Machine Learning driven airfoil generator

A Python & ML pipeline for CST based airfoil design

Project context

Problem:

- Public libraries of 2D airfoils are limited in size, diversity, and reliability for building solid **Machine Learning** ready databases for aerodynamic design.

- Classical **NACA** series provide deterministic shapes but limited variability.

- Large-scale airfoil generation methods like **CST** (Class Shape Transformation) offer more flexibility but can produce unphysical shapes if not properly validated.

Project scope:

- Create a **Python** based pipeline that combines CST geometry and machine learning to automatically generate, classify, and validate airfoils enabling the creation of large, **XFOIL** ready datasets for ML-driven aerodynamic design.

- Next slides will show the main topics and steps of the project

CST Theory

- **What is CST:**

  - **C**lass **S**hape **T**ransformation defines airfoil as a combination of:
    - A **class function *C(x)*** → governs general edge behavior (leading and trailing edge).
    - A **shape function *S(x)*** → provides local geometry control via Bernstein polynomials of order *n.*

$$y(x) = C(x) \cdot S(x) = x^{N_1}(1-x)^{N_2} \cdot \sum_{i=0}^{n} A_i \cdot \binom{n}{i} x^i (1-x)^{n-i}$$

- **Why is it powerful for ML application:**

  - Easy to generate thousands of variations by sampling coefficient values ($A_0 \dots A_n$).
  - Enable smooth, watertight and continuous airfoil geometry.
  - Independent control over upper and lower surfaces.

- **Degree of Freedom:**

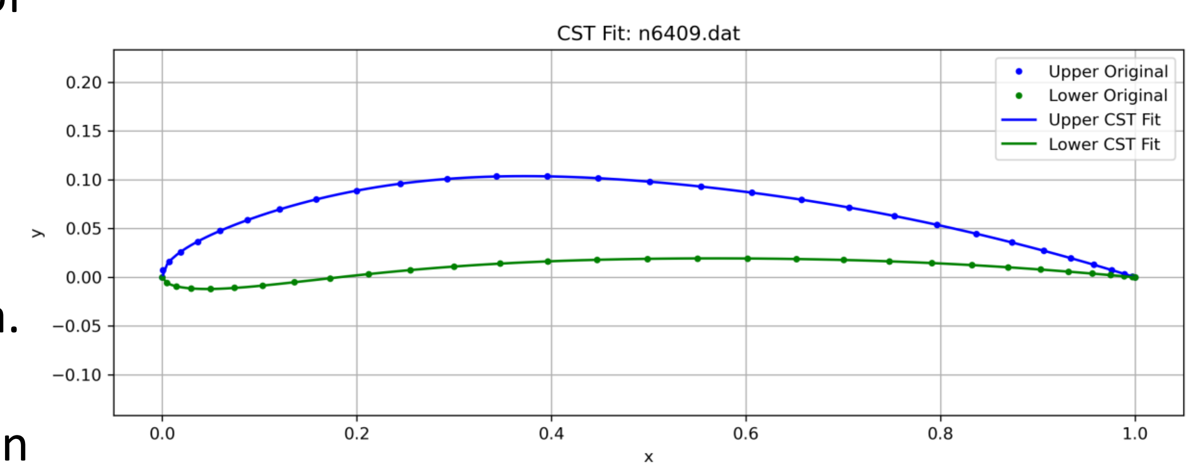
  - Each coefficient adjust local curvature (*number of coefficients = polynomial order + 1*).
  - The higher the order of Bernstein polynomial the greater geometry flexibility but also more noise.

- **Coefficient fitting & generation stability:**

  - To ensure the stability of random airfoil generation, it is essential to begin from a set of well-fitted, physically meaningful CST coefficients.

