CFD Project Report - Drag & Lift Prediction Using Random Forest Regressio

1. Objective

The goal of this project is to predict the drag (Cd) and lift (Cl) coefficients of various airfoil designs using surface coefficient data and a machine learning model (Random Forest Regressor). The dataset used consists of over 860,000 samples with 70+ features derived from CFD simulations.

2. Data Overview

File Name: combinedAirfoilDataLabeled.csv

Shape: 867,098 rows × 71 columns

Key Features: Coefficients from the upper surface (e.g., upperSurfaceCoeff1, ..., upperSurfaceCoeff70)

Target Columns: coefficientDrag, coefficientLift

Dropped Columns: airfoilName (categorical), botXTR (assumed non-numeric or irrelevant)

3. Preprocessing

Loaded the dataset using pandas.read_csv().

Removed rows with missing values using dataset.dropna().

Extracted:

- X (features): All surface coefficients

- y (targets): coefficientDrag, coefficientLift

Split the data:

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=29)

Standardized the features using StandardScaler to normalize inputs.

4. MemoryError Issue and Resolution

Due to the large size of the dataset, a MemoryError occurred during the training of the

RandomForestRegressor. To resolve this:

- Downsampled the dataset:

  x_small = x_train_scaled[:10000]

  y_small = y_train[:10000]

- Reduced model complexity:

  model_cd = RandomForestRegressor(n_estimators=20, random_state=29)

5. Model Training and Prediction

Trained the model using the sampled data:

model_cd.fit(x_small, y_small)

Predicted on the full test set:

y_pred = model_cd.predict(x_test_scaled)

Output example:

[[0.0893 1.1772]

 [0.0085 -0.3298]

 [0.0334 -0.5504]

  ...]

6. Additional Plots

Included below are additional plots generated during the analysis.

