Milestone 2: Feature Engineering, Feature Selection, and Data Modeling

Introduction to Data Science

Team: Sainath Chettupally (939444937), Roop Yaswanth Nagabhairava (45439754)

Objective: The objective of this project is to create a chatbot capable of analyzing and interacting with U.S. power plant data and environmental trends. The chatbot will allow users to query and dynamically explore the relationships between U.S. power plant data, pollution levels, and temperature changes within the United States.

The goal of this milestone is to build a machine learning model that predicts whether a new U.S. power plant will have a Positive, Negative, or Neutral environmental impact.

The prediction is based on air quality, temperature data, and plant features from before the plant was commissioned. This model will later power a chatbot that helps users explore power plant impacts in the U.S.

Type of tool: Conversational Agent (Chatbot): Allows users to query the power plant database and related datasets using natural language.

Data Used:

1) U.S. Power Plant Database: Contains information on power plants in the United States, including location, capacity, fuel type, and operational status.

(https://www.kaggle.com/datasets/ramjasmaurya/global-powerplants)

2) U.S. Pollution Dataset (AirNow API Documentation) Contains air quality index (AQI) data for various locations in the U.S.

3) U.S Temperature dataset (Collected from NOAA using Open-meteo API) contains temperature data for various locations, recorded quarterly.

The respective names of datasets are:

merged_aqi_cleaned.csv

optimized_aqi_request_data_with_temperature.csv

global_power_plant_cleaned_usa1_clustered.csv

Final dataset is saved as Filtered_data.csv

The main attributes of final data are:

- **cluster:** Numerical identifier grouping power plants and environmental measurements in the same geographic area. This is the primary idenfier.
- **year:** Calendar year of the environmental measurement or power plant data point.
- **mean_aqi:** Average Air Quality Index value for a specific cluster and year, with higher values indicating worse air quality.
- **avg_temp:** Mean temperature in Celsius for a specific cluster and year, derived from temperature_2m_mean measurements.
- **country:** Nation where the power plant is located
- **capacity_mw:** Maximum electricity generation capacity of the power plant measured in megawatts.
- **latitude:** North-south geographic coordinate of the power plant location.
- **longitude:** East-west geographic coordinate of the power plant location.
- **primary_fuel:** Main energy source used by the power plant (e.g., solar, wind, coal, gas).
- **commissioning_year:** Year when the power plant began operations.
- **generation_gwh_2013:** Total electricity generation in gigawatt-hours for the year 2013.
- **generation_gwh_2014:** Total electricity generation in gigawatt-hours for the year 2014.
- **generation_gwh_2015:** Total electricity generation in gigawatt-hours for the year 2015.
- **generation_gwh_2016:** Total electricity generation in gigawatt-hours for the year 2016.
- **generation_gwh_2017:** Total electricity generation in gigawatt-hours for the year 2017.
- **impact_period:** Categorical variable indicating whether the record is from before ('pre'), during ('commissioning'), or after ('post') the plant's commissioning.

The shape of final dataset shape is: (18496, 20)

Tech Stack:

Programming Languages: Python

Libraries and frameworks:

- **Data Manipulation:** Pandas, NumPy, Scipy
- **Data Visualization:** Matplotlib, Seaborn
- **Machine Learning:** scikit learn

Project Timeline:

Milestone	Task	Deadline
Milestone 3:	Evaluation, Interpretation, Tool Development and Presentation.	23rd April 2025

Data Preprocessing:

We have performed some preprocessing steps such as converted DateObserved and time columns to datetime format, Extracted year from date fields, Ensured commissioning_year was numeric for merging and filtering.

Environmental Data Aggregation

- AQI data was grouped by cluster and year, taking the mean AQI per group.
- Temperature data was similarly grouped by cluster and year, calculating the average temperature.
- These two were merged to create a cluster-year level environmental dataset with AQI and temperature indicators.

Merging and Filtering by Commissioning Year

- Merged the cluster-level environmental dataset with plant metadata on cluster.
- Filtered to retain data from two years before to three years after each plant's commissioning year.
- Added a new column impact_period to label whether a year falls in the pre, post, or commissioning phase.

Environmental Impact Labeling

- Grouped data by plant and commissioning year.
- Calculated average AQI and temperature before and after commissioning.
- Created features: delta_aqi, delta_temp, comm_aqi, comm_temp.
- Based on changes, assigned labels:
    - Positive: AQI ↓ > 3 units and Temp ↓ > 0.5°C
    - Negative: AQI ↑ > 3 units or Temp ↑ > 0.5°C
    - Neutral: All other cases

Exploratory Data Analysis:

- Total power plants analyzed: 2852
- Time period covered: 2010 to 2018
- Number of fuel types: 12