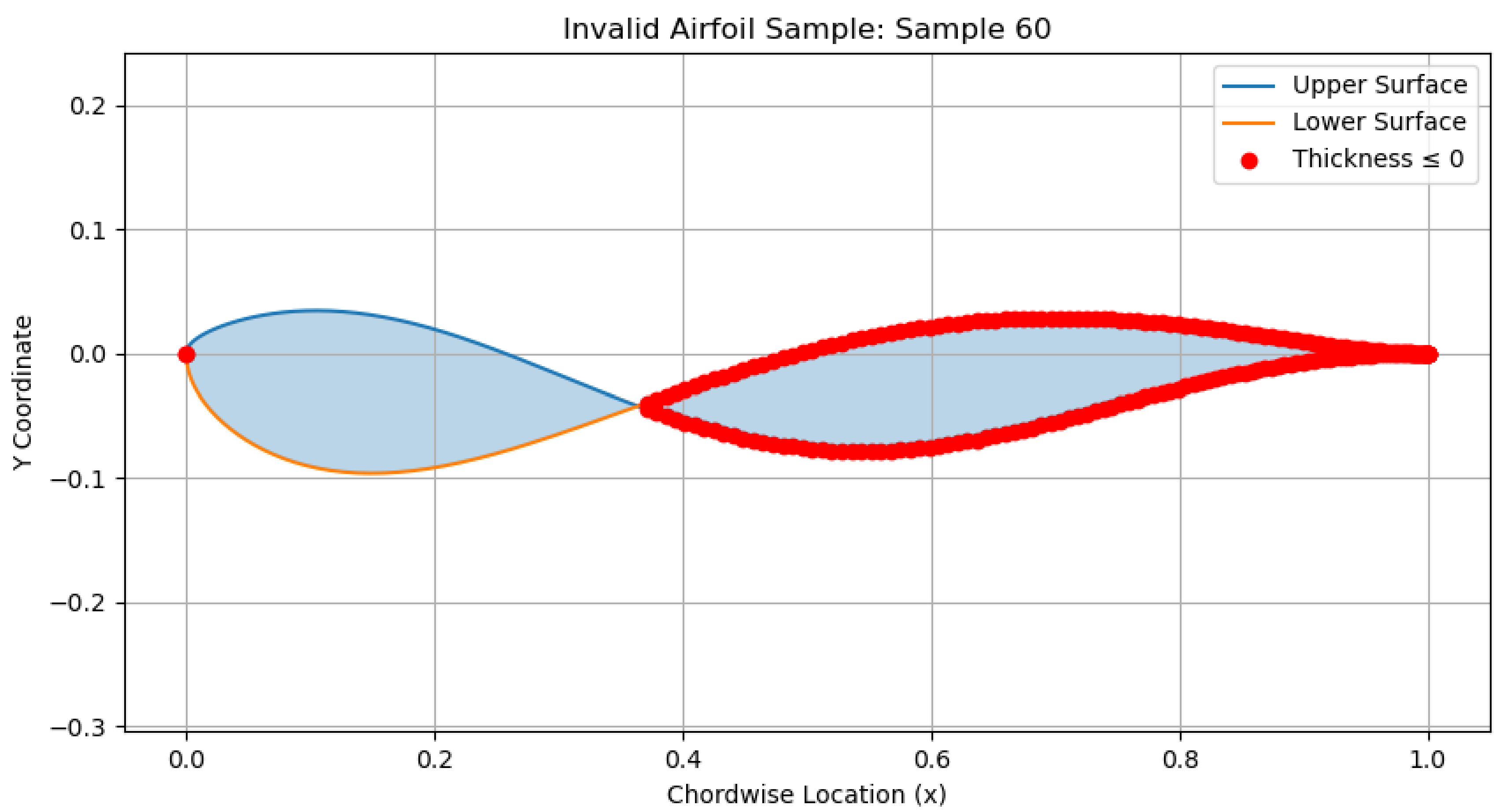
  - The script ***00_CST_NACA_coefficientFit.py*** extracts such coefficients from real airfoil data.

  - These fitted references serve as the foundation for modeling coefficient distributions and avoiding invalid geometries during large-scale sampling.



