CO_2 Emissions Prediction Using Machine Learning

Detailed Report of the project
Yashaswika Thota
Spirit ID : 1036572

1. Introduction

The escalating global concern over climate change has underscored the critical importance of analyzing and predicting CO_2 emissions. This report details a project focused on leveraging machine learning techniques to forecast CO_2 emissions, utilizing the "CO_2 Emissions by Country" dataset available on Kaggle. The primary objective of this project is to apply robust data preprocessing methodologies and construct predictive models capable of accurately forecasting future emission trends. This analysis aims to provide valuable insights into emission patterns, thereby aiding in the formulation of effective mitigation strategies and policy decisions.

2. Data Description

The "CO_2 Emissions by Country" dataset provides a comprehensive record of CO_2 emissions across various countries and years. The dataset's origin is the Climate Watch Data portal, with data sourced from the CAIT data source. This dataset is structured to facilitate the analysis of emission trends over time, offering a range of variables that capture different facets of emissions data. Key variables within the dataset include:

- **Country:** Categorical variable indicating the country for which emissions data is recorded.
- **Year:** Temporal variable specifying the year of the emission record.
- **Emissions:** Numerical variable representing the quantity of CO_2 emissions.

Other variables such as 'Data source', 'Sector', 'Gas', and 'Unit' are also present, providing additional context to the emissions data. The dataset spans a significant period, covering emissions data from 1990 onwards, making it suitable for time series analysis and trend forecasting.

3. Data Cleaning and Transformation

The raw dataset underwent several critical cleaning and transformation steps to prepare it for machine learning modeling. These steps were essential to ensure data quality, consistency, and suitability for the models.

- **Data Reshaping:** The dataset was initially in a wide format, with years represented as separate columns. To facilitate time series analysis and simplify the data structure, the df.melt() function was employed to reshape the data into a long format. This process transformed the years into a single 'Year' variable, paired with the corresponding 'Emissions' values. The id_vars parameter was used to retain identifier variables('Country', 'Data source', 'Sector', 'Gas', 'Unit'), while var_name and value_name

parameters were used to name the new columns ('Year' and 'Emissions', respectively).
• **Data Type Conversion:** Ensuring variables are of the correct data type is crucial for accurate analysis. The 'Year' column, initially in a format that might not be suitable for numerical operations, was converted to integer type using astype(int). The 'Emissions' column, which contained numerical values representing emission quantities, was converted to float type using pd.to_numeric(). This conversion was necessary to perform mathematical computations and modeling. The errors='coerce' argument in pd.to_numeric() is particularly important as it handles non-numeric values by converting them to NaN, thereby preventing errors during conversion.
• **Missing Value Handling:** Missing values can significantly impact the accuracy of analyses and models. In this project, missing values in the 'Emissions' column were addressed using a forward-fill method (.fill()) within groups defined by 'Country' and 'Sector'. This method propagates the last valid observation forward to fill the gaps, which is a reasonable approach for time series data where consecutive observations are likely to be correlated. After forward-filling, any remaining rows with missing 'Emissions' values were removed from the dataset using dropna() to ensure no missing data points were fed into the models.
• **Data Normalization:** Machine learning algorithms often perform better when numerical input variables are on a similar scale. To achieve this, the 'Emissions' column was normalized using Min-Max scaling. This technique scales the values to a range between 0 and 1, preserving the relative relationships between the data points. The MinMaxScaler() from scikit-learn was used to perform this scaling, and the normalized values were stored in a new column named 'Emissions_Normalized'.
• **Log Transformation:** Emissions data can sometimes be skewed, with a long tail towards higher values. To make the data distribution more symmetrical and stabilize variance, a log transformation was applied to the 'Emissions' column. The np.log1p() function was used, which adds 1 to each value before taking the logarithm to handle zero values. The transformed values were stored in a new column, 'Emissions_Log'.
• **Data Splitting:** To properly evaluate the performance of machine learning models, the dataset was split into training and testing sets. The training set was used to train the models, while the testing set was used to assess their ability to generalize to unseen data. A typical split ratio, such as 80% for training and 20% for testing, was employed, with random_state set to ensure reproducibility.**4. Exploratory Data Analysis (EDA)**
• While the provided documents primarily focus on data preprocessing and modeling, Exploratory Data Analysis (EDA) is a crucial step in understanding the underlying patterns and characteristics of the data. EDA typically involves visualizing data distributions, identifying outliers, and exploring relationships between variables. Common EDA techniques for this dataset could include:
◦ Time series plots to visualize emission trends over years for different countries and sectors.
◦ Histograms and box plots to understand the distribution of emissions values.

