Project Title: Predicting Carbon Emissions from Flight Data

---

1. Project Overview

The objective of this project is to build a predictive machine learning model that estimates **carbon emissions** of flights based on a range of aircraft, flight, and environmental features. The dataset used is synthetically generated but represents real-world flight parameters, which makes it suitable for developing a proof-of-concept model and analysis pipeline.

---

2. Dataset Description

The dataset realistic_synthetic_flight_data_single_file.csv contains **million rows** and **50 columns**, each representing distinct measurable or categorical features related to:

- **Flight operations** (e.g., Flight_Duration, Distance, Taxi_Time)
- **Aircraft specifications** (e.g., Aircraft_Weight, Engine_Hours, Fuel_Consumption)
- **Environmental conditions** (e.g., Altitude, Humidity_Level, Outside_Temperature)
- **Performance metrics** (e.g., Speed, Thrust_Level, Fuel_Efficiency)
- **Emission indicators** (e.g., CO2_Emission, SO2_Emission)
- **Maintenance and operational states** (e.g., Maintenance_Flag, Sensor_Error_Code)

The **target variable** is:

- **Carbon_Emissions** — the amount of carbon emitted during a flight (in tons)

---

3. Project Flow

1. **Data Ingestion**
2. **Data Exploration & Profiling**
3. **Exploratory Data Analysis (EDA)**
4. **Feature Engineering**
5. **Model Preparation**
6. **Model Evaluation**
7. **Conclusion & Recommendations**

---

4. Data Ingestion

- Load the dataset into a PySpark or Pandas environment.
- Check for schema correctness, missing data, duplicate records, and data types.

---

5. Exploratory Data Analysis (EDA)

5.1. General Data Profiling

- Total rows, columns
- Data types per column
- Memory usage and loading time
- Basic statistics (mean, median, min, max, std) using .describe()

5.2. Target Variable Exploration: Carbon_Emissions

- Distribution plot (histogram / KDE)
- Outlier detection (boxplot)
- Skewness and kurtosis
- Check if data is normally distributed or needs transformation (e.g., log)

5.3. Missing Value Analysis

- Count and percentage of missing values per column
- Visualization using heatmaps or missingno plots
- Strategy to handle missing values: imputation vs. deletion

5.4. Correlation Analysis

- Compute Pearson correlation matrix
- Visualize heatmap for top correlated features with Carbon_Emissions
- Detect multicollinearity (VIF or pairwise correlations)

5.5. Univariate Analysis

- Distributions of key features like Flight_Duration, Fuel_Consumption, Speed, etc.
- Use histograms, KDE plots, and boxplots
- Log transformation for skewed distributions

5.6. Bivariate Analysis

- Scatter plots of each feature vs. Carbon_Emissions
- Trendlines to observe linear/non-linear relationships
- Categorical columns: bar plots showing average emissions per category (if any)

5.7. Multivariate Exploration

- 3D scatter plots (e.g., Fuel_Consumption vs. Flight_Duration vs. Carbon_Emissions)
- Feature combinations that might jointly impact emissions
- PCA or t-SNE for pattern detection

5.8. Outlier Detection

- Identify extreme values in continuous features
- Use Z-score or IQR methods
- Impact of outlier removal on Carbon_Emissions

---

6. Feature Engineering

- **Transformations**: Log scaling, normalization, or standardization
- **Interaction Terms**: Combine Speed * Aircraft_Weight or Altitude / Distance
- **Derived Features**:
  - Fuel per km = Fuel_Consumption / Distance
  - Emissions per km = Carbon_Emissions / Distance
  - Efficiency Score = Fuel_Efficiency / Thrust_Level
- **Handling multicollinearity**: Drop or combine highly correlated features

---

7. Model Preparation

7.1. Train-Test Split

- 80–20 or 70–30 split
- Stratify if using categories (e.g., aircraft type in a real-world scenario)

7.2. Model Candidates

- **Linear Regression**

- **Random Forest Regressor**
- **Gradient Boosted Trees (e.g., XGBoost)**
- **Support Vector Regressor**
- **Neural Networks (if using deep learning frameworks)**

7.3. Baseline Model

- Mean Predictor or Linear Regression as baseline

7.4. Model Evaluation Metrics

- **R^2 Score**
- **Mean Absolute Error (MAE)**
- **Root Mean Squared Error (RMSE)**
- **Residual Plots**

---

8. Model Tuning and Optimization

- Use **Grid Search** or **Random Search** for hyperparameter tuning
- **Cross-validation** (k-fold or time-based if temporal data)
- Feature importance plots from tree-based models

---

9. Conclusion & Recommendations

- Highlight the **most important features** influencing carbon emissions
- Provide **recommendations** to reduce emissions:
    - Optimizing fuel consumption
    - Adjusting cruise speed or altitude
    - Monitoring engine conditions
- Evaluate whether the model is production-ready or requires more robust real-world data