CSE 587
Spring 2024

DIC Project Phase 3

Group Members:

Andrew Balotin

Deep Shahane

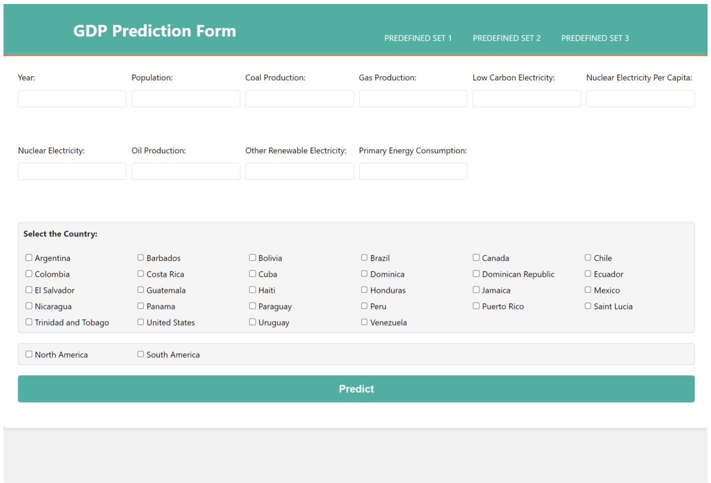
Bhushan Mahajan

Table of Contents

User Instructions:

The following are detailed instructions on how to use the user interface for the GDP prediction application.

When first booting up the GUI, the first screen that is shown is given in the Figure below:



Year:

Population:

Coal Production:

Gas Production:

Low Carbon Electricity:

Nuclear Electricity Per Capita:

Nuclear Electricity:

Oil Production:

Other Renewable Electricity:

Primary Energy Consumption:

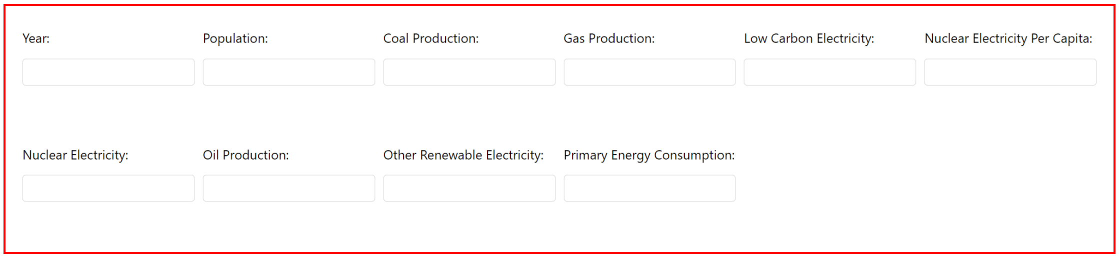
Select the Country:

☐ Argentina ☐ Barbados ☐ Bolivia ☐ Brazil ☐ Canada ☐ Chile
☐ Colombia ☐ Costa Rica ☐ Cuba ☐ Dominica ☐ Dominican Republic ☐ Ecuador
☐ El Salvador ☐ Guatemala ☐ Haiti ☐ Honduras ☐ Jamaica ☐ Mexico
☐ Nicaragua ☐ Panama ☐ Paraguay ☐ Peru ☐ Puerto Rico ☐ Saint Lucia
☐ Trinidad and Tobago ☐ United States ☐ Uruguay ☐ Venezuela

☐ North America ☐ South America

Figure 1: Main Screen

This screen is where all the user inputs and the output prediction will be displayed. The screen is split up into a couple of different sections. The main part of the user interface is the input prompts which allow a user to populate the text boxes. These prompt boxes are essential for calculating an accurate GDP prediction. These variables include the year, population, coal production, gas production, low carbon electricity, nuclear energy per capita, nuclear electricity, oil production, other renewable electricity, and primary energy consumption. These boxes are shown in the figure below:



Year:

Population:

Coal Production:

Gas Production:

Low Carbon Electricity:

Nuclear Electricity Per Capita:

Nuclear Electricity:

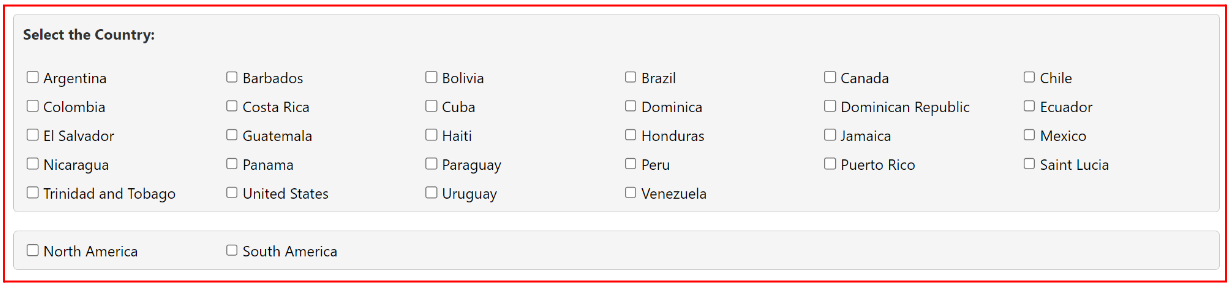
Oil Production:

Other Renewable Electricity:

Primary Energy Consumption:

Figure 2: Prompt Boxes

The user has the ability to predict the GDP for any country that the model was trained on. Once a country is selected, the corresponding continent must also be chosen for proper calculation. This selection is shown in the Figure below:



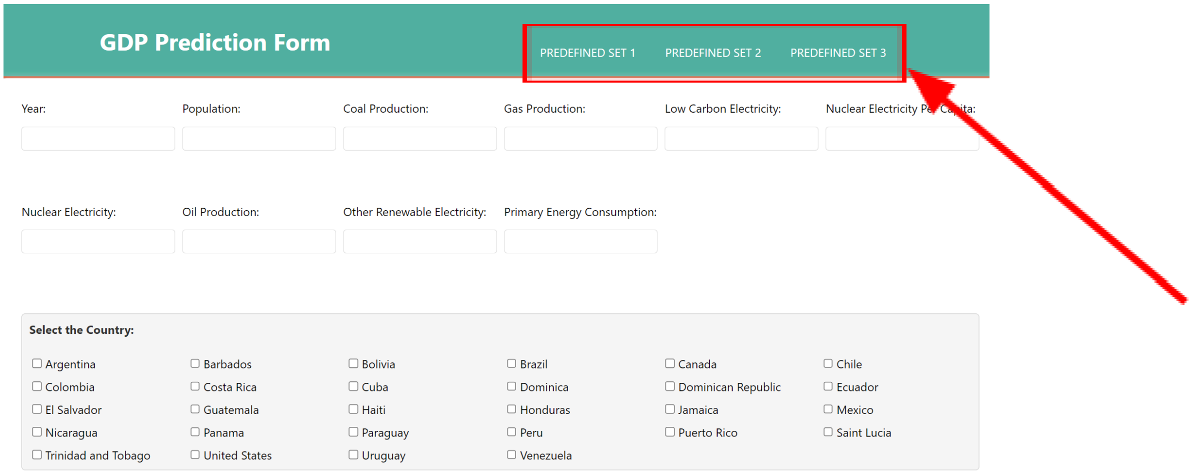
Select the Country:

□ Argentina □ Barbados □ Bolivia □ Brazil □ Canada □ Chile
□ Colombia □ Costa Rica □ Cuba □ Dominica □ Dominican Republic □ Ecuador
□ El Salvador □ Guatemala □ Haiti □ Honduras □ Jamaica □ Mexico
□ Nicaragua □ Panama □ Paraguay □ Peru □ Puerto Rico □ Saint Lucia
□ Trinidad and Tobago □ United States □ Uruguay □ Venezuela

□ North America □ South America

Figure 3: Country and Continent Selection

The user also has the ability to test the functionality of the application with predefined values. These predefined sets are located in the top right of the application and are shown in the Figure below:



GDP Prediction Form

Year: Population: Coal Production: Gas Production: Low Carbon Electricity: Nuclear Electricity Per Capita:

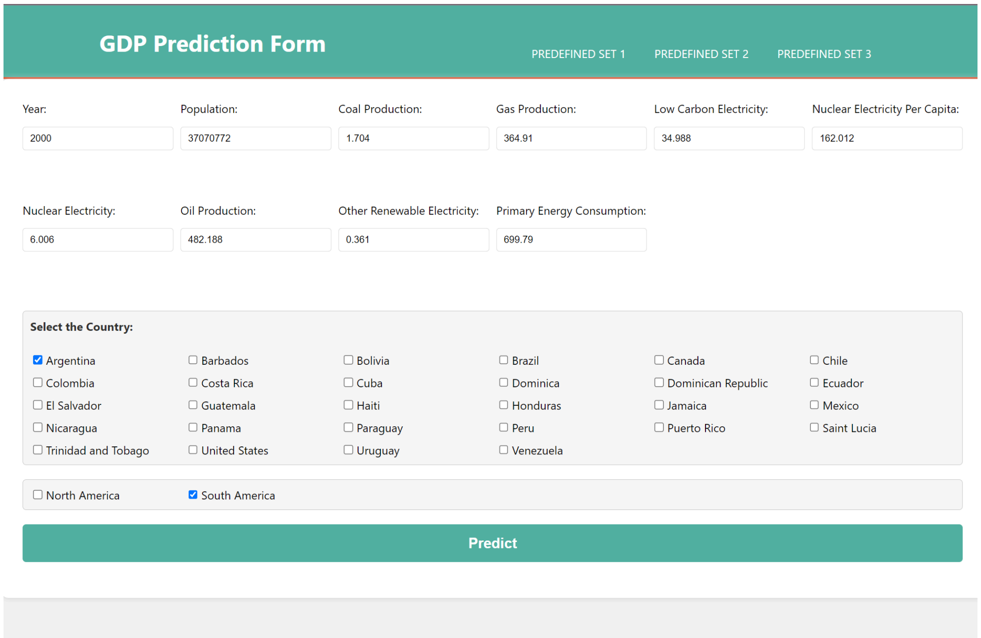
Nuclear Electricity: Oil Production: Other Renewable Electricity: Primary Energy Consumption:

Select the Country:

□ Argentina □ Barbados □ Bolivia □ Brazil □ Canada □ Chile
□ Colombia □ Costa Rica □ Cuba □ Dominica □ Dominican Republic □ Ecuador
□ El Salvador □ Guatemala □ Haiti □ Honduras □ Jamaica □ Mexico
□ Nicaragua □ Panama □ Paraguay □ Peru □ Puerto Rico □ Saint Lucia
□ Trinidad and Tobago □ United States □ Uruguay □ Venezuela

Figure 4: Option Selection

Each of these predefined sets are selectable. Once, one of these predefined sets are selected, values are populated inside the text boxes along with the country and continent selection. This process is shown for “PREDEFINED SET 1” in the Figure below:



GDP Prediction Form

Year:	Population:	Coal Production:	Gas Production:	Low Carbon Electricity:	Nuclear Electricity Per Capita:
2000	37070772	1.704	364.91	34.988	162.012
Nuclear Electricity:	Oil Production:	Other Renewable Electricity:	Primary Energy Consumption:		
6.006	482.188	0.361	699.79		

