Predicting the Impact of Climate Change on Crop Yields

12.19.2021

—

Jesus Pena, Muizz Mullani, Raheel Bhimani

CIS 545 - Big Data Analytics

Colab Notebook

Overview

Crop yield production is an escalating agricultural problem as the human population continues to grow coupled with understanding of the impacts of global warming. According to the Natural Resources Defense Council, the global annual temperature has increased by 0.32 degrees Fahrenheit per decade since the 1980s and 5 of the warmest years on record have occurred since 2015. Further, even though each region in the world has varying dietary preferences, the simple ingredients are generally the same (i.e., basic crops like wheat, corn, rice, etc.).

Considering the impact that climate change plays on crop yield production, we decided to take crop yield production data from Our World in Data, climate data from National Oceanic and Atmospheric Administration, and CO2 Emissions data from Our World in Data to train and produce models to predict the impact of climate change on crop yields from 1961 to 2018 across 9 countries.

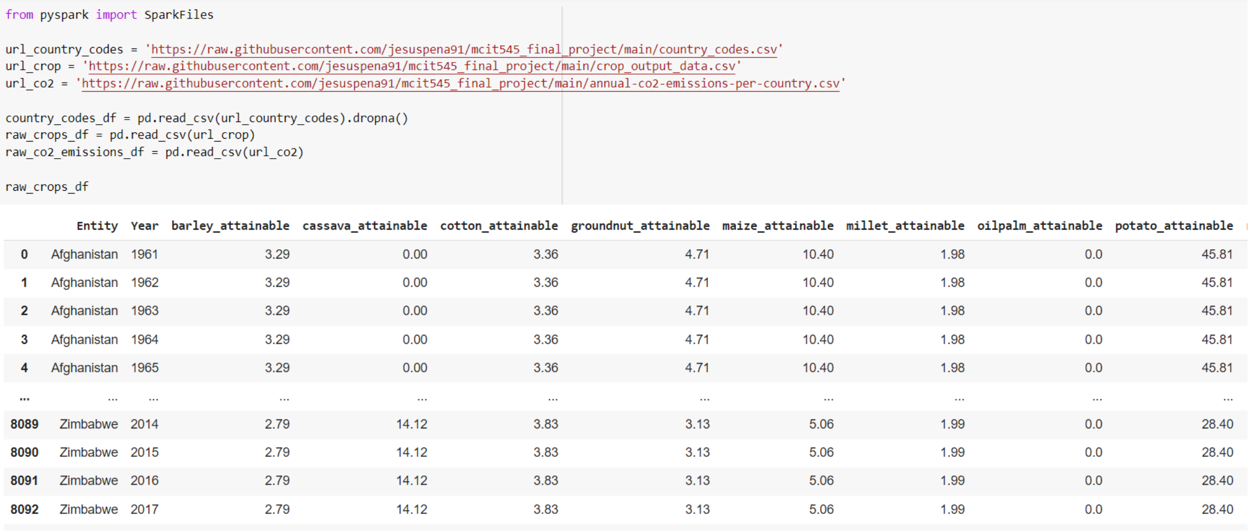
Regression analysis includes a set of machine learning methods that allow us to predict a continuous outcome variable (y) based on the value of one or multiple predictor variables (x). In this project, we used the following models:

- Linear Regression Analysis
- Random Forest Regressor

The models were evaluated using Mean Squared Error (MSE) and the R^2 (coefficient of determination). Both of these metrics will represent the proportion of variance for crops in the models. The MSE metric is a statistical measure that shows how close a fitted line is to data points. If the MSE is closer to 0 then the model is a perfect representation of the data, but this is generally unlikely. Our goal is to minimize the MSE metric. The R^2 metric is a statistical measure between 0 and 1 that shows how similar a regression line is to the underlying data. If the R^2 metric is closer to 0 then the model is not a good representation of the variance. Conversely, if the R^2 metric is closer to 1 then the model is a good representation of the variance.

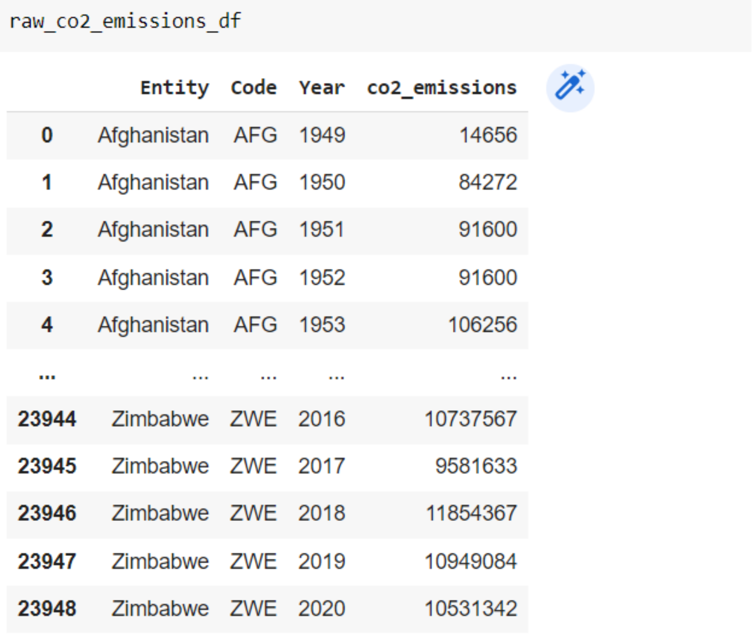
Evaluating and Analyzing the Data

After importing the required libraries, we read in the crop and CO2 emissions raw data.



```python
from pyspark import SparkFiles

url_country_codes = 'https://raw.githubusercontent.com/jesuspena91/mcit545_final_project/main/country_codes.csv'
url_crop = 'https://raw.githubusercontent.com/jesuspena91/mcit545_final_project/main/crop_output_data.csv'
url_co2 = 'https://raw.githubusercontent.com/jesuspena91/mcit545_final_project/main/annual-co2-emissions-per-country.csv'

country_codes_df = pd.read_csv(url_country_codes).dropna()
raw_crops_df = pd.read_csv(url_crop)
raw_co2_emissions_df = pd.read_csv(url_co2)

raw_crops_df
```