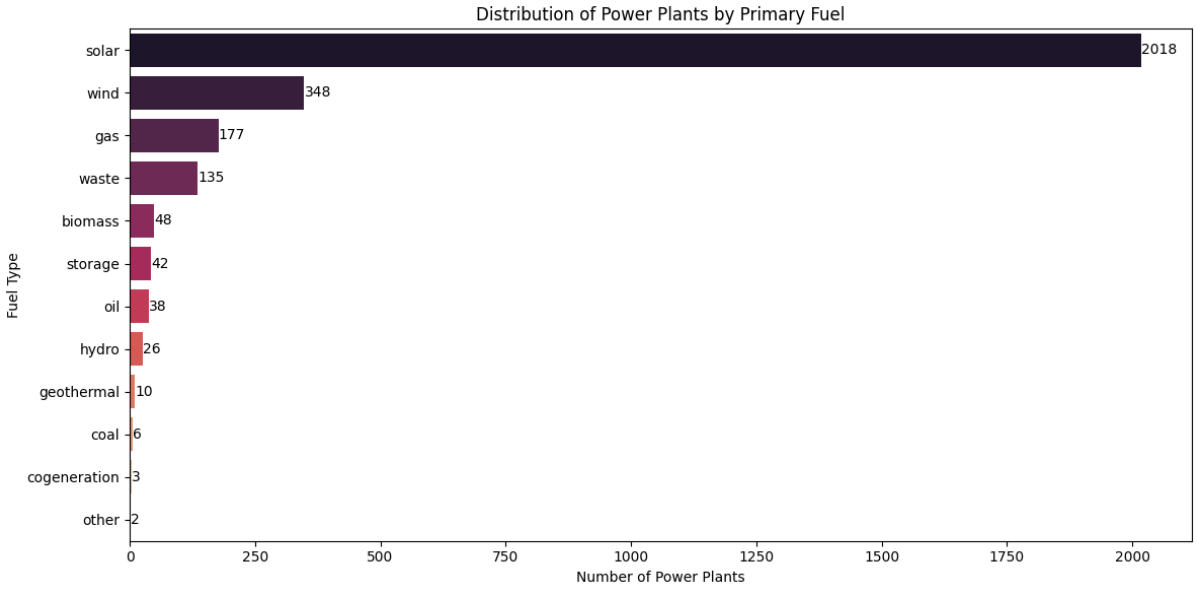


Figure 1: Distribution of power plantsby Primary fuel

From the above analysis, we can see that more than 80% of power plants are run by using solar as their primary fuel which is the renewable energy.

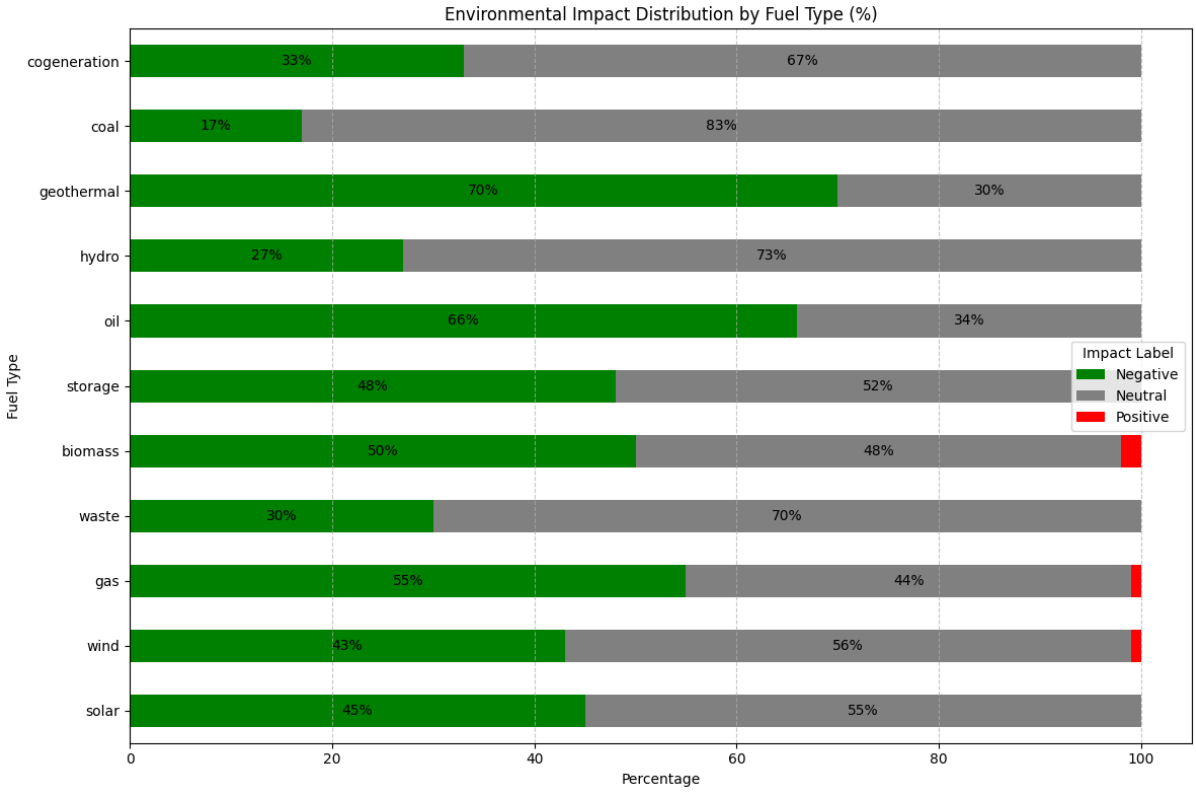


Figure 2: Environment Impact Distribution by fuel type

From the above analysis, we can find that geothermal then oil is mostly negatively affected on the environment.