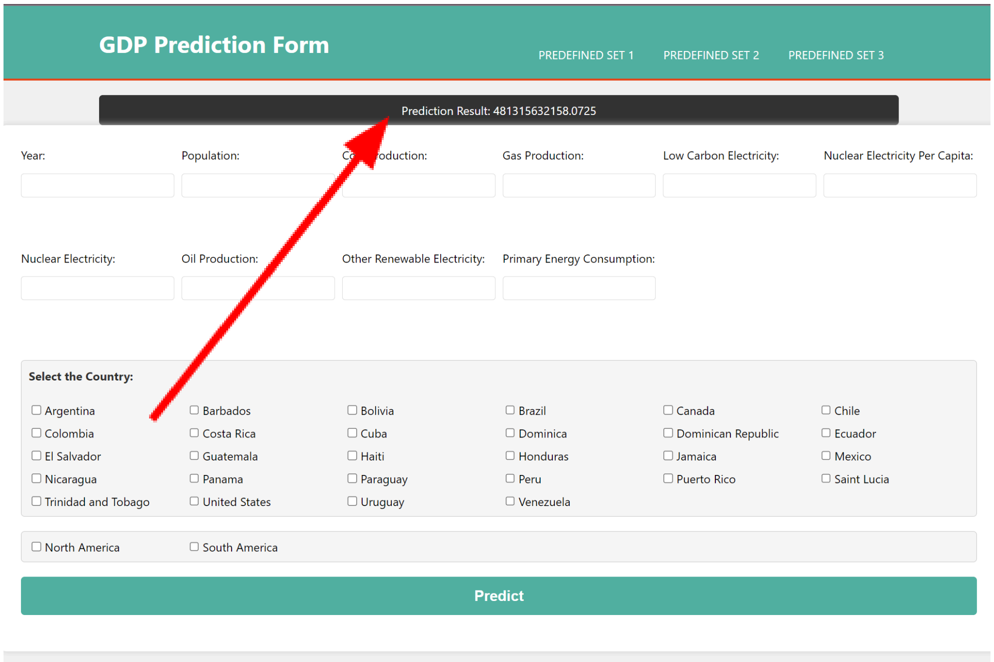
Select the Country:

☑ Argentina	☐ Barbados	☐ Bolivia	☐ Brazil	☐ Canada	☐ Chile
☐ Colombia	☐ Costa Rica	☐ Cuba	☐ Dominica	☐ Dominican Republic	☐ Ecuador
☐ El Salvador	☐ Guatemala	☐ Haiti	☐ Honduras	☐ Jamaica	☐ Mexico
☐ Nicaragua	☐ Panama	☐ Paraguay	☐ Peru	☐ Puerto Rico	☐ Saint Lucia
☐ Trinidad and Tobago	☐ United States	☐ Uruguay	☐ Venezuela		

☐ North America ☑ South America

Figure 5: Predefined Set Autofill

Once the values have been autofilled, the predict button can then be pressed to output a GDP prediction. This output is shown in a black box at the top of the screen. The values are then auto cleared for another GDP prediction. This is shown in the Figure below:



GDP Prediction Form

Prediction Result: 481315632158.0725

Year:	Population:	Coal Production:	Gas Production:	Low Carbon Electricity:	Nuclear Electricity Per Capita:
Nuclear Electricity:	Oil Production:	Other Renewable Electricity:	Primary Energy Consumption:		

Select the Country:

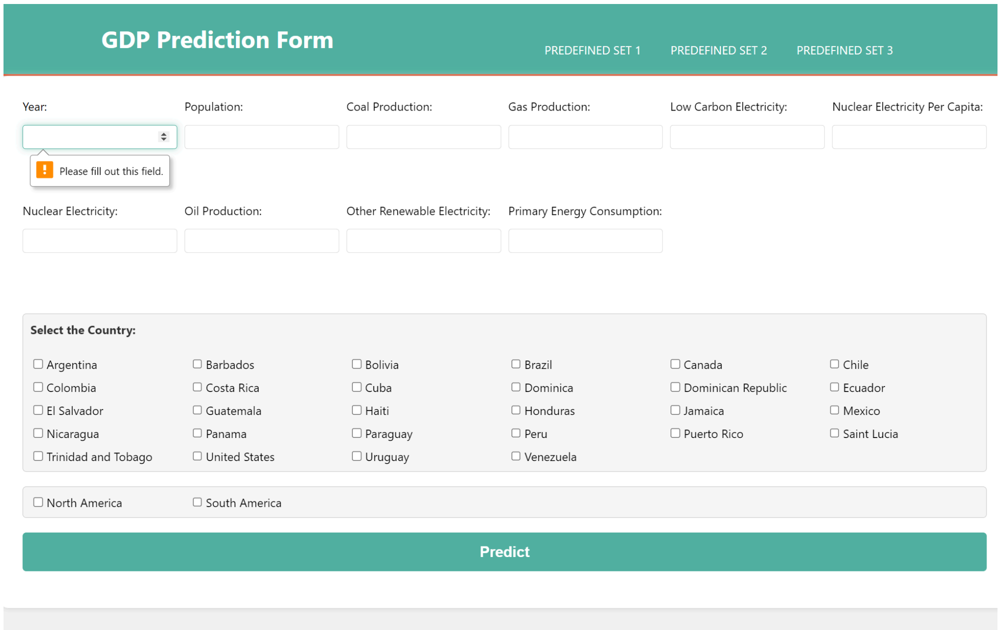
☐ Argentina	☐ Barbados	☐ Bolivia	☐ Brazil	☐ Canada	☐ Chile
☐ Colombia	☐ Costa Rica	☐ Cuba	☐ Dominica	☐ Dominican Republic	☐ Ecuador
☐ El Salvador	☐ Guatemala	☐ Haiti	☐ Honduras	☐ Jamaica	☐ Mexico
☐ Nicaragua	☐ Panama	☐ Paraguay	☐ Peru	☐ Puerto Rico	☐ Saint Lucia
☐ Trinidad and Tobago	☐ United States	☐ Uruguay	☐ Venezuela		

