Qorvo

Semiconductor Manufacturer
During the manufacturing of semiconductor wafers Qorvo tests many individual circuits dozens of times to ensure quality. This creates a lot of data that is hard to analyze, they need a machine learning pipeline to identify patterns and anomalies in this dataset.

Solution

Pre-processing
In the pipeline there are preprocessing options allowing the users to define whether they want to impute missing values, correlated tests, or even aggregate test fails into a single column
Model
After preprocessing the data is given to a VAE. The VAE, a variational autoencoder, begins by training on the data in "reverse" taking images, compressing them down, reconstructing the original image from the compressed data and comparing that to the original image, creating a loss value. This is repeated until the loss is at an acceptable level. The model is then run normally on the data, creating a similarity matrix by comparing the compressed data in the latent space.
Clustering
This similarity matrix is then translated to a distance equivalent:
$Dm = sqrt(2 * (1 - Sm))$
The distance matrix is then passed to either a hierarchical or k-means clustering algorithm depending on whether the user wanted a specific or dynamic number of groups. This creates groups of wafers based on the similarity created by the VAE.

Who are We?

We are a group of students at Oregon State University - Cascades in our senior capstone program. We were provided a number of projects proposed by local companies to work on as pseudo interns to get work experience. We ended up on this project for Qorvo. We found the data tough to start working with having vague labels, dozens of columns, and tens of thousands of rows. Once we understood the data we were able to start looking at models to use, it took a couple months, but eventually we found what we were looking for. With this we split into two groups, with one person working on fine tuning the model and understanding it, and the other two people creating pre and post processing steps.

- The technical details of your implementation. What is the tech stack? How does it all work? (Be brief.)

Environment

Databricks

The entire project in contained within databricks in a series of jupyter notebooks

Notebooks

We have two primary folders of jupyter notebooks in our directory, one where we did all of our development, and one where we run the pipeline on multiple datasets using notebooks for each with documentation for every step. Within this folder there is another that holds all the notebooks for the pipelines source code, containing files for every step with pydocs for every function.

Current Status

The pipeline has been run on multiple different products, producing groups as expected.These pipeline runs are in descriptive jupyter notebooks, acting as READMEs for future work.

Next Steps

First thing to do is get the project under version control. The clients already have a GitLab workflow that they are moving the project into. Further steps include:

- Aggregate Tests
    - With aggregate tests, we used them for some training runs and compared them to outputted wafer maps. We also thought it may be beneficial to weigh the test failures even more by using it as a main feature in the training dataset.
- Feature reduction
    - We decided to try and reduce the feature set to a more manageable size by averaging correlated test columns. There could be tuning done here to figure out the best representative feature set, like PCA, or simply chucking more data into it.
- Cross Product comparison
    - It could be helpful to compare outputs from different products, or contain multiple products in a training set. As different products are not the same size, padding or rescaling may be required. The other option may be to try and assign names to clusters, such as classifying clusters as high null, or ring patterned, or so on. This could help in comparing relative size of clusters in a dataset.

- Clustering
    - We tried a multitude of clustering algorithms, at different stages in the process. We settled for KMeans for most things, but are using a force based network graph to create the final cluster visualization from the similarity score. There may be more rigorous options for clustering.
- Continued Classification
    - Once you have model weights, it is possible to ask the model to classify a single wafer map compared to the ones it has been trained on. We have not explored this area deeply.
    - The model only handles the whole dataset provided to it right now, however it would be useful to introduce new data to an already trained model, returning a new matrix with this new data.

Deployment

Notebooks

A user is able to make a copy of one of the existing pipeline notebooks, change the database to the data they want to run the pipeline on, adjust settings on the steps used, and adjust the input size for the model. The user could also create their own from scratch and use the functions provided whoever they which, skipping preprocessing steps as they wish.