Homework 5: Build The Best ML Model

Due Date: Wednesday, December 3rd, 2025, 4:00PM

This is the last homework of 157/272A. Well done!

Introduction

This final homework will be a comprehensive assignment that integrates everything you have done in your previous homework assignments.

In Homework 1 and 2, you learned how to build Machine Learning models using simple datasets. In this process, you explored not only how to write prompts when using LLMs to write code (Prompt Engineering), but also how to review generated code.

Homework 3 used a dataset where you had to perform Feature Engineering yourself. In this process, you experienced how to write prompts to LLMs when writing long and complex code, and how to provide context when writing long code step by step, which differs from writing a single short piece of code.

Homework 4, we introduced Cline. Unlike before when you had to collaborate with LLMs for prompt writing, code execution, code debugging, etc., using Cline allows you to complete code using only natural language. This is because Cline helps with all aspects including plan generation, code generation, execution, and debugging.

Once you complete this homework, you will have accumulated multiple experiences of Machine Learning model building as well as experience writing ML code from start to finish using Prompt Engineering and Cline.

In this homework, you will build a Machine Learning model using the WM-811K dataset that you used in Homework 3. What makes this homework different from Homework 3 is that this homework has no template or storyline, and unlike Homework 3 where you just needed to exceed the test accuracy threshold, in this homework your test accuracy is your final score!

Tasks

The dataset you will use this time is completely identical to the dataset used in Homework 3. Based on the code you wrote in Homework 3, try various approaches to improve test accuracy. You will submit a CSV file containing predictions for the test dataset and a `README.md` written by Cline.

The CSV file, just like in Homework 3, is a file containing your final model's failure type predictions. The CSV must have the column name (`failureType`) in the first row. There is no limit to the number of submissions, but you cannot check your test accuracy until after the due date. For reference, the best accuracy was 99.35 (last year) and 95.31 (this year's Homework 3).

You may use any LLM or Agent to complete your code. For example, you can use Cline, Codex, Claude Code, as well as ChatGPT, Gemini, Claude, etc. However, after completing the code, you must use Cline to write the `README.md`.

Follow the process below to create the optimal Machine Learning model!

Data Preparation (Part 1)

- Wafer maps have different shapes. Resize all wafer maps to (64, 64).
- Convert the failure type string to numerical value.

Feature Engineering

Create features for the wafer maps. At this time, make sure to verify that the functions you created in Homework 3 work according to the feature descriptions. Also, you can try various approaches to increase accuracy. For example, are all the features below equally important? You can also try removing unimportant features. To do this, check which combinations of features can distinguish each failure type. For example, the Edge-loc type will have high Edge Yield Loss, a Major Axis Ratio of 1 or greater, a Minor Axis Ratio of 0.1 or less, and a large Min Distance from Center. You can do this directly based on feature descriptions, or you can figure it out through Decision Tree plots from Homework 3. Or you can also add new features. You can have Cline or an LLM recommend features by explaining the dataset characteristics and task. Wafer map pattern classification and the WM-811K dataset are relatively well-known.

- Area Ratio: Ratio of the area of the salient region to the area of the wafer map
- Perimeter Ratio: Ratio of the perimeter of the salient region to the radius of the wafer map
- Max Distance from Center: Maximal distance between the salient region and the center of the wafer map
- Min Distance from Center: Minimal distance between the salient region and the center of the wafer map
- Major Axis Ratio: Ratio of the length of the major axis of the estimated ellipse surrounding the salient region to the radius of the wafer map
- Minor Axis Ratio: The ratio of the length of the minor axis of the estimated ellipse surrounding the salient region to the radius of the wafer map
- Solidity: The proportion of failed dice in the estimated convex hull in the salient region
- Eccentricity: The shape of the estimated ellipse surrounding the salient region, where the value is 0 for a circle, or 1 for a line
- Yield Loss: Ratio of the failed dice on the wafer map to the total number of dice on the wafer map
- Edge Yield Loss: Ratio between the number of failing dice within two pixels of a wafer edge to the total number of dice within two pixels of a wafer edge.

Data Preparation (Part 2)

Select the features and prepare the dataset for training and validation.

Model Training and Validation

Select a model, train the model, and use training accuracy and validation accuracy to find the optimal model and hyperparameters. In Homework 3, you used Decision Tree and SVC, but in this homework you can choose any model. Explain the dataset to Cline, or directly to an LLM, and think together about which model would be good to use. Don't forget to adjust hyperparameters after selecting your model.

Model Testing

Once you have completed all selections and tuning, load the test dataset, preprocess it, predict the failure type, and output prediction to a `scores.csv` with a single column, `failureType`.

README.md

Even if you didn't use Cline to get to this point, you must use Cline in this step. Create a markdown file called `README.md`. This should include your data preparation process, feature engineering (including which features you used), model and hyperparameter choice, and training and validation accuracy of your model. Ask Cline to read the folder containing all the code you used and create a `README.md` with this content.

What to turn in

- `scores.csv`: A CSV file with one column, failureType, containing your model's prediction for the wafers in wafermap_testing.npy.
- `README.md`: Markdown file made by Cline that includes a description about your data preparation process, feature engineering (including which features you used), model and hyperparameter choice, and training and validation accuracy of your model.