☐ North America ☐ South America

:

Figure 6: GDP Prediction

The user interface also includes error handling. If no values are inputted into the boxes and the predict button is clicked, a warning message will appear in the box that prevents the model from being calculated. This is shown in the Figure below:



GDP Prediction Form

Year: Population: Coal Production: Gas Production: Low Carbon Electricity: Nuclear Electricity Per Capita:

Please fill out this field.

Nuclear Electricity: Oil Production: Other Renewable Electricity: Primary Energy Consumption:

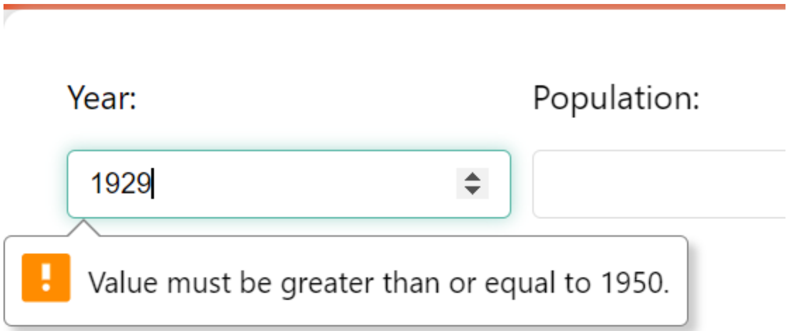
Select the Country:

□ Argentina □ Barbados □ Bolivia □ Brazil □ Canada □ Chile
□ Colombia □ Costa Rica □ Cuba □ Dominica □ Dominican Republic □ Ecuador
□ El Salvador □ Guatemala □ Haiti □ Honduras □ Jamaica □ Mexico
□ Nicaragua □ Panama □ Paraguay □ Peru □ Puerto Rico □ Saint Lucia
□ Trinidad and Tobago □ United States □ Uruguay □ Venezuela

□ North America □ South America

Figure 7: Null Warning Message

Furthermore, since the model was trained on data from 1950 - Present, the prompt will not allow the user to input a year that is before 1950. This error is shown in the Figure below:



Year: 1929

Population:

Value must be greater than or equal to 1950.

Figure 8: Year Error Message

Models Used and Tuning Parameters:

The models used for GDP prediction were selected by using a couple different metrics. First, we went through the results of phase 2 and analyzed the mse, mae, accuracy, feature importance, and where applicable, confusion matrices. We wanted a model that was high accuracy and could predict the GDP with a high degree of confidence. The models that had lower than a 95% accuracy were removed for consideration. This left us with two models: linear regression and xgboost. The results for both of models are given below:

Linear Regression
Score: 0.99381
MAE: 0.03237
MSE: 0.00597

Xgboost
Score: 0.99814
MAE: 0.01451
MSE: 0.00209

Since we used a large majority of our columns for model training, we knew that we needed to reduce the variables for the final application. Reducing the variables decreased the accuracy depending on what variable was removed. In order to combat this, we looked at the feature importance for each model we were considering. For example, for our xgboost model, the feature importance is given in the Figure below:

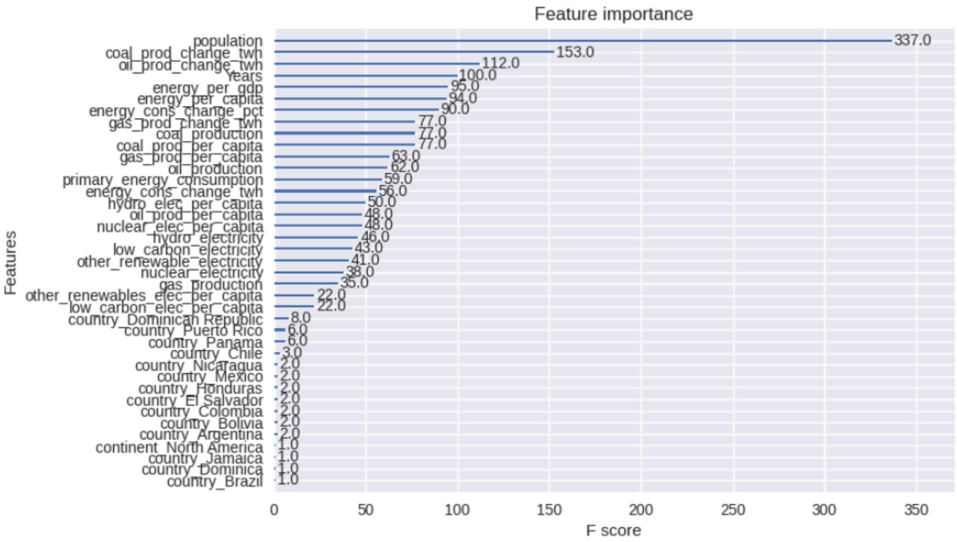


Figure 9: Xgboost Feature Importance

The higher the number on the x-axis, the higher correlation it has with calculating the final output. Furthermore, we also looked at the heatmap for the entire dataset to see what correlated the most with GDP. This is shown in the Figure below:

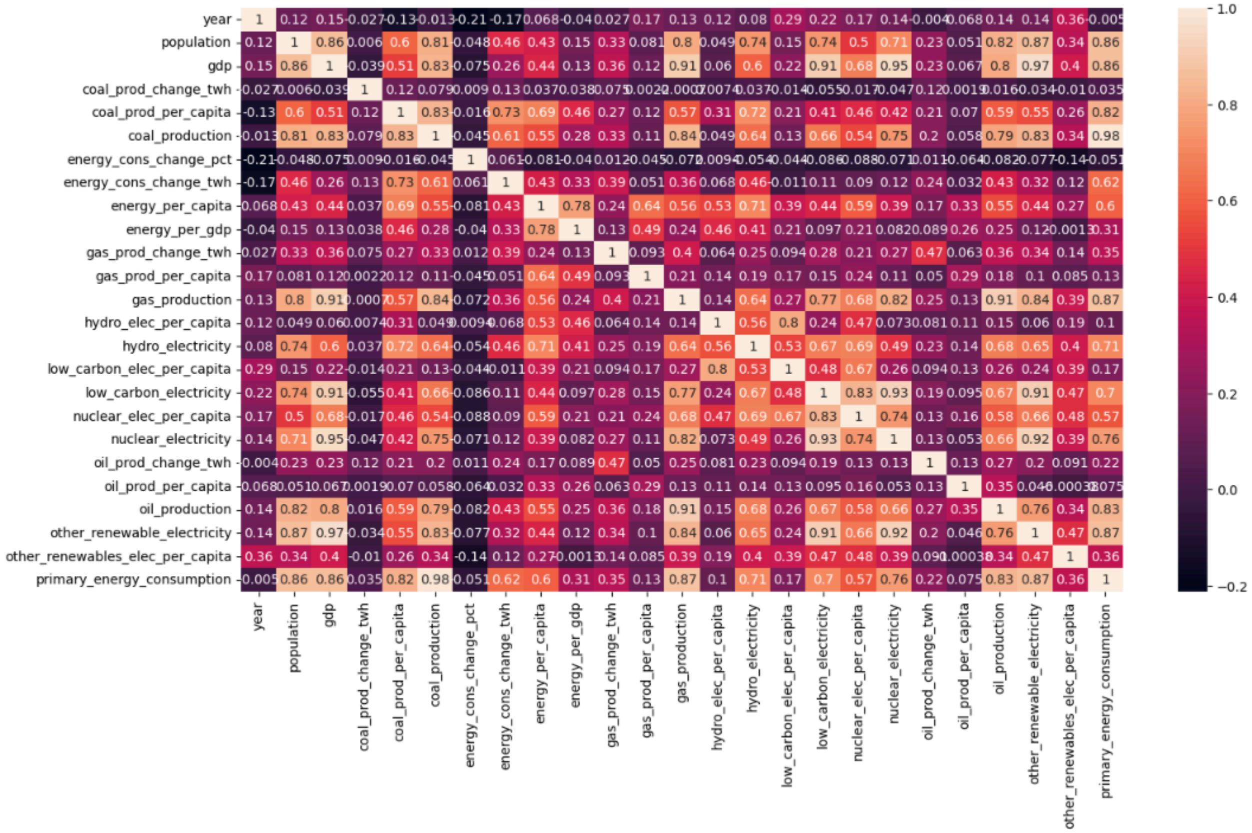


Figure 10: Data Heatmap

We determined that anything below a 0.6 correlation would be removed from the training variables. On top of this consideration, continents with a large number of countries were removed for better training. This was to ensure our models would not have a large amount of parameters. With all of the above in consideration, the training variables were reduced to 10. The continents were reduced to only North and South America. The models were then re-evaluated and the results are given below:

Linear Regression
Score: 0.98755
MAE: 0.03777
MSE: 0.01202

Xgboost
Score: 0.99814

MAE: 0.02438
MSE: 0.00179

Because of the higher accuracy and lower MAE and MSE, xgboost was chosen to predict the GDP.

Recommendations:

The problem statement that was originally created for this project will now be revisited. The following statement was taken from our phase 1 document: “The goal is to enhance economic forecasting by creating a robust model that is capable of simulating the impact of energy consumption changes on GDP”. The application that was built is a mirror of what our original problem statement contained. It is a robust model that is able to predict a country’s GDP based on their energy consumption and production. Users are able to learn how the changing economic factors can influence a country’s annual growth or decline. These factors include changes to the population, energy production, and energy consumption. For example, it can be used to simulate the growth of nuclear energy in a South American country. If a country produces higher nuclear energy output, the application can be used to analyze how that will affect the country’s GDP. Another example is production of oil. If a country finds an oil reserve, it can show how much that increase in oil production will change the GDP. On the contrary, if an oil reserve is depleted, how that will decrease economic growth. Finally, another example could be how rising population and energy consumption without the increase of renewable energy could decrease a country’s GDP. The previous examples are just a few situations for a user that might benefit from this model and there are many more applications that this would be useful for.

One way that this project could be expanded is to extend the model to different continents. With more training power, this model could be used to predict the GDP for every country in the world. In order to do this, each country would have to be added to the final dataset for training. The hyperparameters, such as the feature coefficients, might have to be adjusted once the new countries are in the model. Potentially, different feature coefficients might have to be used for the added continents. In addition to extending this model to different countries, exploration could be done to add import/export statistics to the final model. This feature could show how external interactions could influence a country's GDP. If a country is not producing a lot of oil internally, but is importing a lot of oil, that could be a major factor in economic stability and growth. In conclusion, this model can predict a country’s GDP based on factors like internal energy consumption and production. With this model, it can show how changes to these variables will affect economic growth or decline. If this model were to be extended, it has the potential to predict the GDP for every country in the world, giving an accurate prediction to global growth.

