Climate Change: Analyzing Trends and Developing Sustainable Solutions

Beyond the Data
Kammari Sadguru Sai - 23955A6715
Eerla Venkatesh - 23955A6718
Department of Data Science

January 31, 2025

Contents

1 Introduction 3

2 Requirements 3

3 Setup and Libraries Used 3

4 Data Preprocessing 3

5 Exploratory Data Analysis (EDA) 4

6 Model Implementation 4

6.1 Algorithms and Models Used 4

7 Model Evaluation 4

8 Deployment Using Streamlit 4

9 References 5

10 Code and Model Access 5

1 Introduction

This document provides an overview of the climate change analysis project, including data preprocessing, exploratory data analysis, model implementation, and deployment.

2 Requirements

The following resources are required for this project:

- Computer with internet access
- Python environment with the following libraries installed:
    - Pandas
    - NumPy
    - Matplotlib
    - Seaborn
    - Streamlit
    - Scikit-learn
- Dataset: Climate Change Dataset (2020-2024)

3 Setup and Libraries Used

The following Python libraries were imported and used in this project:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import streamlit as st
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, r2_score
```

These libraries were used for data manipulation, visualization, and analysis.

4 Data Preprocessing

The dataset underwent preprocessing steps such as handling missing values, data transformations, and feature engineering. Example:

```
df = pd.read_csv(”climate_data.csv”)
df.replace([”Unknown”, ”N/A”, ”NaN”, ””], np.nan, inplace=True)
# Replace unknowns with NaN
df.fillna(method=”ffill”, inplace=True)  # Forward fill missing values
df = pd.get_dummies(df, drop_first=True)  # Convert categorical variables
```

5 Exploratory Data Analysis (EDA)

EDA was performed using visualization techniques to understand data distributions and relationships. Examples:

```
sns.pairplot(df)
plt.show()
```

6 Model Implementation

A machine learning model was trained using the preprocessed data.

6.1 Algorithms and Models Used

- **Random Forest Regressor**: A robust ensemble learning method that combines multiple decision trees to improve prediction accuracy. - **Advantages**:

- Handles missing data effectively.
- Reduces overfitting by averaging multiple decision trees.
- Works well with non-linear relationships.

Example model training:

```
X_train, X_test, y_train, y_test = train_test_split(df.drop(’target’, axis=
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

7 Model Evaluation

After training, the model’s performance was evaluated using Mean Absolute Error (MAE) and R-squared (R^2) score:

```
y_pred = model.predict(X_test)
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(”Mean Absolute Error:”, mae)
print(”R Score:”, r2)
```

8 Deployment Using Streamlit

This model was deployed as a web application using Streamlit. The key functionalities include:

- Uploading a climate dataset in CSV format.
- Handling missing and non-numeric values automatically.
- Allowing users to select features and target variables for prediction.

- Training a Random Forest model and evaluating it in real-time.
- Visualizing actual vs. predicted values using scatter plots.
- Accepting new user input and making predictions based on the trained model.

Example of user interaction:

```
input_data = {}
for feature in feature_cols:
    input_data[feature] = st.number_input(f"Enter {feature}", value=float(

if st.button("Predict"):
    input_df = pd.DataFrame([input_data])
    prediction = model.predict(input_df)
    st.write(f"Predicted {target_col}: {prediction[0]:.2f}")
```

9 References

For additional details, refer to:

- Python documentation: `https://docs.python.org/3/`
- Pandas documentation: `https://pandas.pydata.org/`
- Scikit-learn documentation: `https://scikit-learn.org/`
- Streamlit documentation: `https://docs.streamlit.io/`

10 Code and Model Access

The complete source code and trained model can be accessed from the following links:

- **Code:** Climate Change Analysis
- **Model Deploy:** Climate Change Predection Model