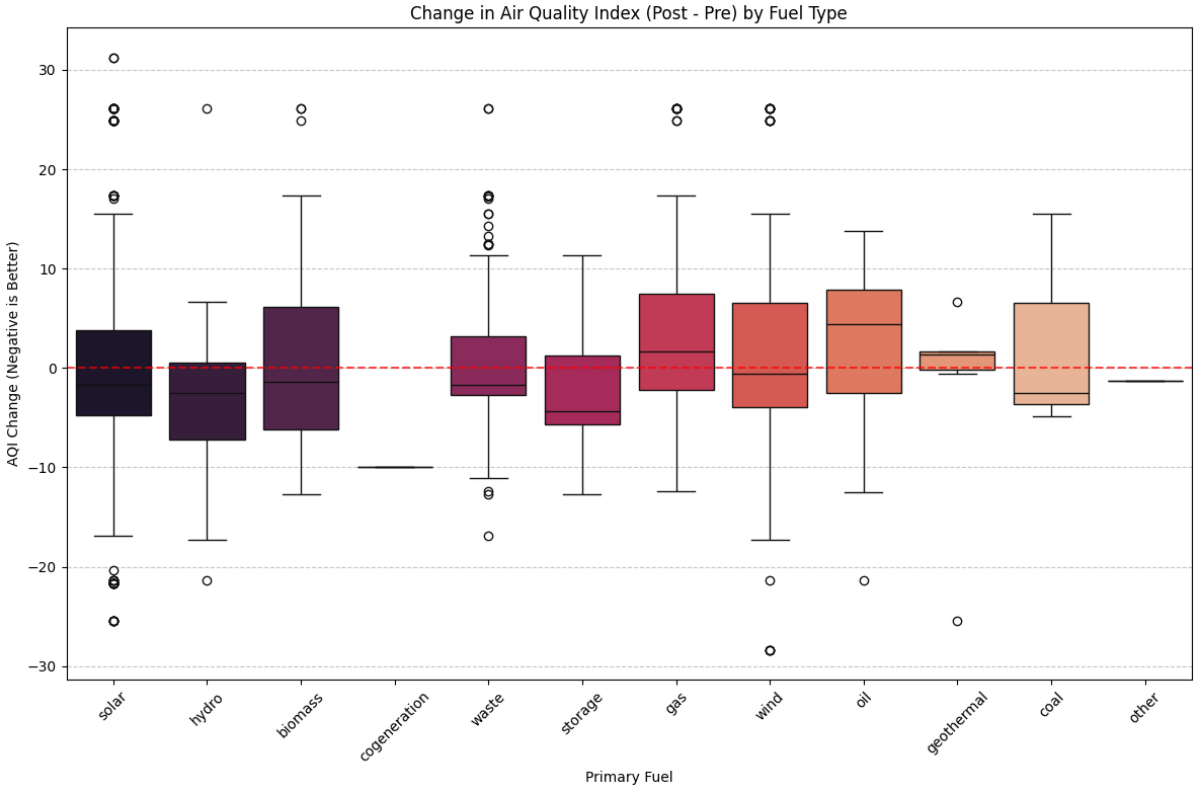


Figure 3: AQI Change by fuel type

The above visualizations show the impact of fuel on AQI. From the visualization, we can say that gas, oil, geothermal fuel types negatively affect the Air quality.