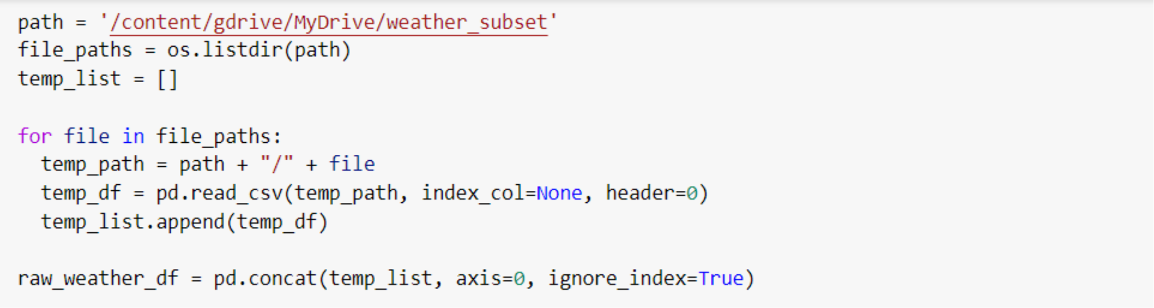
	Entity	Year	barley_attainable	cassava_attainable	cotton_attainable	groundnut_attainable	maize_attainable	millet_attainable	oilpalm_attainable	potato_attainable
0	Afghanistan	1961	3.29	0.00	3.36	4.71	10.40	1.98	0.0	45.81
1	Afghanistan	1962	3.29	0.00	3.36	4.71	10.40	1.98	0.0	45.81
2	Afghanistan	1963	3.29	0.00	3.36	4.71	10.40	1.98	0.0	45.81
3	Afghanistan	1964	3.29	0.00	3.36	4.71	10.40	1.98	0.0	45.81
4	Afghanistan	1965	3.29	0.00	3.36	4.71	10.40	1.98	0.0	45.81
...	...	...	...	...	...	...	...	...	...	...
8089	Zimbabwe	2014	2.79	14.12	3.83	3.13	5.06	1.99	0.0	28.40
8090	Zimbabwe	2015	2.79	14.12	3.83	3.13	5.06	1.99	0.0	28.40
8091	Zimbabwe	2016	2.79	14.12	3.83	3.13	5.06	1.99	0.0	28.40
8092	Zimbabwe	2017	2.79	14.12	3.83	3.13	5.06	1.99	0.0	28.40



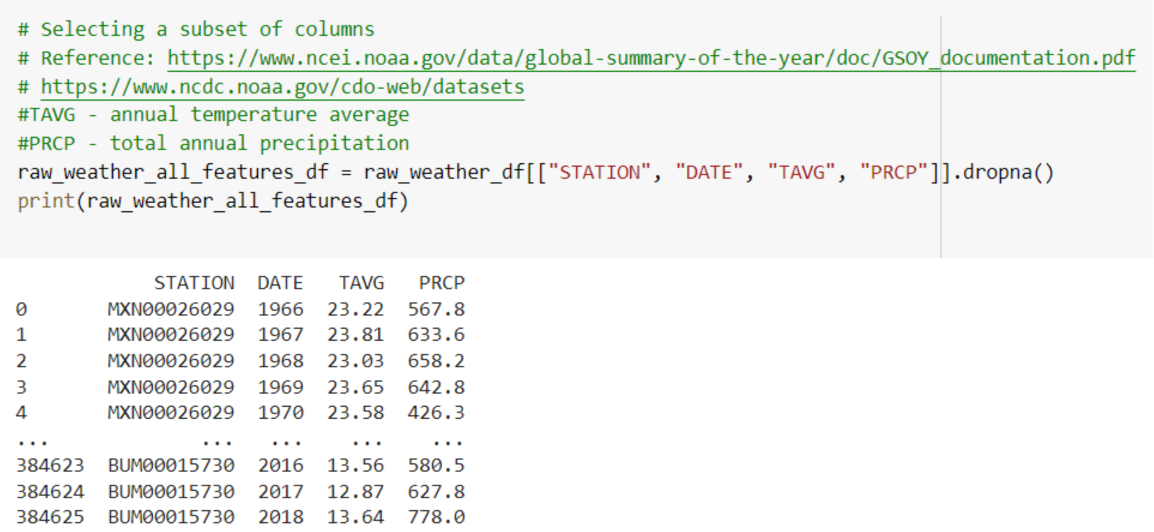
```python
raw_co2_emissions_df
```

	Entity	Code	Year	co2_emissions
0	Afghanistan	AFG	1949	14656
1	Afghanistan	AFG	1950	84272
2	Afghanistan	AFG	1951	91600
3	Afghanistan	AFG	1952	91600
4	Afghanistan	AFG	1953	106256
...	...	...	...	...
23944	Zimbabwe	ZWE	2016	10737567
23945	Zimbabwe	ZWE	2017	9581633
23946	Zimbabwe	ZWE	2018	11854367
23947	Zimbabwe	ZWE	2019	10949084
23948	Zimbabwe	ZWE	2020	10531342

Then, we read in the climate data into a dataframe. The data looks organized but we dropped some of the columns that will not be of use in our analysis. The remaining columns include: Station ID, Year, Average Temperature, and Total Annual Precipitation. We printed out a preview of the dataframe.



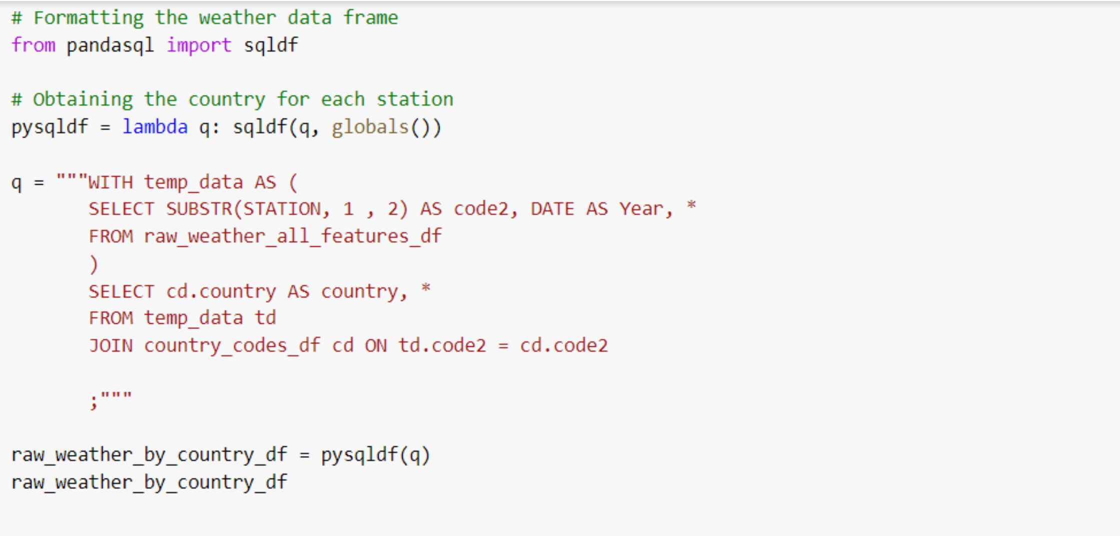
```
path = '/content/gdrive/MyDrive/weather_subset'
file_paths = os.listdir(path)
temp_list = []

for file in file_paths:
  temp_path = path + "/" + file
  temp_df = pd.read_csv(temp_path, index_col=None, header=0)
  temp_list.append(temp_df)

raw_weather_df = pd.concat(temp_list, axis=0, ignore_index=True)
```