- Histogram with KDE for Lift Coefficient

- Actual vs Predicted Drag Coefficient

- Actual vs Predicted Lift Coefficient

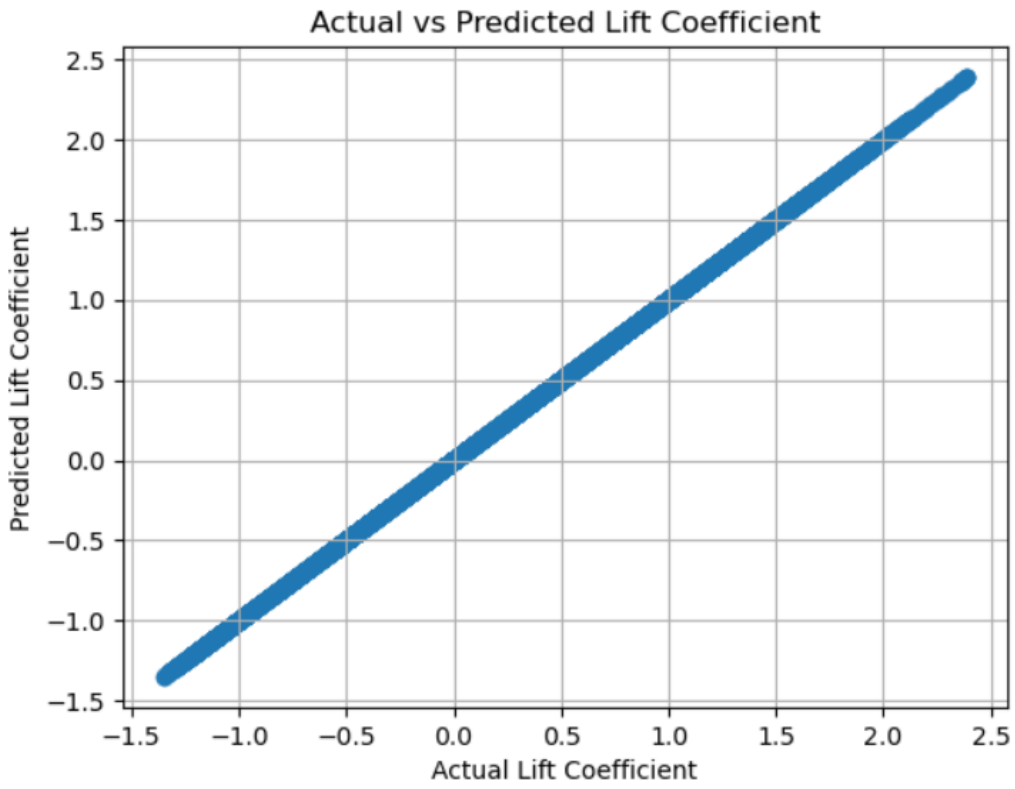
7. Future Improvements

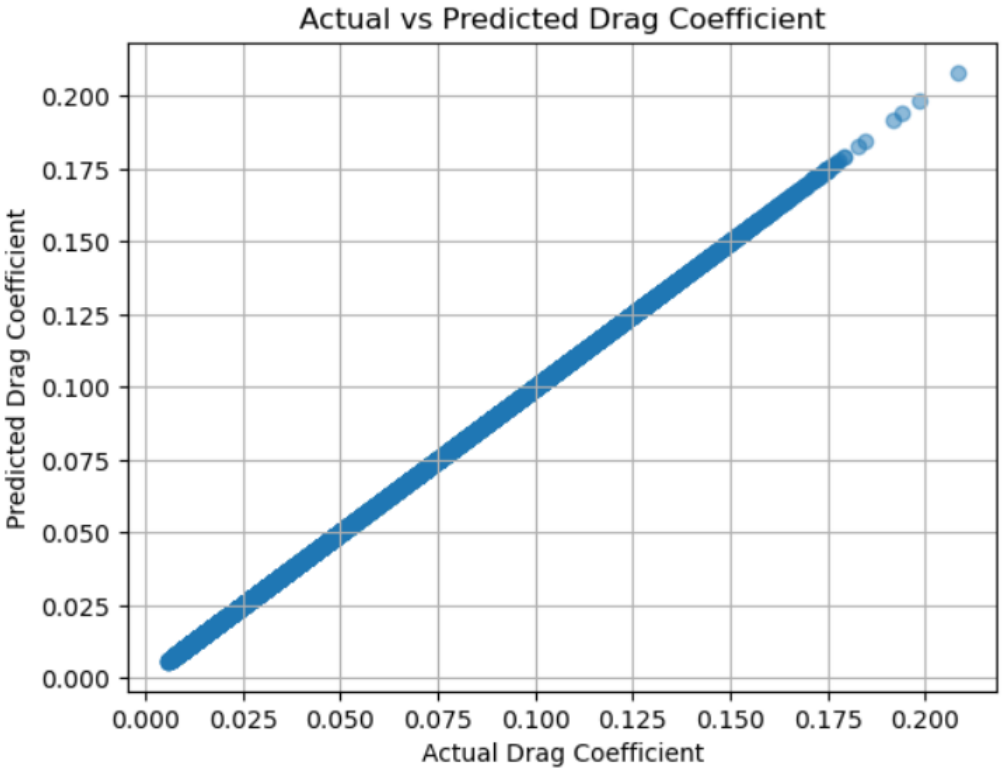
- Train on larger subsets using batch-wise training or cloud compute.

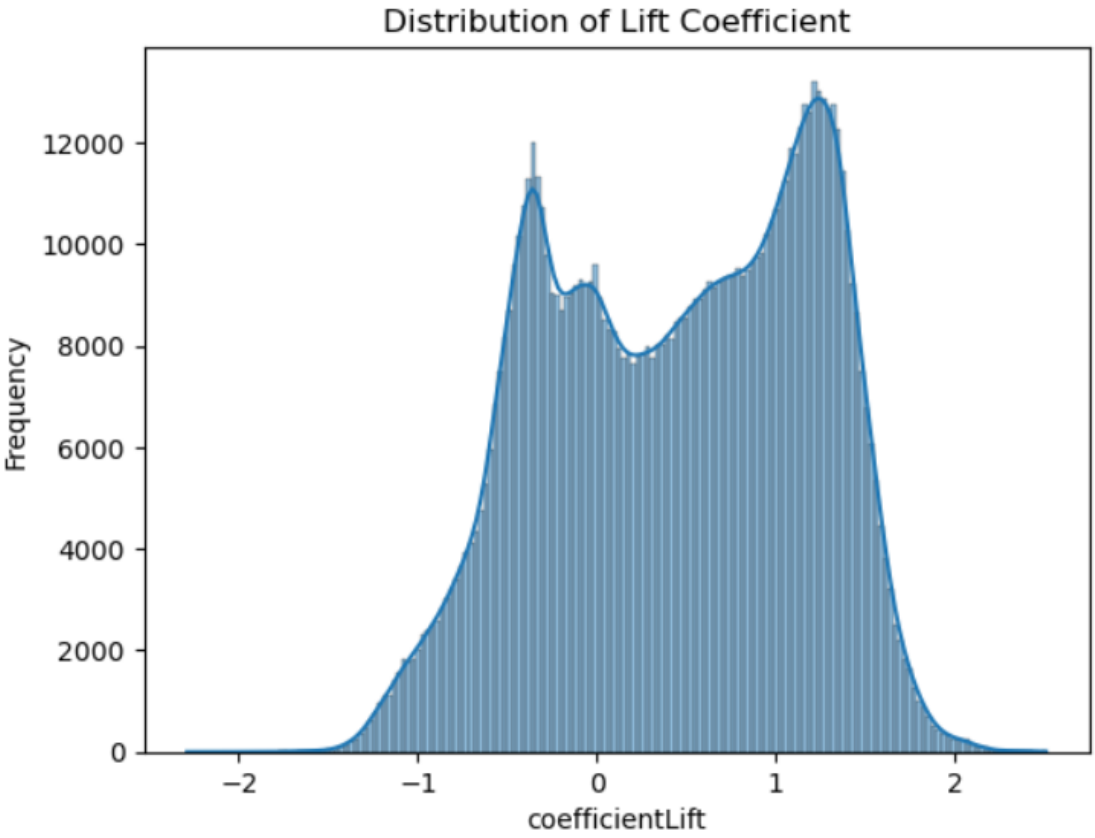
- Use MultiOutputRegressor for multi-target prediction.