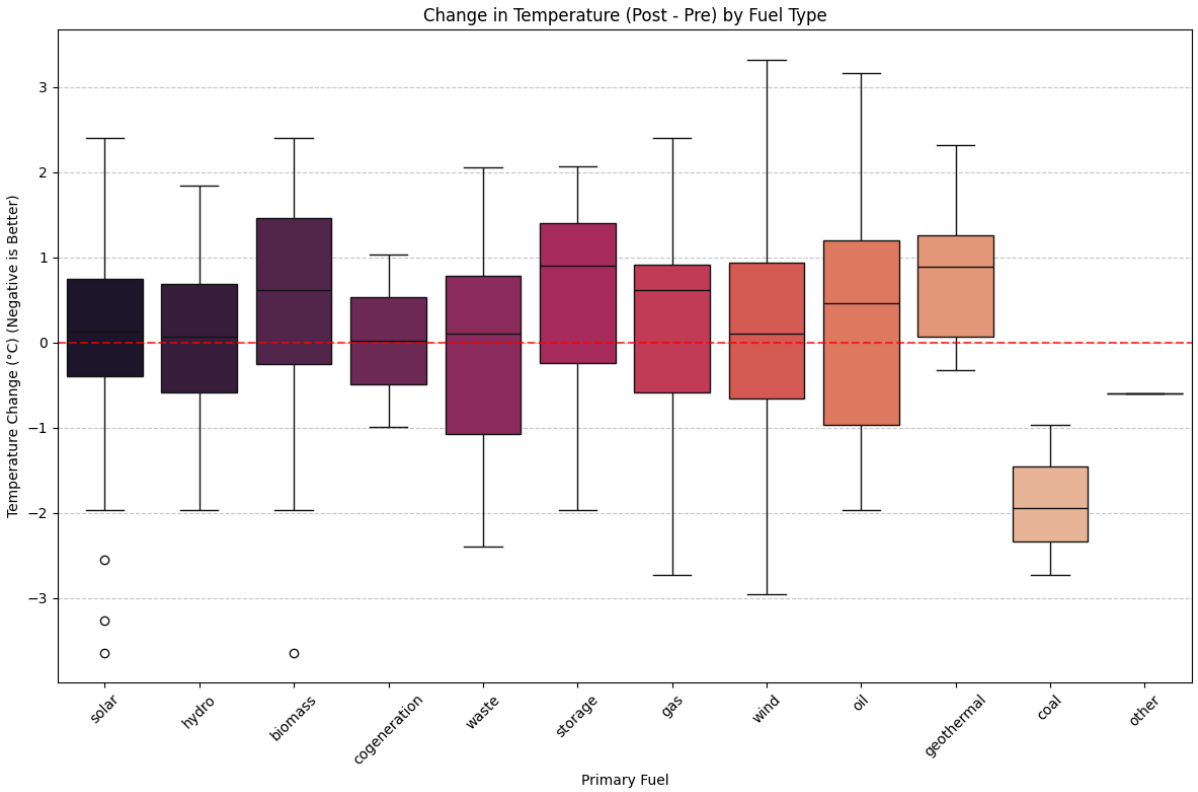


Figure 4: Temperature change by fuel type

From the temperature visualization, we can say that solar, hydro, coal, wind, are the fuel types which doesn't affect the temperature much, remaining all have a little to significant effect on increase in temperature.

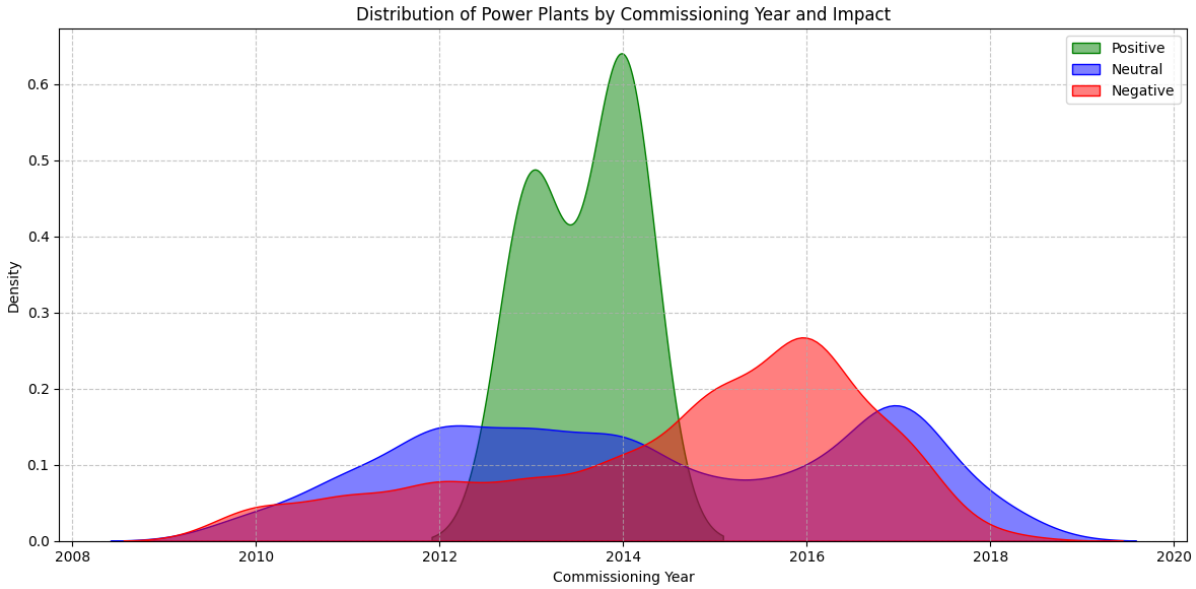


Figure 5: Commissioning year trends

The power plants which are started in the year between 2012 – 2015 has a positive impact with higher density.