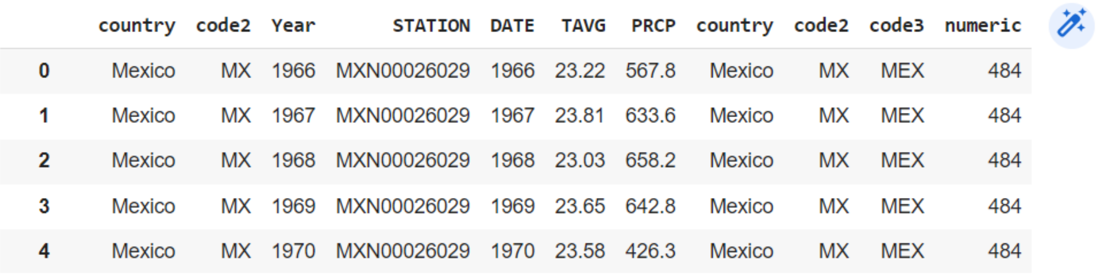
```
# Selecting a subset of columns
# Reference: https://www.ncei.noaa.gov/data/global-summary-of-the-year/doc/GSOY_documentation.pdf
# https://www.ncdc.noaa.gov/cdo-web/datasets
#TAVG - annual temperature average
#PRCP - total annual precipitation
raw_weather_all_features_df = raw_weather_df[["STATION", "DATE", "TAVG", "PRCP"]].dropna()
print(raw_weather_all_features_df)
```

```
             STATION  DATE   TAVG   PRCP
0        MXN00026029  1966  23.22  567.8
1        MXN00026029  1967  23.81  633.6
2        MXN00026029  1968  23.03  658.2
3        MXN00026029  1969  23.65  642.8
4        MXN00026029  1970  23.58  426.3
...              ...   ...    ...    ...
384623   BUM00015730  2016  13.56  580.5
384624   BUM00015730  2017  12.87  627.8
384625   BUM00015730  2018  13.64  778.0
```

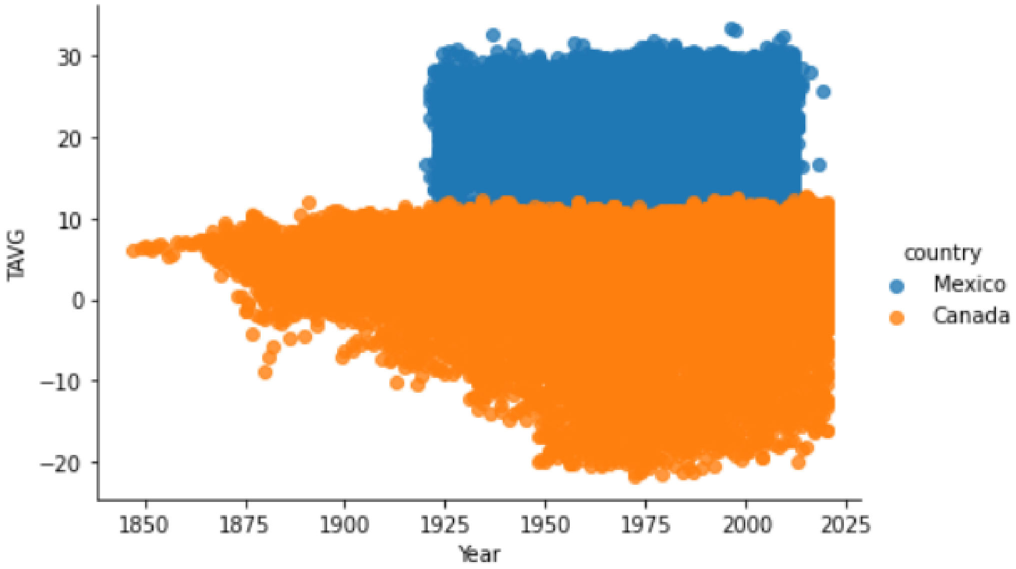
Next, we joined the list of country codes with the climate data. In order to better understand the data we had at hand, we proceeded to plot the Average Temperature for Mexico and Canada using the python visualization package ‘seaborn’. As expected, there were colder weather and more observations registered in Canada than in Mexico.



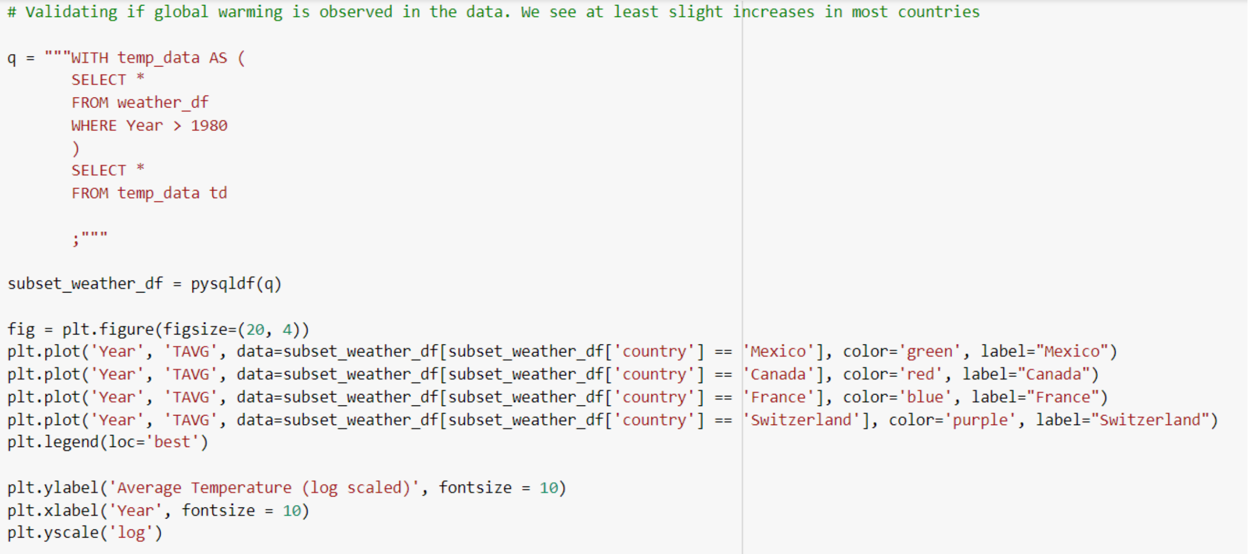
```python
# Formatting the weather data frame
from pandasql import sqldf

# Obtaining the country for each station
pysqldf = lambda q: sqldf(q, globals())

q = """WITH temp_data AS (
       SELECT SUBSTR(STATION, 1 , 2) AS code2, DATE AS Year, *
       FROM raw_weather_all_features_df
       )
       SELECT cd.country AS country, *
       FROM temp_data td
       JOIN country_codes_df cd ON td.code2 = cd.code2

       ;"""

raw_weather_by_country_df = pysqldf(q)
raw_weather_by_country_df
```