Code Documentation and New Dataset Instructions:

Phase1:

In this phase of development, the dataset was cleaned and prepped for exploratory data analysis. The code is mostly adaptable, meaning that as long as the new dataset inputted has a gdp, country, and population column, the code will be able to clean the data. This can be done by mounting the dataset folder in a drive folder and running the third cell from the top. This will input the dataset and then each cell can be individually run to clean the data. If the user does not have the libraries required, they can run pip install using the same format as the cell above it. Then, the remaining half of the code can be run which contains the EDA. The contents of the new dataset can be visualized and then exported to use in the code for phase 2.

Phase 2:

In this phase, we've delved into training and evaluating multiple models. Initially, we imported and meticulously cleaned our dataset from phase 1. Given its substantial size - boasting 21 columns and 2170 rows - users need to specify the path or mount to the drive for storage. To seamlessly run the code, users must have several key libraries imported: pandas, numpy, matplotlib, seaborn, and sklearn. Once set up, users can explore the code for generating a heatmap, a visual representation showcasing the most crucial features for model training. In our analysis, we handpicked the top 11 features alongside country and continent details. Subsequently, we tackled one-hot encoding where necessary and saved a pkl file containing the predicted column. Then, we transformed the data to scalar format, saving the resulting file. The dataset was then divided into 80% for training and 20% for testing, though users have the flexibility to adjust this split to suit their needs. For Linear Regression, the process is straightforward: fitting the model, predicting on the test dataset, and calculating errors. The regression line is automatically plotted for visual clarity. Moving on to XGBoost, users need to ensure they've installed it before proceeding. We fine-tuned the model using specific hyperparameters such as max_depth = 5 and reg_lambda = 0.4. Executing the subsequent cells yields model scores, mean absolute error, mean squared error, and root mean squared error, accompanied by a graph plotting. For Logistic Regression, users can define features in X and the prediction target in y. The code also facilitates the plotting of feature coefficients specified in X. Similarly, users can explore other models such as Random Forest, KNN, Gaussian Bayes, and SVM. These require installation from sklearn, and the code prints the accuracy of each respective model. Lastly, we present a confusion matrix for all models, requiring users to import confusion_matrix from sklearn. This comprehensive analysis offers users insights into the performance and efficacy of various machine learning algorithms on our dataset.

Phase 3:

In this phase of development, we've crafted a user-friendly Graphical User Interface (GUI) to enable effortless interaction with our model. We've built this interface using Python, leveraging the Flask framework. To get started, users need to have Python installed on their systems, alongside Flask, request, and render_template libraries. Our model's architecture involves storing the model's weights and the scalar used for input transformation in a serialized format (pkl). This setup ensures smooth communication between user inputs and model predictions. Therefore, the system necessitates the inclusion of the 'pickle' module to handle serialization and deserialization tasks. We've opted for XGBoost as our primary model due to its exceptional performance. Consequently, users must have XGBoost installed to ensure the seamless operation of the system. Launching the system is straightforward. Users only need to execute the 'app.py' file. This action initiates local hosting of the application, generating a unique link. By pasting this link into their web browsers, such as Chrome, users gain access to the GUI, simplifying interaction with the system. In summary, the integration of a GUI not only enhances user experience but also streamlines the utilization of our model, making input and output interactions more intuitive and accessible.

- Please write the names of your group members.

Group member 1 : Bhushan Mahajan

Group member 2 : Deep Nitin Shahane

Group member 3 : Andrew Balotin

- Rate each groupmate on a scale of 5 on the following points, with 5 being HIGHEST and 1 being LOWEST.

Evaluation Criteria	**Group member 1**	**Group member 2**	**Group member 3**
How effectively did your group mate work with you?	5	5	5
Contribution in writing the report	5	5	5
Demonstrates a cooperative and supportive attitude.	5	5	5
Contributes significantly to the success of the project .	5	5	5
TOTAL	20	20	20

Also please state the overall contribution of your teammate in percentage below, with total of all the three members accounting for 100% (33.33+33.33+33.33 ~ 100%) :

Group member 1 : 33.33

Group member 2 : 33.33

Group member 3 : 33.33