Dataset preparation

- **Machine Learning robust dataset:**
  - **CST** provides wide freedom to generate thousands of airfoils but not all are physically meaningful.
  - A machine learning model requires clear, labeled examples of both valid and invalid shapes to determine good or bad shapes.
  - Geometric features must be extracted consistently to enable pattern recognition.
  - Labeling is critical: supervised learning depends on high-quality input/output pairs.
- **Training set formulation for supervised learning:**
  - Valid profiles are obtained by fitting real airfoils from **Selig database.**
  - Synthetic invalid profiles are generated by sampling CST coefficients beyond realistic bounds.
  - Labeled as valid/invalid and transformed into feature vectors (thickness, camber, curvature, etc.).
- **Input for Machine Learning classifier:**
  - These labeled feature vectors serve as input for a machine learning classifier, enabling automated detection of valid vs invalid airfoil geometries based on shape characteristics and coefficient values.



Example of bad airfoil

Machine Learning training

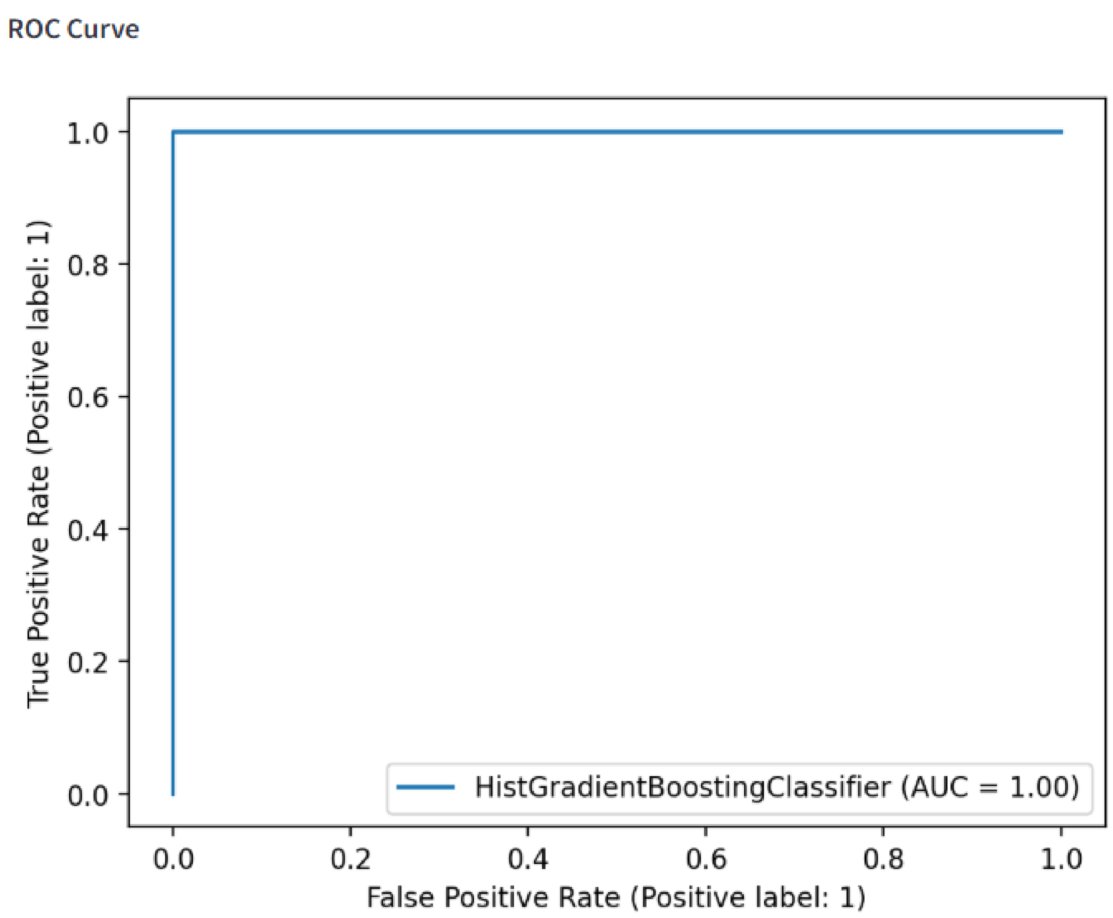
- **Machine Learning models used:**

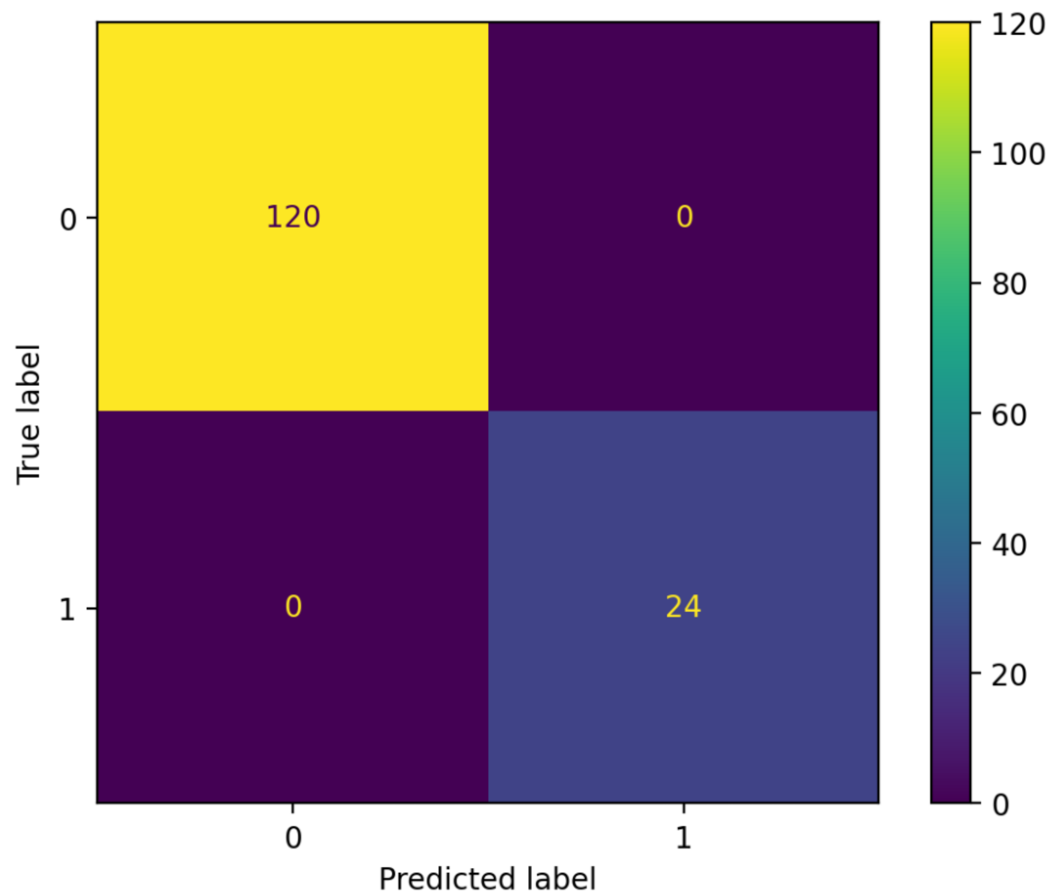
  - **Random Forest (RF):** Combines the predictions of many decision trees trained on slightly different data. It reduces overfitting and is easy to interpret.
  - **Gradient Boosting (GB)**: Builds decision trees one after another, where each tree tries to correct the mistakes of the previous one. It often achieves higher accuracy but needs careful tuning.

- **Performance metrics of ML models:**

  - **ROC Curve**: Measures classification confidence across thresholds (AUC ≈ 1 = excellent separation)
  - **Confusion Matrix**: Shows prediction results (true/false positives and negatives)

- Interactive model comparison is available via the Streamlit app **Streamlit** app *02_04_Streamlit_ML_comparison.py*





The ROC curve illustrates the classifier's ability to separate good and bad airfoil geometries with high confidence