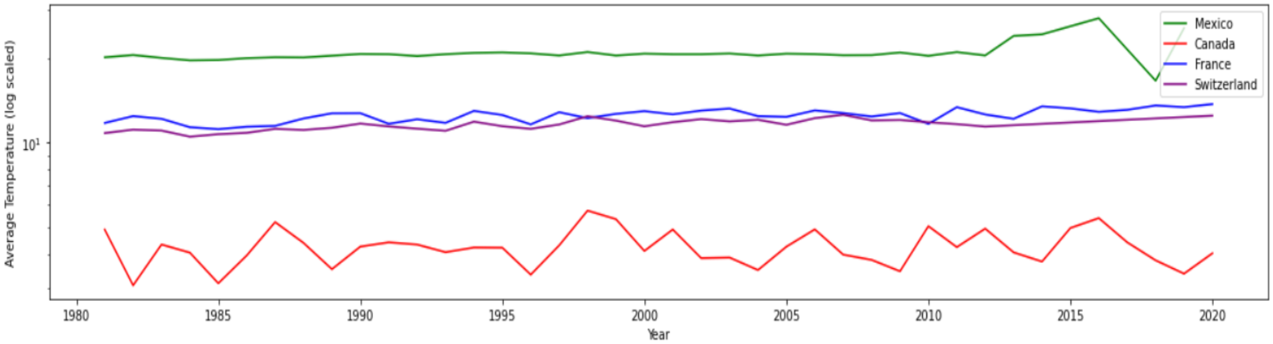
	country	code2	Year	STATION	DATE	TAVG	PRCP	country	code2	code3	numeric
0	Mexico	MX	1966	MXN00026029	1966	23.22	567.8	Mexico	MX	MEX	484
1	Mexico	MX	1967	MXN00026029	1967	23.81	633.6	Mexico	MX	MEX	484
2	Mexico	MX	1968	MXN00026029	1968	23.03	658.2	Mexico	MX	MEX	484
3	Mexico	MX	1969	MXN00026029	1969	23.65	642.8	Mexico	MX	MEX	484
4	Mexico	MX	1970	MXN00026029	1970	23.58	426.3	Mexico	MX	MEX	484



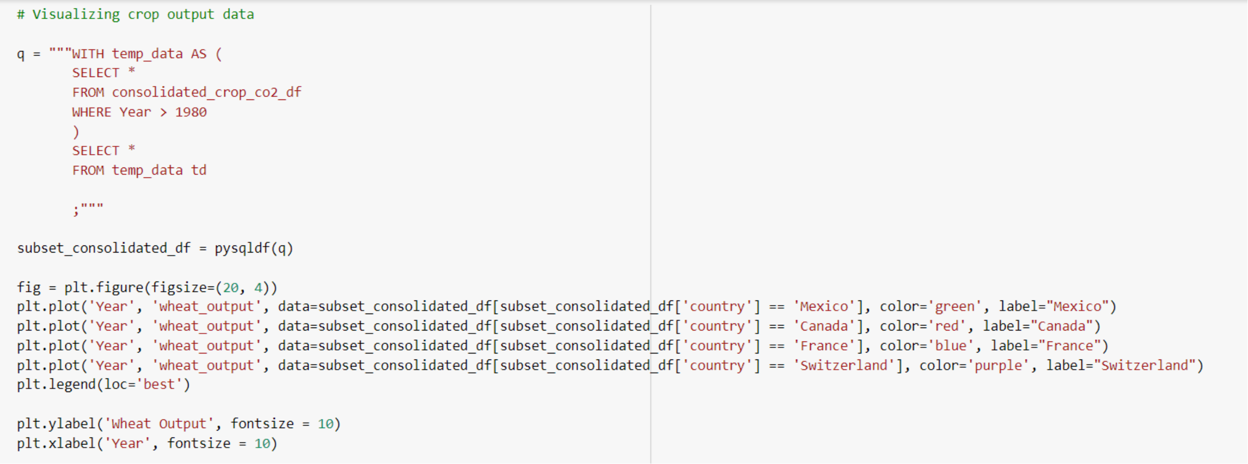
To ensure we were comparing the data points on a similar basis, we calculated the average annual temperature and precipitation by aggregating station data for the year and joining on country and year. Further, we evaluated whether the claim by the Natural Resources Defense Council about whether temperatures on average have risen each year was valid by plotting the average annual temperature for Canada, Mexico, France, and Switzerland since 1980.



