is gradual increase all through these years. Hence, we can say there is increase in temperature overall.

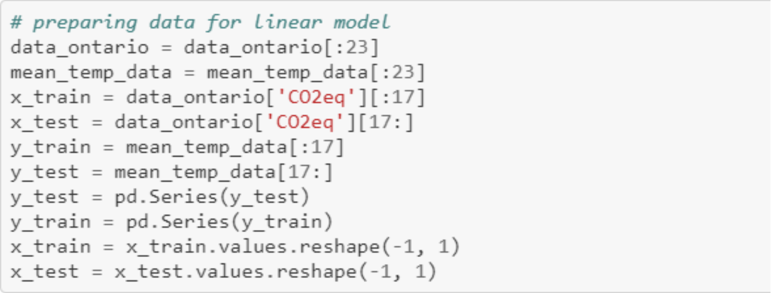
Linear regression model

Let us try to create a linear regression model for the data we have processed so far.

Before, we create our model we need to separate the data for training and testing data.



```
# importing linear regression library
from sklearn import linear_model
```

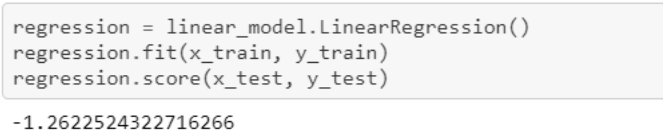


```
# preparing data for linear model
data_ontario = data_ontario[:23]
mean_temp_data = mean_temp_data[:23]
x_train = data_ontario['CO2eq'][:17]
x_test = data_ontario['CO2eq'][17:]
y_train = mean_temp_data[:17]
y_test = mean_temp_data[17:]
y_test = pd.Series(y_test)
y_train = pd.Series(y_train)
x_train = x_train.values.reshape(-1, 1)
x_test = x_test.values.reshape(-1, 1)
```

Figure 4 Preparing data for creating model

Let us create a model to check if regression model will work for us or not.

NOTE: we have used **sklearn** library for creating the model



```
regression = linear_model.LinearRegression()
regression.fit(x_train, y_train)
regression.score(x_test, y_test)
```

```
-1.2622524322716266
```

Figure 5 Linear regression model

We are getting a negative score; this can mean to things:

- There is no relation between data
- The data is too noisy or not consistent

Next steps

Clean and make the data more consistent and use other models to test our theory.