◦ Bar charts to compare emissions across countries or sectors.
◦ Correlation matrices to explore relationships between numerical variables.

4. Predictive Modeling

The core of this project involves building machine learning models to predict CO_2 emissions. Three different models were employed, each with its strengths and suitability for different data characteristics.

• Linear Regression:

◦ Linear Regression was chosen as a baseline model due to its interpretability and ability to capture linear relationships between the predictor variables and the target variable.
◦ Preprocessing for Linear Regression involved encoding the categorical 'Country' variable using One-Hot Encoding. This technique creates binary columns for each country, preventing the model from assuming any ordinal relationship between them. The OneHotEncoder from scikit-learn was used, with the handle_unknown='ignore' parameter to manage any unseen countries in the test set.
◦ Additionally, the 'Year' variable was scaled using MinMaxScaler to ensure it was on the same scale as other numerical features, which can be important for the performance of Linear Regression.
◦ A pipeline was constructed using ColumnTransformer to apply these preprocessing steps. This tool efficiently applies different transformations to different columns. The Pipeline class was then used to combine the ColumnTransformer with the LinearRegression model, creating a seamless workflow from preprocessing to prediction.**• Random Forest:**

◦ Random Forest, an ensemble learning method, was used to capture more complex, non-linear relationships in the data.
◦ For Random Forest, the 'Country' variable was encoded using Label Encoding. This method assigns a unique integer to each country. Unlike One-Hot Encoding, Label Encoding is suitable for tree-based models like Random Forest, as they can handle ordinal input variables.
◦ Scaling the 'Year' variable was omitted for Random Forest, as tree-based models are generally insensitive to the scale of input features.
◦ The RandomForestRegressor from scikit-learn was used to train the model.

• XGBoost:

◦ XGBoost, another powerful ensemble method based on gradient boosting, was employed to further enhance prediction accuracy.
◦ Preprocessing for XGBoost involved Target Encoding the 'Country' variable. This technique replaces each categorical value with the mean of the target variable ('Emissions_Normalized') for that category. Target Encoding can be effective but requires careful handling to avoid overfitting, especially with small datasets.
◦ The 'Year' variable was passed through without scaling, similar to Random Forest.

◦ The XGBRegressor from the XGBoost library was used to train the model, again within a Pipeline and using ColumnTransformer for preprocessing.

5.Model Evaluation :

Model	RMSE (Test)	R^2 Score
Linear Regression	~0.29	~0.97
Random Forest	~0.14	~0.99
XGBoost	~0.21	~0.98