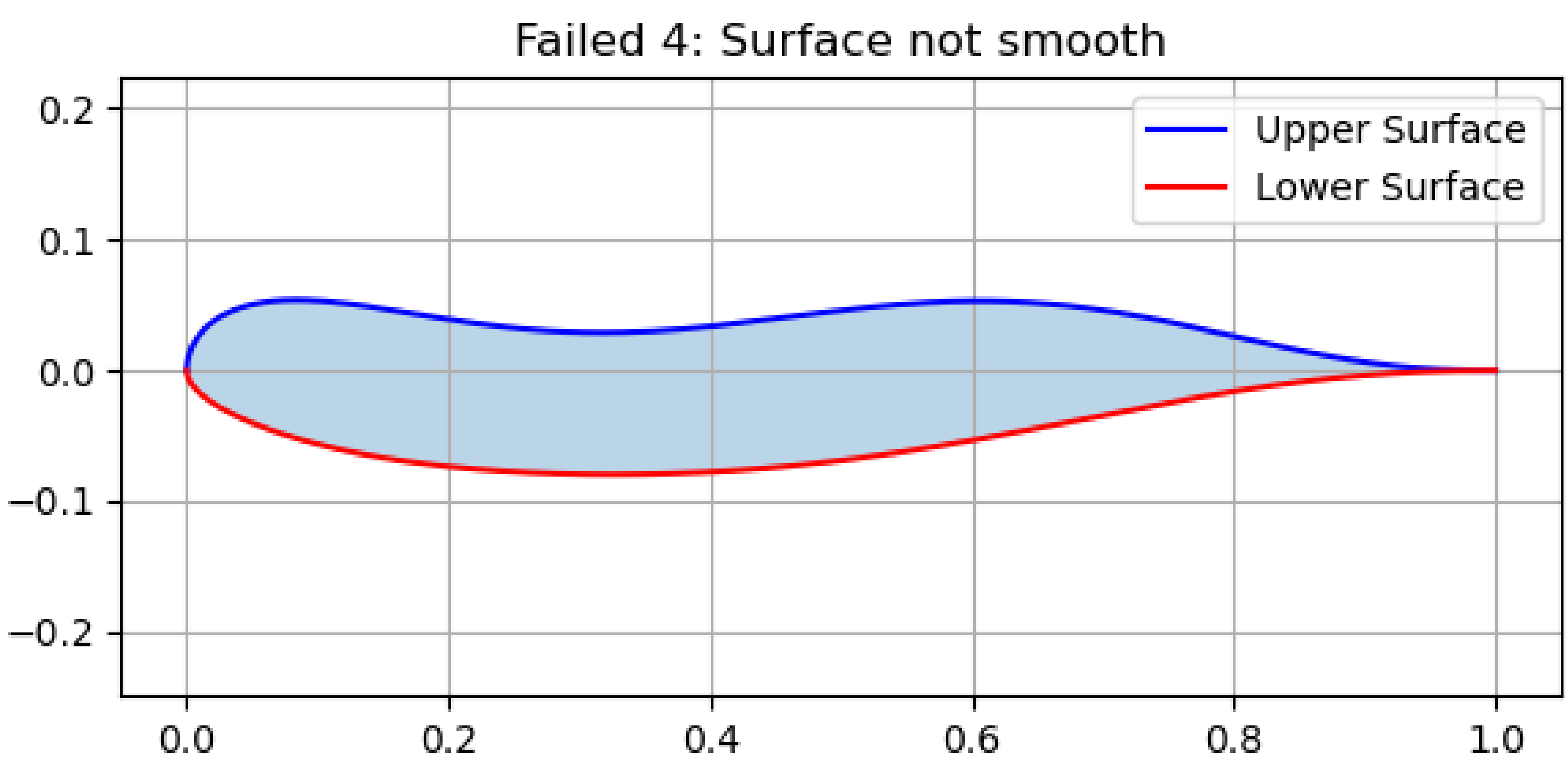
Confusion matrix shows high accuracy in distinguishing valid vs invalid CST airfoils.

Final geometry validation

- **From CST coefficient to real geometry:**
  - The ML model was trained to classify CST coefficient combinations.
  - However, valid coefficients do not guarantee **XFOIL** compatible airfoils.
  - Even plausible combinations can yield problematic shapes (e.g. overlapping surfaces, sharp gaps).

