**Project Description**

This project is from semiconductor wafer fabrication industry. Wafer manufacturing happens through several machines in fab and this process is monitored through sensors in machines. After complete processing, wafers will send for electrical testing to sort out Good materials.

This project is about building a machine learning classification methodology to predict the quality of semiconductor wafers based on given training data.

**Architecture**

The complete architecture for this project is defined as below.



**Data Description (Training)**

The client will send data in multiple sets of files in batches at a given location. This data will contain Wafer names and 590 columns of different sensor values for each wafer which are the inputs. The last column will have the "Good/Bad" value for each wafer which is the target.

"Good/Bad" column will have two unique values +1 and -1. "+1" represents Bad wafer and "-1" represents Good Wafer.

Apart from training files, client will also have a "schema" file, which contains all relevant information about training files such as: Name of files, Length of Date value in FileName, Length of Time value in FileName, Number of Columns, Name of Columns and their datatype.

**Data Validation (Training)**

Various validation activities has performed on the training files.

1. File Name Validation : validate name of files based on the schema file. This is done by creating a regex pattern as per the name given in the schema file. After validating pattern in the name, check for other details like length of date in the file name as well as length of time in the file name. If all the values are as per requirement, move such files to "Good\_Data\_Folder" else move such files to "Bad\_Data\_Folder."
2. Number of Columns : validate number of columns present in the files, and if it doesn't match with value given in the schema file, then that file is moved to "Bad\_Data\_Folder."
3. Name of Columns : The name of columns is validated and should be same as given in the schema file. If not, then that file is moved to "Bad\_Data\_Folder".
4. Datatype of columns : The datatype of columns is given in the schema file. This is validated when files insert into Database. If datatype is wrong, then its moved to "Bad\_Data\_Folder".
5. Null values in columns : If any of the columns in the file have all values as NULL or missing, then discard such file and moved to "Bad\_Data\_Folder".

**Data Insertion in Database (Training)**

It consists of below steps:

1. Database Creation and connection : Create a database with the given name passed. If the database is already created, open the connection to the database.
2. Table creation in database : Table with name "Good\_Data" is created in the database for inserting files in "Good\_Data\_Folder" based on given column names and datatype in schema file. If the table is already present, then new table is not created and new files are inserted in the already present table since training can be done on new as well as old training files.
3. Insertion of files in table : All files in the "Good\_Data\_Folder" are inserted in above-created table. If any file has invalid data type in any of the columns, then that file is not loaded in table and will be moved to "Bad\_Data\_Folder".

**Model Training**

Below activities done as part of model training:

1. Data Export from Db : The data in training database is exported as a CSV file to be used for model training.
2. Data Preprocessing : comprises following

a) If null values present in columns, then impute them using KNN imputer.

b) If any column has zero std deviation, remove them as those are not useful features.

1. Clustering : KMeans algorithm is used to create clusters in the preprocessed data. The optimum number of clusters is selected by plotting elbow plot and dynamic selection of number of clusters done by "KneeLocator" function. The idea behind clustering is to implement different algorithms to train data whcich are in different clusters. The Kmeans model is trained over preprocessed data and that model is saved for further use in prediction.
2. Model Selection : Two algorithms, "Random Forest" and "XGBoost" are used to find best model for each cluster. For each cluster, both algorithms are passed with best parameters which derived from GridSearch CV. AUC scores for both models has calculated and selected the model with best score. This process has repeated for each cluster and all such models are saved for use in prediction.

**Data Description (Prediction)**

Client will send data in multiple set of files in batches at a given location together with “schema” file. Data will contain Wafer names and 590 columns of different sensor values for each wafer.

**Data Validation & Data Insertion in Database (Prediction)**

Similar to Training steps as detailed above

**Prediction**

Below activities done as part of model prediction:

1. Data Export from Db : The data in prediction database is exported as CSV file to be used for prediction.
2. Data Preprocessing : similar to model training
3. Clustering : KMeans model created during training is loaded and clusters for prediction data is predicted.
4. Prediction : Based on the cluster number, respective model (either RF or XGBoost) is loaded and is used to predict the data for that cluster.
5. Once prediction is made for all clusters, the predictions along with Wafer names are saved in a CSV file at a given location and the location is returned to the client.

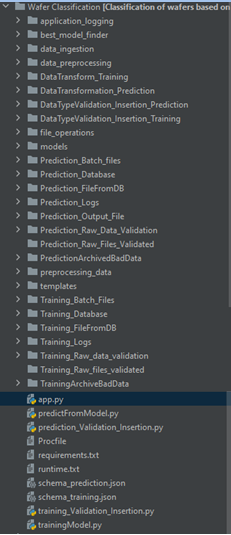
**Project folder structure :**

**requirements.txt** consists of all packages that needed to deploy the app in the cloud.

**app.py** is entry point of application, where flask server starts.

**runtime.txt** contains python version.

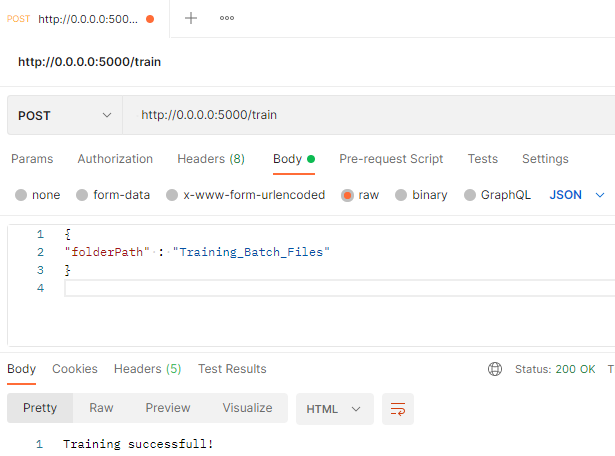
**Procfile** shows entry point of the app.

****

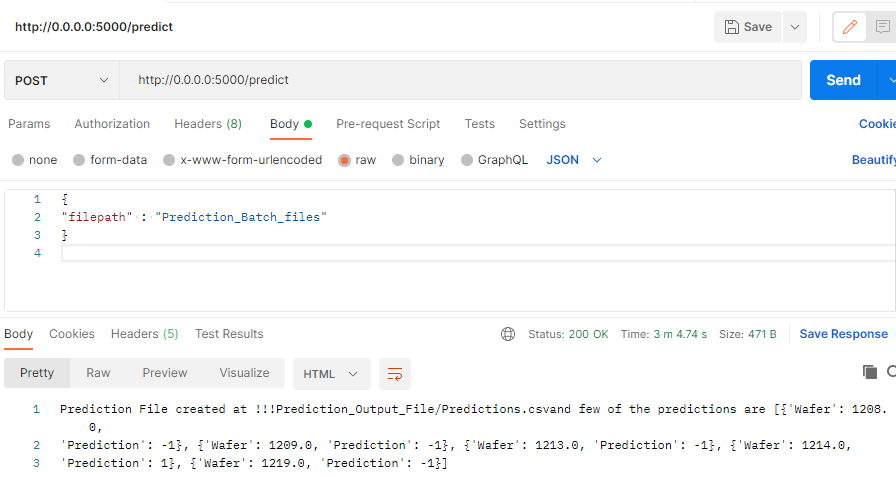
**Application Testing**

Project testing has done using Postman.

**Training :**



**Prediction :**



**Cloud Deployment**

The model has deployed in Heroku Cloud platform.