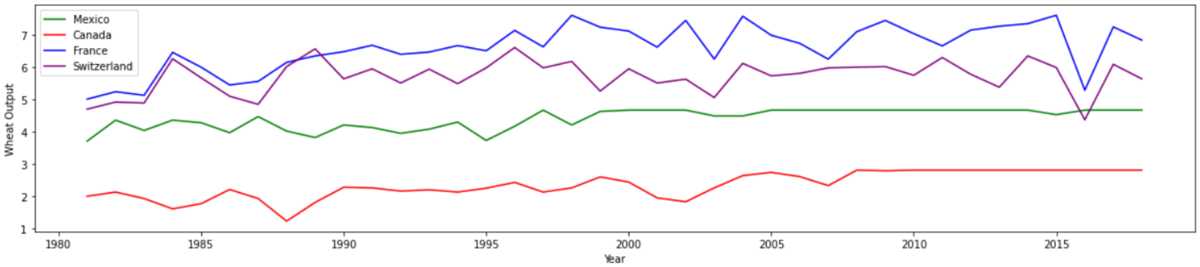
```
# Validating if global warming is observed in the data. We see at least slight increases in most countries

q = """WITH temp_data AS (
       SELECT *
       FROM weather_df
       WHERE Year > 1980
       )
       SELECT *
       FROM temp_data td

       ;"""

subset_weather_df = pysqldf(q)

fig = plt.figure(figsize=(20, 4))
plt.plot('Year', 'TAVG', data=subset_weather_df[subset_weather_df['country'] == 'Mexico'], color='green', label="Mexico")
plt.plot('Year', 'TAVG', data=subset_weather_df[subset_weather_df['country'] == 'Canada'], color='red', label="Canada")
plt.plot('Year', 'TAVG', data=subset_weather_df[subset_weather_df['country'] == 'France'], color='blue', label="France")
plt.plot('Year', 'TAVG', data=subset_weather_df[subset_weather_df['country'] == 'Switzerland'], color='purple', label="Switzerland")
plt.legend(loc='best')

plt.ylabel('Average Temperature (log scaled)', fontsize = 10)
plt.xlabel('Year', fontsize = 10)
plt.yscale('log')
```

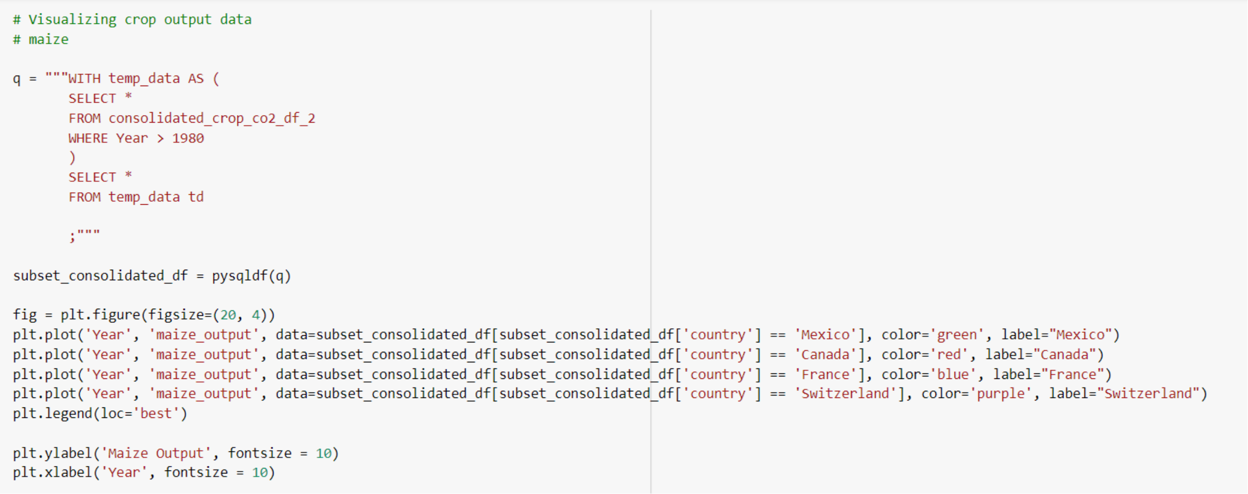


Based on our evaluation of the crops yield data, the two crops we decided to focus on for our analysis were Wheat and Maize, specifically focusing on the output for these two crops. We visualized the output for both wheat and maize in Mexico, Canada, France, and Switzerland since 1980 to understand trends over recent years. We can observe that there are fluctuations in the output. We now proceed to see if weather changes and CO2 emissions are key factors.

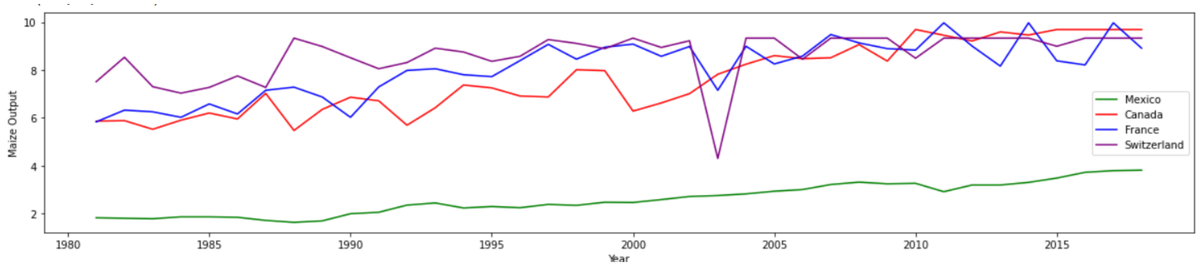


```
# Visualizing crop output data

q = """WITH temp_data AS (
       SELECT *
       FROM consolidated_crop_co2_df
       WHERE Year > 1980
       )
       SELECT *
       FROM temp_data td

       ;"""

subset_consolidated_df = pysqldf(q)

fig = plt.figure(figsize=(20, 4))
plt.plot('Year', 'wheat_output', data=subset_consolidated_df[subset_consolidated_df['country'] == 'Mexico'], color='green', label="Mexico")
plt.plot('Year', 'wheat_output', data=subset_consolidated_df[subset_consolidated_df['country'] == 'Canada'], color='red', label="Canada")
plt.plot('Year', 'wheat_output', data=subset_consolidated_df[subset_consolidated_df['country'] == 'France'], color='blue', label="France")
plt.plot('Year', 'wheat_output', data=subset_consolidated_df[subset_consolidated_df['country'] == 'Switzerland'], color='purple', label="Switzerland")
plt.legend(loc='best')

plt.ylabel('Wheat Output', fontsize = 10)
plt.xlabel('Year', fontsize = 10)
```