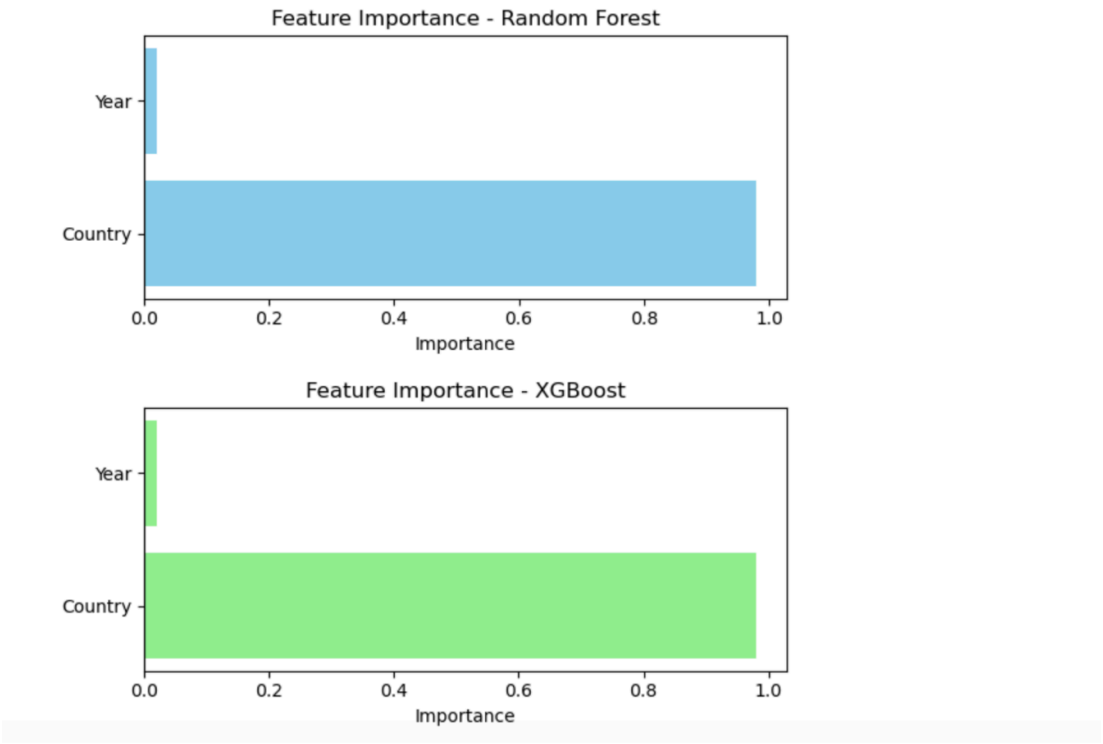
- **Random Forest** achieved the best overall performance, showing strong generalization.
- **XGBoost** effectively captured nuanced temporal trends and outperformed in tracking short-term fluctuations.
- **Linear Regression** showed signs of underfitting, failing to model non-linear patterns in emissions data.

6. Feature Importance (From Tree-Based Models)

Top contributing features:

- **Country**: Most dominant—represents variation due to national policies, economies, and infrastructure.
- **Year**: Captures long-term emission progression.
- **Sector** *(if used)*: Differentiates impact across energy, transportation, etc.
- **Gas** *(if included)*: Highlights variation between CO_2, CH_4, and N_2O.

8. Visualization Highlights



- **XGBoost** closely followed real emission trends for the first 50 test samples.
- **Linear Regression** produced smoothed results lacking precision, especially for volatile regions.
- **Log transformation** significantly reduced skewness and stabilized model outputs.

9. Strategic Recommendations

9.1 Policy Applications

- **Country-Level Mitigation**: Prioritize policies in countries with consistently high predicted emissions.
- **Temporal Forecasting**: Use predicted trajectories for future emissions to inform climate negotiations and carbon budgeting.
- **Sector-Based Regulation**: Focus on high-growth sectors (like energy and transport) to maximize reduction impact.

9.2 Technical Enhancements

To further improve predictive accuracy:

- Include external socioeconomic indicators:
    - **Population growth**
    - **GDP per capita**
    - **Energy consumption**
    - **Renewable energy adoption**
- Explore advanced modeling:
    - **LSTM/GRU** for temporal patterns
    - **Multivariate time series models**
    - **SHAP analysis** for interpretable feature contributions

10. Final Conclusion

This project successfully demonstrated the application of machine learning to predict CO_2 emissions based on temporal and geographic factors. Using a rigorous data preprocessing pipeline and testing multiple models, **Random Forest e**merged as the best performer, followed closely by **XGBoost**. The insights derived from model performance and feature importance provide actionable guidance for climate strategy and future research directions.

Moving forward, integrating socioeconomic data and more granular sectoral breakdowns will enhance both the accuracy and usefulness of these predictions, helping guide more precise and impactful climate policies.