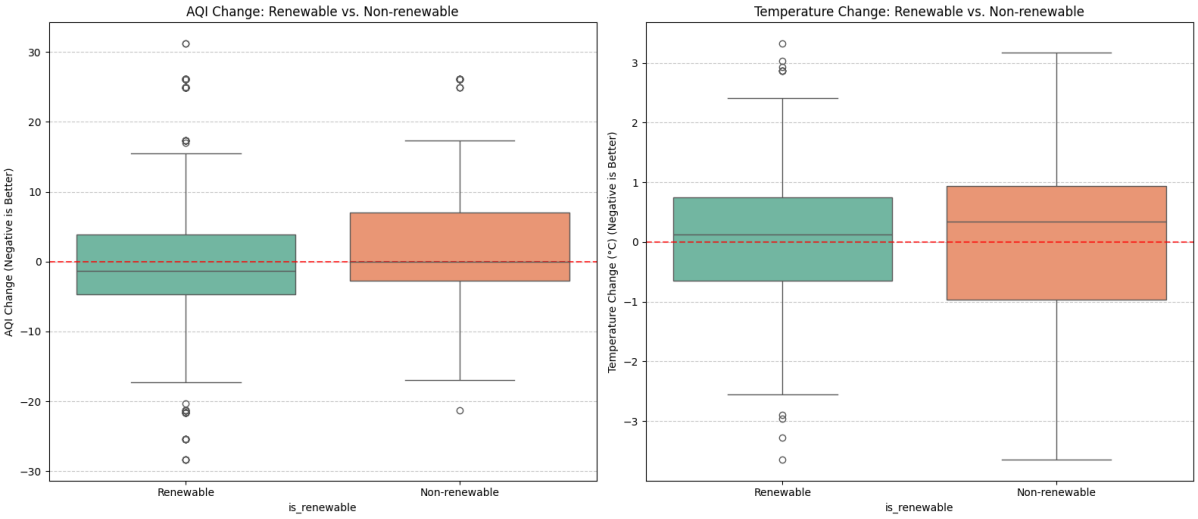


Figure 6: AQI and temperature changes wrt energy sources.

This clearly depicts that the renewable energy sources are better for the environment, as there is a decrease in 2% of AQI and only a little increase in temperature when using renewable energy sources, where as non-renewable energy sources has negative effect on environment.

Feature Engineering:

Feature engineering was implemented to transform raw data into more meaningful features that could better capture power plant environmental impacts. We created several new derived features and encoded categorical variables to enhance model performance.

```
# 1. Plant Age: Older plants may have different environmental influence
patterns
df['plant_age'] = 2024 - df['commissioning_year']

# 2. Renewable Indicator: Classify whether the plant uses renewable fuel
renewables = ['solar', 'wind', 'hydro', 'geothermal']
df['is_renewable'] =
df['primary_fuel'].str.lower().isin(renewables).astype(int)

# 3. Interaction Feature: Captures compounding effect of AQI and temperature
changes
df['aqi_temp_interaction'] = df['delta_aqi'] * df['delta_temp']

# 4. One-Hot Encode fuel types for categorical modeling
df = pd.get_dummies(df, columns=['primary_fuel'], drop_first=True)

# 5. Define target variable
target = 'impact_label'

# 6. Define input features (dropping metadata)
drop_cols = ['name', 'cluster', 'commissioning_year', 'comm_aqi', 'comm_temp']
X = df.drop(columns=[target] + drop_cols)
y = df[target]
```

Creating New Features:

The three key features created beyond simple encoding are:

1. **Plant Age (plant_age = 2024 - commissioning_year)**:
    - **Justification**: Older plants may behave differently due to evolving regulations, equipment wear, or technological upgrades over time. This feature captures temporal effects on environmental impact that aren't directly measured in the raw data.
    - **Implementation**: Calculated as the difference between current year and commissioning year.
2. **Renewable Energy Indicator (is_renewable)**:
    - **Justification**: Renewable energy plants typically have fundamentally different environmental footprints than fossil fuel plants. This binary feature provides a simplified signal about the plant's environmental characteristics.
    - **Implementation**: Created by checking if the primary fuel is in the renewable energy list.

3. **AQI-Temperature Interaction (aqi_temp_interaction = delta_aqi * delta_temp)**:
    - **Justification**: This interaction term captures the compounding effects when both air quality and temperature change simultaneously. A plant causing both air quality degradation and temperature increases would have a higher negative environmental impact than one affecting only one metric.
    - **Implementation**: Multiplied the delta values of AQI and temperature to produce a single metric representing combined environmental stress.

Categorical Variable Encoding:

We applied one-hot encoding to the primary_fuel categorical variable because:

primary_fuel is a nominal variable with no inherent ranking (e.g., "solar" isn't greater or less than "wind") and many models (Logistic Regression, Decision Trees, Random Forests) perform better with binary features than with arbitrary numeric labels that might imply false ordinal relationships.

The dataset contains relatively few unique fuel types, so one-hot encoding doesn't dramatically increase dimensionality. Label encoding would assign arbitrary integers (e.g., coal = 0, gas = 1, solar = 2), potentially misleading tree-based models into assuming ordinal relationships between fuel types.

```
# One-Hot Encode fuel types for categorical modeling
df = pd.get_dummies(df, columns=['primary_fuel'], drop_first=True)
```

The drop_first=True parameter was used to avoid the dummy variable trap (perfect multicollinearity), enhancing model stability.

We also dropped non-informative identifiers like name, cluster, and commissioning-year-specific values (comm_aqi, comm_temp) which doesn't help in model prediction.

Feature Selection:

To ensure optimal model performance and interpretability, we used multiple methods to evaluate feature importance and select the most relevant subset of features.