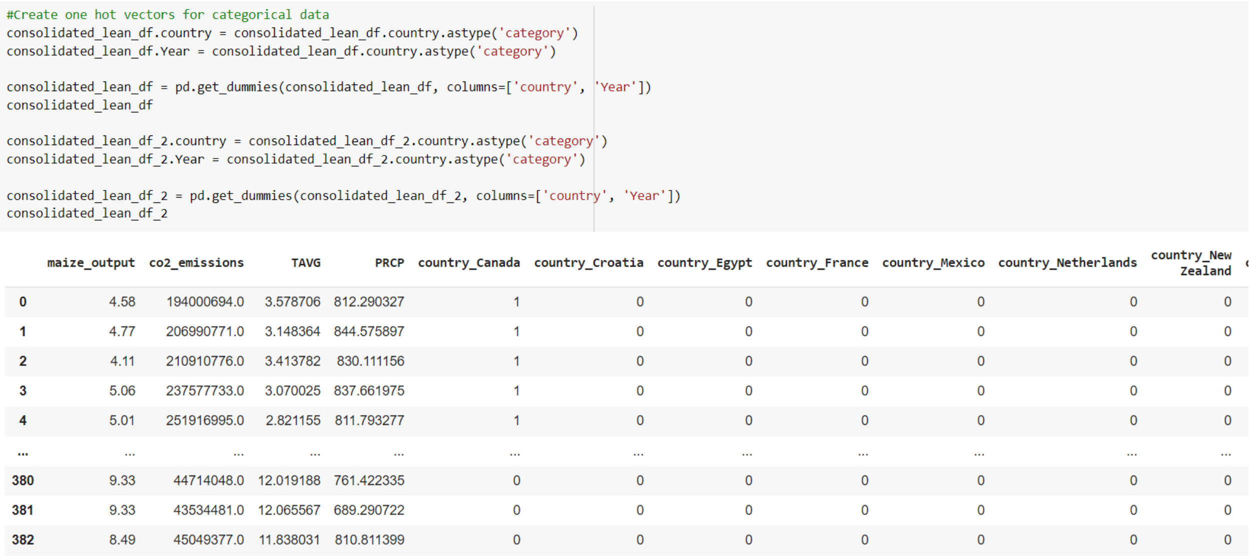
```
# Visualizing crop output data
# maize

q = """WITH temp_data AS (
       SELECT *
       FROM consolidated_crop_co2_df_2
       WHERE Year > 1980
       )
       SELECT *
       FROM temp_data td

       ;"""

subset_consolidated_df = pysqldf(q)

fig = plt.figure(figsize=(20, 4))
plt.plot('Year', 'maize_output', data=subset_consolidated_df[subset_consolidated_df['country'] == 'Mexico'], color='green', label="Mexico")
plt.plot('Year', 'maize_output', data=subset_consolidated_df[subset_consolidated_df['country'] == 'Canada'], color='red', label="Canada")
plt.plot('Year', 'maize_output', data=subset_consolidated_df[subset_consolidated_df['country'] == 'France'], color='blue', label="France")
plt.plot('Year', 'maize_output', data=subset_consolidated_df[subset_consolidated_df['country'] == 'Switzerland'], color='purple', label="Switzerland")
plt.legend(loc='best')

plt.ylabel('Maize Output', fontsize = 10)
plt.xlabel('Year', fontsize = 10)
```



Methodology

As part of the Exploratory Data Analysis (EDA), we processed and analyzed all of the datasets that we planned to use in our model. The climate, crop yield and CO2 emissions data all had approximately 60 years of data for approximately 100 different stations in numerous countries and crops, which posed several challenges for us as we were reading in the data.

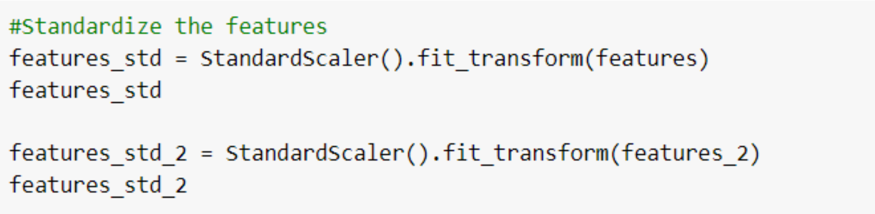
In our final dataframe there are two categorical columns, which represent variables that contain labels rather than numeric values, specifically Country and Year. We used one-hot encoding to convert the categorical data to numerical form. The one-hot encoding created binary columns for each category and the matrix that was returned is shown below.



```
#Create one hot vectors for categorical data
consolidated_lean_df.country = consolidated_lean_df.country.astype('category')
consolidated_lean_df.Year = consolidated_lean_df.country.astype('category')

consolidated_lean_df = pd.get_dummies(consolidated_lean_df, columns=['country', 'Year'])
consolidated_lean_df

consolidated_lean_df_2.country = consolidated_lean_df_2.country.astype('category')
consolidated_lean_df_2.Year = consolidated_lean_df_2.country.astype('category')

consolidated_lean_df_2 = pd.get_dummies(consolidated_lean_df_2, columns=['country', 'Year'])
consolidated_lean_df_2
```

	maize_output	co2_emissions	TAVG	PRCP	country_Canada	country_Croatia	country_Egypt	country_France	country_Mexico	country_Netherlands	country_New Zealand
0	4.58	194000694.0	3.578706	812.290327	1	0	0	0	0	0	0
1	4.77	206990771.0	3.148364	844.575897	1	0	0	0	0	0	0
2	4.11	210910776.0	3.413782	830.111156	1	0	0	0	0	0	0
3	5.06	237577733.0	3.070025	837.661975	1	0	0	0	0	0	0
4	5.01	251916995.0	2.821155	811.793277	1	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...
380	9.33	44714048.0	12.019188	761.422335	0	0	0	0	0	0	0
381	9.33	43534481.0	12.065567	689.290722	0	0	0	0	0	0	0
382	8.49	45049377.0	11.838031	810.811399	0	0	0	0	0	0	0

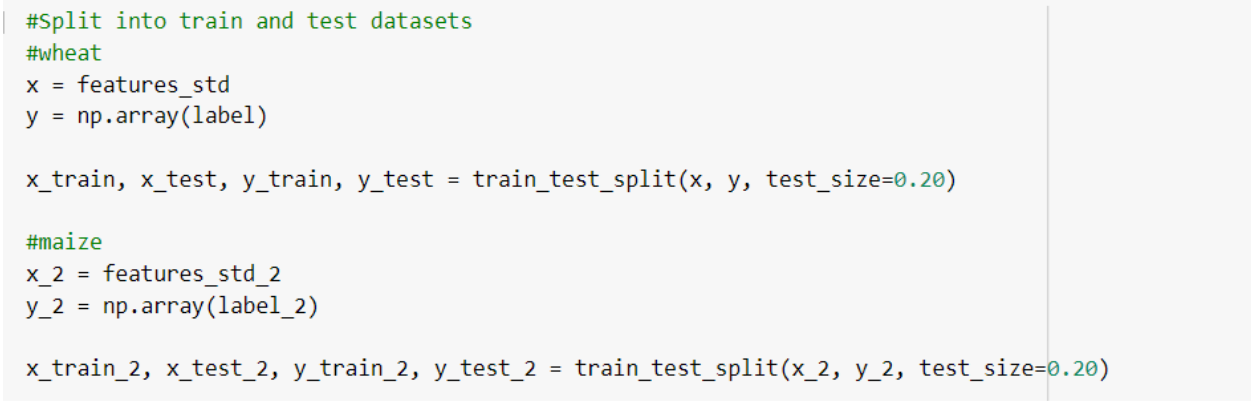
In order to bring all features with high magnitudes to the same level of magnitudes, we scaled the features using StandardScaler. The StandardScaler standardizes features by removing the mean and scaling to unit variance.