- **Final geometry checks:**
  - After reconstructing airfoil geometry, deterministic filters were applied to detect:
    - Surfaces discontinuities and smoothness.
    - Open trailing edge.
    - Crossing or self intersecting surfaces.
  - Only geometries passing all checks were saved to be used afterwards in XFOIL.

- **XFOIL compatibility:**
  - XFOIL is a well-known panel method for low-speed airfoil analysis.
  - Invalid shapes cause solver divergence or misleading results.
  - Misleading results degrade pipeline reliability and compromise downstream **ML** accuracy for preliminary aerodynamic design.



What’s next

- **Current pipeline outcome:**

  - The current pipeline generates a large set of physically valid airfoils**,** ready to be analyzed with **XFOIL**.
  - These profiles are verified for geometric integrity and are saved in *.dat* format for future aerodynamic simulations.

- **Next steps:**

  - The next step is to simulate across a grid of angles of attack (AoA) and Reynolds regime these validated airfoils in XFOIL to extract:
    - Lift coefficient (CL).
    - Drag coefficient (CD).
    - Moment coefficient (CM).

- **ML powered design vision:**

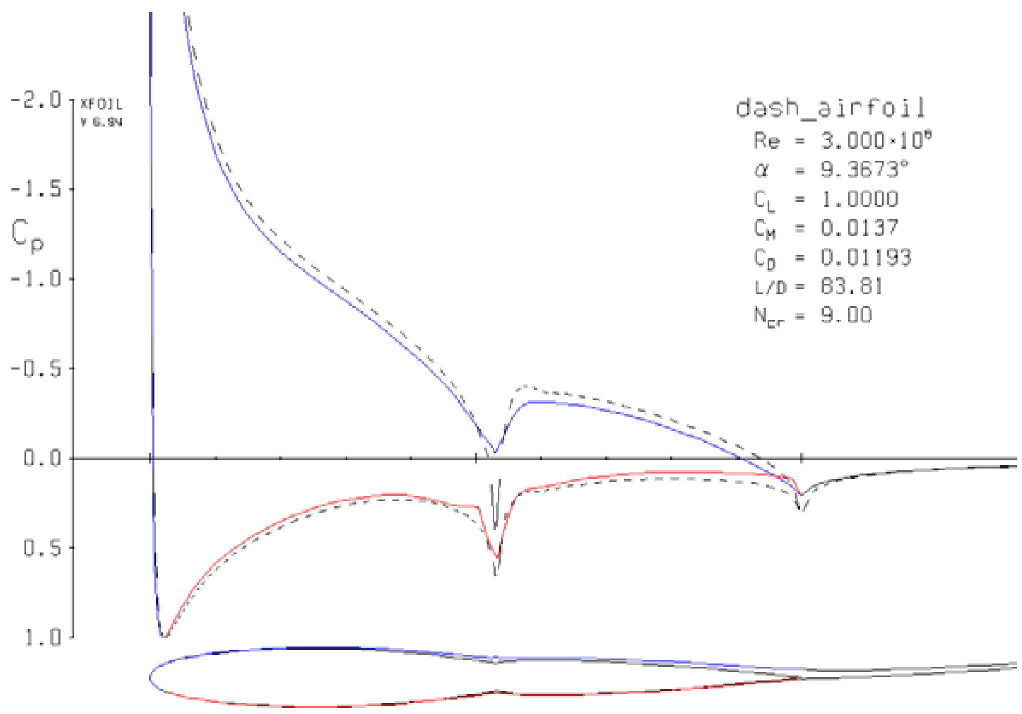
  - These simulation results will serve as **training data** for **regression models** (e.g. GB, RF, neural networks).
  - Goal: Predict aerodynamic coefficients from airfoil shape, AoA, and Re in **real-time.**

- **Ultimate objective:**

  - Enable **early-stage airfoil selection.**
  - Minimize simulation dependency.
  - Support fast informed design decisions.

DOWNLOAD THE GITHUB REPOSITORY MENTIONED IN THE COMMENT AND GIVE IT A TRY!

GitHub



Example of pressure coefficient distribution from XFOIL simulation.