Feature Importance Evaluation:

1. **Correlation Analysis:** We computed a correlation matrix to identify relationships between features:

```
plt.figure(figsize=(12, 8))
sns.heatmap(df.corr(numeric_only=True), annot=True, fmt=".2f",
cmap="coolwarm")
plt.title("Feature Correlation Heatmap")
```

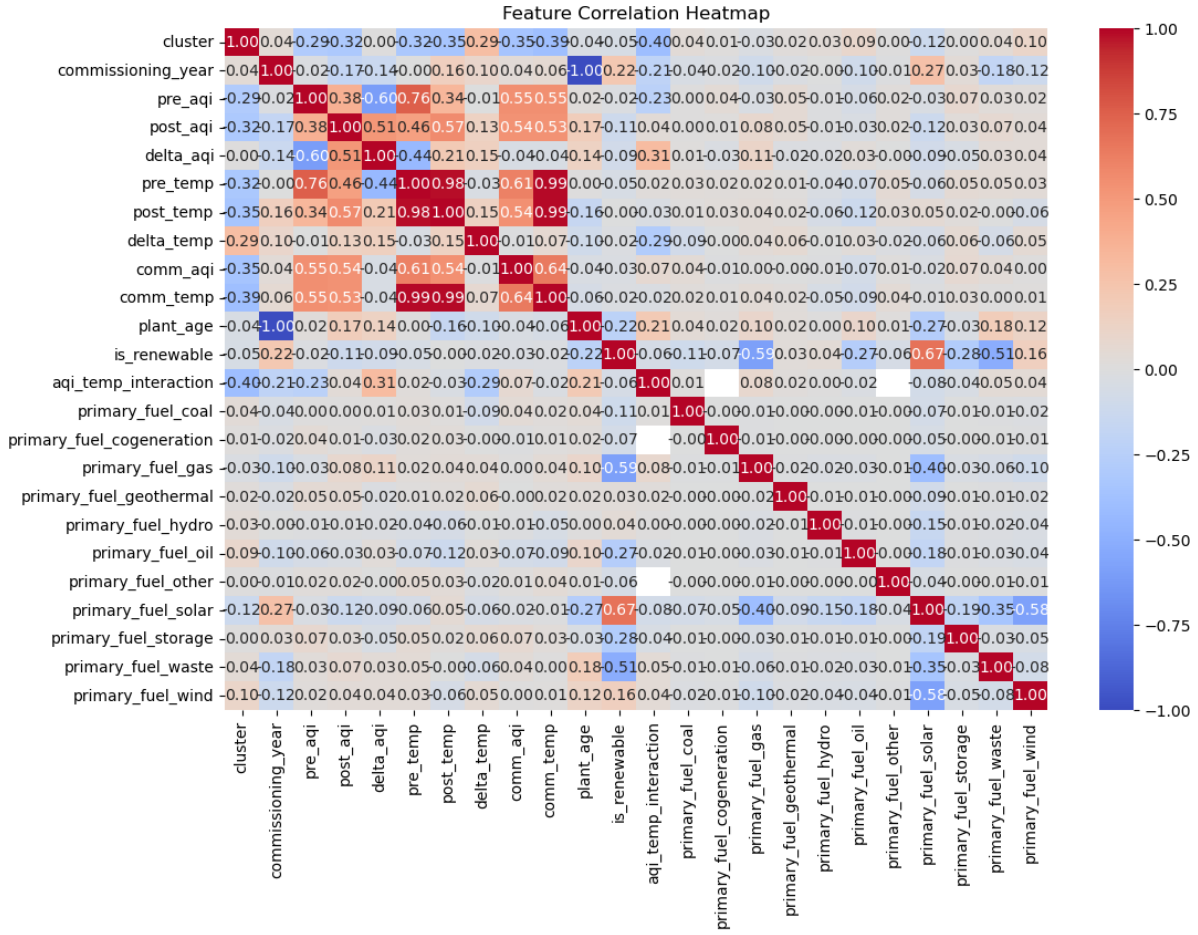


Figure 7: All features correlation matrix

The key findings from correlation analysis is:

- High multicollinearity between temperature features (comm_temp, post_temp, pre_temp)
- Strong correlations between AQI features (comm_aqi, pre_aqi, post_aqi)
- delta_aqi and delta_temp showed strong correlations with the target behavior
- Weak correlations between metadata features (cluster, commissioning_year) and the target

2. Random Forest Feature Importance:

We trained a Random Forest model to quantify feature importance:

```
rf_temp = RandomForestClassifier(random_state=42)
rf_temp.fit(X, y)

importances = pd.Series(rf_temp.feature_importances_, index=X.columns)
importances.sort_values(ascending=False).plot(kind='barh', figsize=(10, 6))
```

- Baseline measurements (pre_aqi, pre_temp) contributed moderate predictive value
- The engineered interaction term aqi_temp_interaction showed meaningful contribution