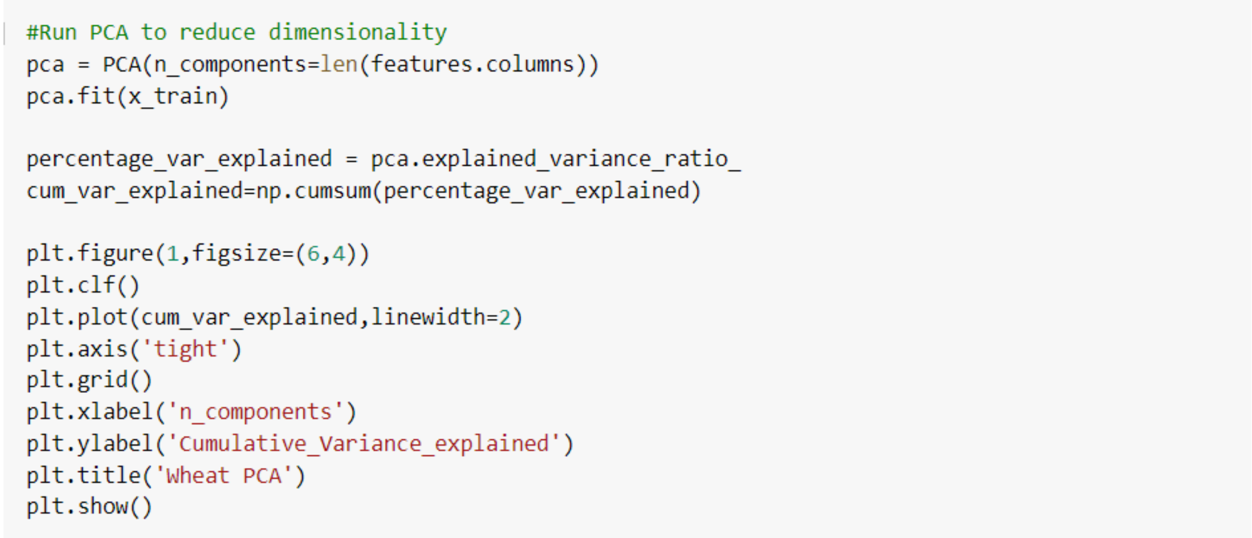
```
#Standardize the features
features_std = StandardScaler().fit_transform(features)
features_std

features_std_2 = StandardScaler().fit_transform(features_2)
features_std_2
```

Next the dataset was split into two datasets, a training dataset and a testing dataset. Given training the model will require more data points, we decided to split the dataset between training and testing with an 80/20 split with 80% of the dataset used to train the model and

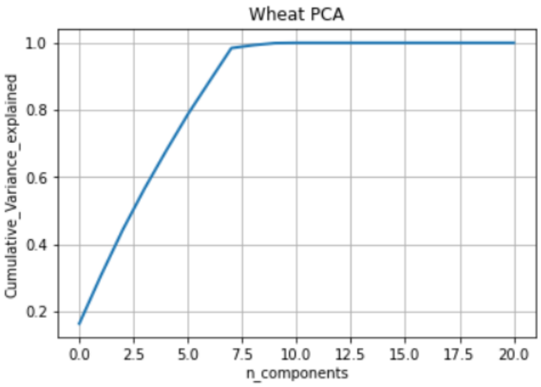
20% of the dataset used to test the model. Then, we ran Principal Component Analysis (PCA) in order to reduce the dimensionality of the data. Through this process, we determined that 7 components are sufficient for our analysis of the wheat and maize crops.

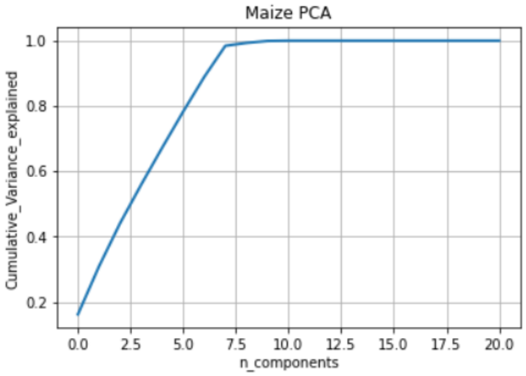


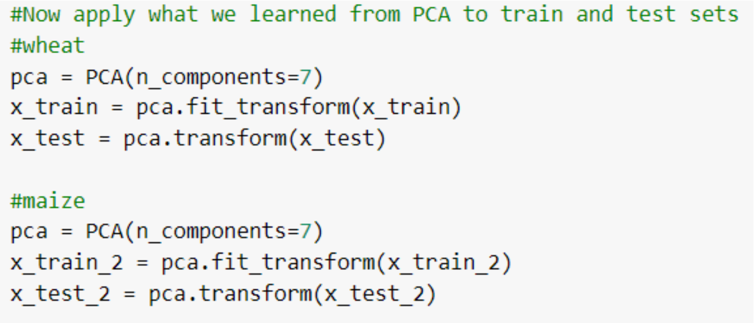
```
#Split into train and test datasets
#wheat
x = features_std
y = np.array(label)

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.20)

#maize
x_2 = features_std_2
y_2 = np.array(label_2)

x_train_2, x_test_2, y_train_2, y_test_2 = train_test_split(x_2, y_2, test_size=0.20)
```



```
#Run PCA to reduce dimensionality
pca = PCA(n_components=len(features.columns))
pca.fit(x_train)

percentage_var_explained = pca.explained_variance_ratio_
cum_var_explained=np.cumsum(percentage_var_explained)

plt.figure(1,figsize=(6,4))
plt.clf()
plt.plot(cum_var_explained,linewidth=2)
plt.axis('tight')
plt.grid()
plt.xlabel('n_components')
plt.ylabel('Cumulative_Variance_explained')
plt.title('Wheat PCA')
plt.show()
```