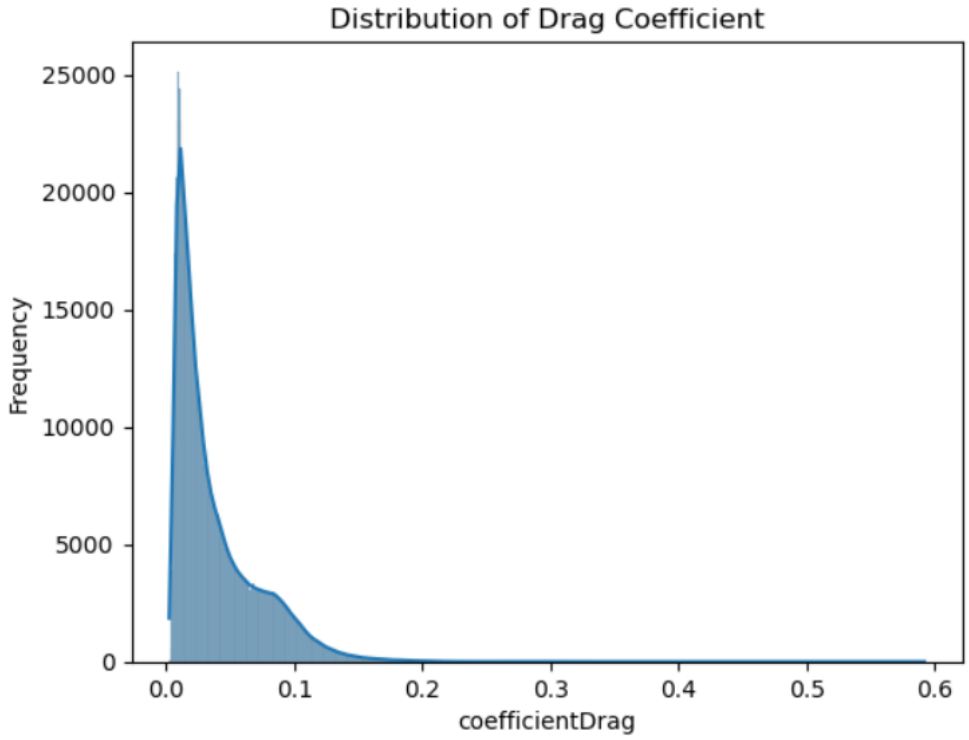
- Optimize max_depth, max_features, and other hyperparameters.

- Try lighter models like GradientBoostingRegressor or XGBoost for better performance with limited RAM.


Actual vs Predicted Lift Coefficient
Predicted Lift Coefficient
Actual Lift Coefficient
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5


Actual vs Predicted Drag Coefficient
Predicted Drag Coefficient
Actual Drag Coefficient
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200


Distribution of Lift Coefficient
Frequency
12000
10000
8000
6000
4000
2000
0
−2
−1
0
1
2
coefficientLift


Distribution of Drag Coefficient
Frequency
25000
20000
15000
10000
5000
0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
coefficientDrag