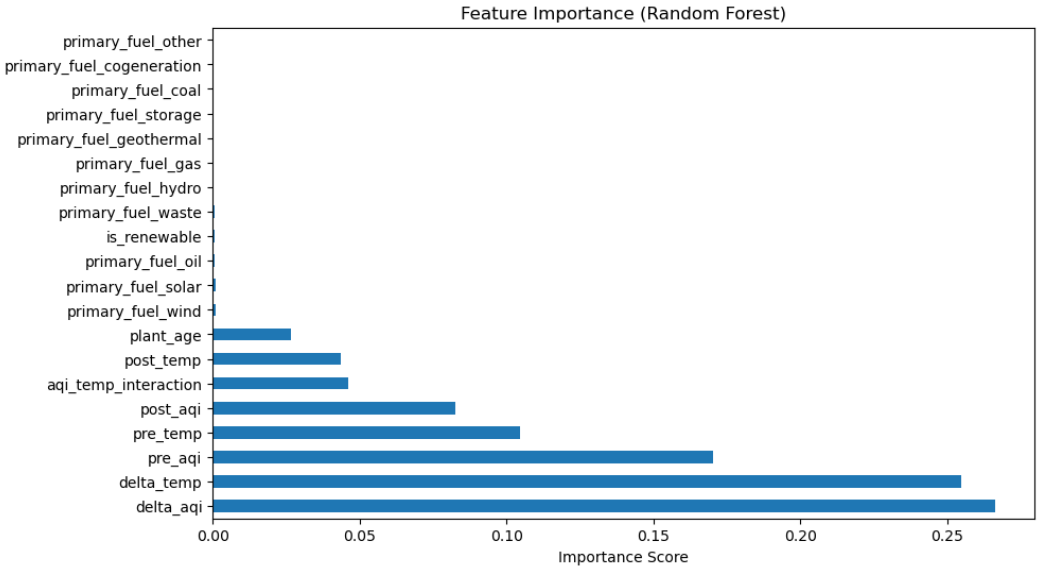


Figure 8: Feature importance using random forest

Feature Selection Decision:

Based on our evaluation, we made the following decisions:

1. **Features Retained**:
    - Environmental_metrics: delta_aqi, delta_temp, pre_aqi, pre_temp, post_aqi, p ost_temp
    - Engineered features: plant_age, is_renewable, aqi_temp_interaction
    - One-hot encoded primary_fuel_ variables
2. **Features Excluded**:
    - Metadata features: name, cluster, commissioning_year
    - Redundant commissioning measurements: comm_aqi, comm_temp

Tree-based feature importance was particularly valuable for our selection process because it captures non-linear relationships and is robust to feature scaling, complementing the linear relationships identified through correlation analysis.

Final Selected Features: After considering interpretability, predictive power, and correlation redundancy, we used the following features:

• pre_aqi, pre_temp, plant_age, is_renewable, aqi_temp_interaction, primary_fuel_ variables.

These features were passed forward for training and evaluation in the modeling phase.

Dimensionality reduction:

We deliberately chose not to implement formal dimensionality reduction techniques like PCA or LASSO for several key reasons:

1. **Feature Set Already Compact:** After our manual feature selection, the final feature set was already relatively small (~10 features depending on the number of one-hot encoded fuel types). This modest dimensionality didn't justify the additional complexity of formal reduction techniques.
2. **Tree-Based Models:** Our performing models were tree-based (Random Forest), which:
    - Handle high-dimensional data effectively without requiring dimensionality reduction, and are resistant to the effects of irrelevant features, can also implicitly perform feature selection during training
3. **Domain Knowledge Application:** We prioritized using domain knowledge to select meaningful features over algorithmic dimension reduction, ensuring that features had clear relationships to environmental impacts.

Instead of automatic dimensionality reduction, we relied on careful manual feature selection based on correlation analysis, feature importance scores, and domain expertise, which resulted in a interpretable feature set that will maintain strong predictive performance.

Data Modelling:

Data Splitting Approach: To ensure robust and reliable model evaluation, we implemented a stratified train-test split with the following parameters:

```
# Stratified Train-Test Split
X_train_ml, X_test_ml, y_train_ml, y_test_ml = train_test_split(
    X_ml, y_ml, test_size=0.2, stratify=y_ml, random_state=42
)
```

- **Test size of 20%**: Provides sufficient data for evaluation while retaining 80% for training
- **Stratification**: Critical due to severe class imbalance in our dataset, particularly for the rare 'Positive' impact class
- **Fixed random state**: Ensures reproducibility of results and consistent model comparison

After splitting, we applied mean imputation to handle missing values, preserving the integrity of our train-test separation by fitting the imputer only on training data:

```
# Imputation strategy
imputer = SimpleImputer(strategy='mean')
X_train_ml_imputed = imputer.fit_transform(X_train_ml)
X_test_ml_imputed = imputer.transform(X_test_ml)
```

For our predictive models, we carefully selected only features available before or at the time of power plant commissioning, eliminating any data leakage. This approach creates a realistic scenario for predicting a new plant's potential environmental impact based solely on pre-operational data.

```
true_ml_features = [
    'pre_aqi', 'pre_temp', 'plant_age', 'is_renewable'
] + [col for col in df.columns if col.startswith('primary_fuel_')]
```

Model Selection and Training:

We implemented three distinct classification models, to capture different patterns in the data:

1. **Logistic Regression**:
    - A linear model as our baseline
    - Offers high interpretability and efficiency
    - Well suited for understanding feature importance through coefficients
2. **Decision Tree**:
    - Non-linear model that captures complex hierarchical relationships
    - Handles feature interactions automatically
    - Provides clear decision rules for classifying environmental impacts
3. **Random Forest**:
    - Ensemble method that reduces overfitting through aggregation
    - Captures non-linear relationships while providing feature importance
    - Robust against noise and outliers

Each model was trained on identical pre-processed data to ensure fair comparison:

```
# Initialize models
logreg_ml = LogisticRegression(max_iter=1000)
dtree_ml = DecisionTreeClassifier(random_state=42)
rf_ml = RandomForestClassifier(random_state=42)

# Train models
logreg_ml.fit(X_train_ml_imputed, y_train_ml)
dtree_ml.fit(X_train_ml_imputed, y_train_ml)
rf_ml.fit(X_train_ml_imputed, y_train_ml)
```