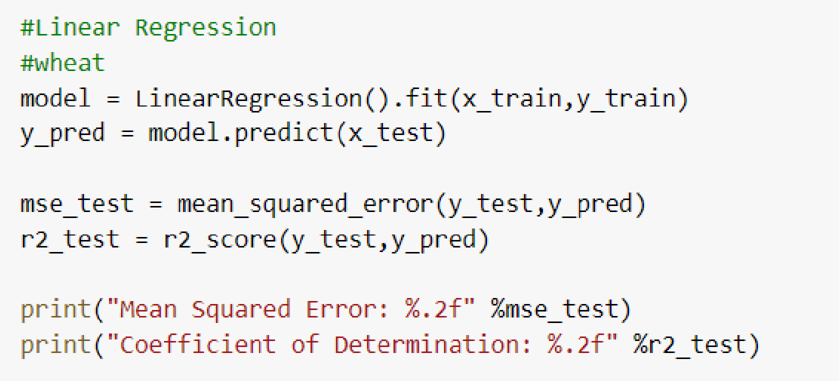




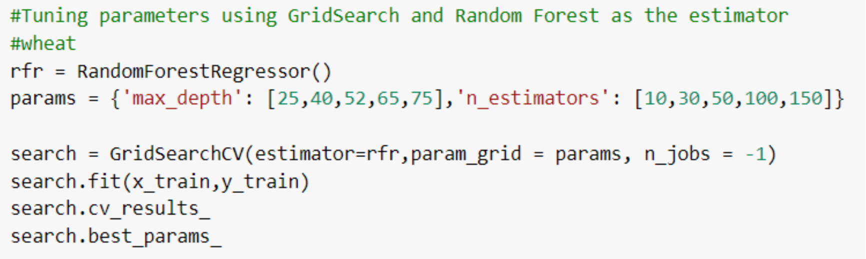
```
#Now apply what we learned from PCA to train and test sets
#wheat
pca = PCA(n_components=7)
x_train = pca.fit_transform(x_train)
x_test = pca.transform(x_test)

#maize
pca = PCA(n_components=7)
x_train_2 = pca.fit_transform(x_train_2)
x_test_2 = pca.transform(x_test_2)
```

Results

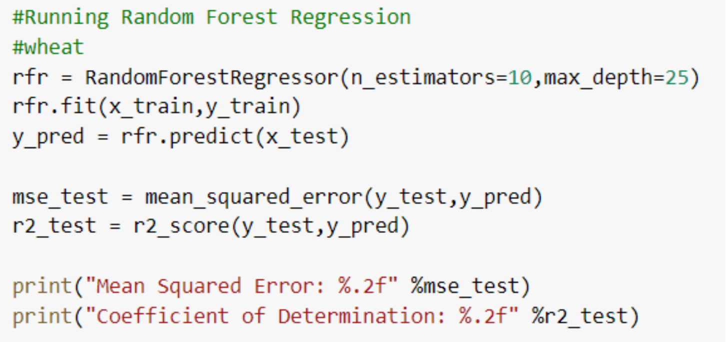
We decided to use a Linear Regression Model, GridSearch, and RandomForestRegressor to evaluate our dataset. Initially, we started with a Linear Regression Model to determine the mean squared error and coefficient of determination. Then we decided to tune the parameters using GridSearch. GridSearch is used to find the optimal hyperparameters of a model that ultimately leads to more accurate predictions. RandomForest Regression is a supervised machine learning algorithm that uses bagging and acts as an estimator to output the most optimal result.



```
#Linear Regression
#wheat
model = LinearRegression().fit(x_train,y_train)
y_pred = model.predict(x_test)

mse_test = mean_squared_error(y_test,y_pred)
r2_test = r2_score(y_test,y_pred)

print("Mean Squared Error: %.2f" %mse_test)
print("Coefficient of Determination: %.2f" %r2_test)
```



```
#Tuning parameters using GridSearch and Random Forest as the estimator
#wheat
rfr = RandomForestRegressor()
params = {'max_depth': [25,40,52,65,75],'n_estimators': [10,30,50,100,150]}

search = GridSearchCV(estimator=rfr,param_grid = params, n_jobs = -1)
search.fit(x_train,y_train)
search.cv_results_
search.best_params_
```



```
#Running Random Forest Regression
#wheat
rfr = RandomForestRegressor(n_estimators=10,max_depth=25)
rfr.fit(x_train,y_train)
y_pred = rfr.predict(x_test)

mse_test = mean_squared_error(y_test,y_pred)
r2_test = r2_score(y_test,y_pred)

print("Mean Squared Error: %.2f" %mse_test)
print("Coefficient of Determination: %.2f" %r2_test)
```

We completed the process above for both of our crops and the MSE and R^2 for both wheat and maize were more accurate for the GridSearch and RandomForest Regressor compared to the Linear Regression Model as we would have expected. Our final iteration of the models resulted in a MSE of 1.32 and R^2 of 0.72 for Wheat when running the Linear Regression Model. The MSE for Wheat is reduced to 0.50 and the R^2 value increased to 0.90 when we used the GridSearch and RandomForest Regressor. Further, our final iteration of the models resulted in a MSE of 2.75 and R^2 of 0.69 for Maize when running the Linear Regression Model. The MSE for Maize was reduced to 1.23 and the R^2 value increased to 0.86 when we used the GridSearch and RandomForest Regressor. Generally, a higher R^2 indicates a better fit for the model. From the obtained results, our Random Forest Regressor models are an above average representation of the data.

Final Thoughts

Our original hypothesis that the crop yield productions are impacted by climate changes and CO2 emissions was validated by performing this analysis and creating our models. Although we recognize that crop production is impacted by additional factors such as population growth, political and economic policies, etc., we can say that climate does play a vital role. Furthermore, the data we aggregated took averages for weather features that were spread across a respective country. By doing so, we made assumptions about the spread of climate being somewhat evenly distributed. This project in future iterations can break down countries into regions and sectors, allowing for more robust predictions. Additionally, population changes, economic crop production minimum/maximums, C02 emissions restrictions could also be factored.

Lastly, we would like to thank our professors and TAs for CIS 545 at University of Pennsylvania for a wonderful introduction class to big data analytics. It’s their support and push that allowed us to find an interesting and complex problem of our world and apply data to find answers.