Model Evaluation and Comparison:

We used multiple metrics to assess model performance:

- **Classification Report** (precision, recall, F1-score):
    - Essential for understanding class-specific performance in our imbalanced dataset, Precision measures reliability of positive predictions; recall measures ability to find all positive instances
- **Confusion Matrix**:
    - Provides detailed insight into model errors and class imbalance handling. Diagonal elements represent correct classifications; off-diagonal elements represent misclassifications
- **ROC Curves and AUC**:
    - Threshold-independent evaluation that assesses discrimination ability. AUC of 1.0 indicates perfect discrimination; 0.5 indicates random guessing.

Performance Results

Classification Metrics:

Model	Class	Precision	Recall	F1-score
Logistic Regression	Negative	0.76	0.66	0.70
	Neutral	0.75	0.83	0.79
	Positive	0.00	0.00	0.00
	Accuracy			**0.75**
Decision Tree	Negative	0.98	1.00	0.99
	Neutral	1.00	0.98	0.99
	Positive	1.00	1.00	1.00
	Accuracy			**0.99**
Random Forest	Negative	0.95	0.98	0.96
	Neutral	0.98	0.96	0.97
	Positive	0.00	0.00	0.00
	Accuracy			**0.97**

The confusion matrix is as follows:

Logistic regression:

```
[[168  87   0]
 [ 54 261   0]
 [  0   1   0]]
```

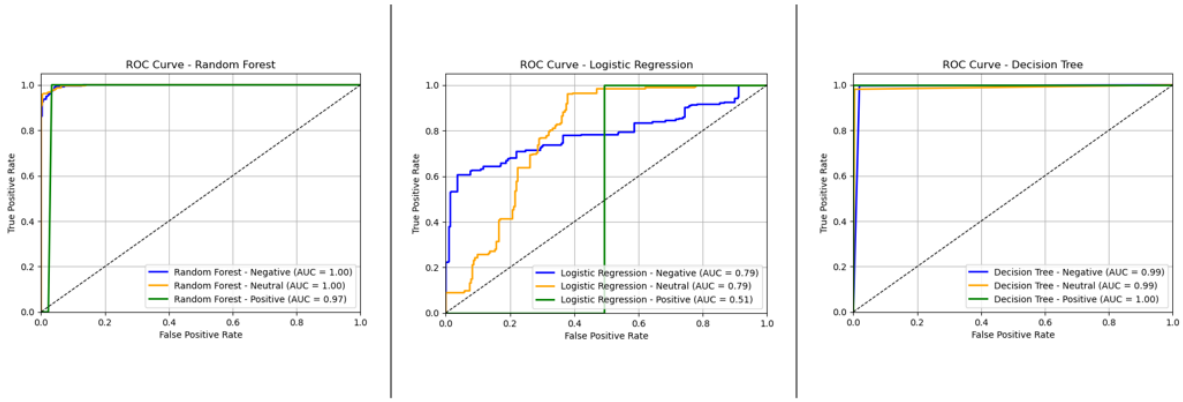
Decision Tree:

```
[[254   1   0]
 [  6 309   0]
 [  0   0   1]]
```

Random Forest:

```
[[249   5   1]
 [ 12 303   0]
 [  1   0   0]]
```

ROC-AUC Performance:



Comparative analysis:

1. Overall Performance:
    - Decision Tree achieved the highest accuracy (99%) with perfect precision and recall across all classes.
    - Random Forest performed strongly overall (97% accuracy) but failed on the rare Positive class.
    - Logistic Regression showed moderate performance (75% accuracy) with weaknesses in identifying the minority class.
2. Class Imbalance Handling:
    - The dataset exhibits severe class imbalance with the Positive class representing only 0.2% of test samples.
    - Decision Tree was remarkably successful at correctly identifying the single Positive instance.
    - Both Logistic Regression and Random Forest failed to correctly predict any Positive instances.

3. Discriminative Ability:
    - ROC analysis revealed excellent class separation abilities for tree-based models (AUC > 0.97).
    - Logistic Regression showed moderate discrimination for Negative and Neutral classes (AUC = 0.79).
    - Only the Decision Tree model maintained perfect discrimination for the critical Positive class.
4. Error Analysis:
    - Logistic Regression confused 87 Negative instances as Neutral and 54 Neutral instances as Negative.
    - Random Forest made fewer errors, primarily confusing 12 Neutral instances as Negative.
    - Decision Tree achieved near-perfect performance with only 7 misclassifications out of 571 test instances.

CONCLUSION:

Based on our evaluation, the **Decision Tree** classifier emerged as the superior model for environmental impact prediction, demonstrating:

1. Highest overall accuracy (99%).
2. Perfect performance on the critical minority class (Positive environmental impact).
3. Excellent AUC scores across all three classes.
4. Strong interpretability through its decision rules.

The Random Forest showed competitive performance but struggled with the rare Positive class, while Logistic Regression demonstrated the limitations of linear models in handling this complex, imbalanced classification problem.