**DETAIL PROJECT REPORT (DPR)**

**Wafer Sensor Project**

**Problem Statement**

To build a machine learning model that classifies whether a wafer is in good working condition or not based on sensor data

**Objective**

Develop a predictive model for detecting the situation of wafer whether it is in good working condition or not

**Benefits**

* Automate the process of determining the conditions of the wafers
* Helps in easy flow for managing resources
* It safes time as it easily identifies which wafer is good/bad without shutting the entire production line
* Helps in increasing working hours which in-turn increase revenue

**Data Sharing Agreement and Data Description**

A non-disclosure Data Sharing Agreement (DSA) is established with the client. The client will send data in multiple sets of files in batches at a given location. Data will contain Wafer names and 590 columns of different sensor values for each wafer. The last column will have the "Good/Bad" value for each wafer.

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

"+1" represents Bad wafer.

"-1" represents Good Wafer.

Apart from training files, we also require a "schema" file from the client, which contains all the relevant information about the training files and it consist of the following

* Sample file name
* Length of date stamp(8)
* Length of time stamp(6)
* Number of columns
* Columns names
* Column data type

**Project Architecture**



**Data Validation and Data Transformation**

* In this step, we perform different sets of validation on the given set of training files.

Name Validation- We validate the name of the files based on the given name in the schema file. We have created a regex pattern as per the name given in the schema file to use for validation. After validating the pattern in the name, we check for the length of date in the file name as well as the length of time in the file name. If all the values are as per requirement, we move such files to "Good\_Data\_Folder" else we move such files to "Bad\_Data\_Folder."

* Number of Columns - We validate the number of columns present in the files, and if it doesn't match with the value given in the schema file, then the file is moved to "Bad\_Data\_Folder."
* Name of Columns - The name of the columns is validated and should be the same as given in the schema file. If not, then the file is moved to "Bad\_Data\_Folder".
* The data type of columns - The data type of columns is given in the schema file. This is validated when we insert the files into Database. If the data type is wrong, then the file is moved to "Bad\_Data\_Folder".
* Null values in columns - If any of the columns in a file have all the values as NULL or missing, we discard such a file and move it to "Bad\_Data\_Folder".

**Data Transformation**

* Data Export from Db - The data in a stored database is exported as a CSV file to be used for model training.
* Data Preprocessing
* Check for null values in the columns. If present, impute the null values using the KNN imputer.
* Check if any column has zero standard deviation, remove such columns as they don't give any information during model training.
* Clustering - KMeans algorithm is used to create clusters in the preprocessed data. The optimum number of clusters is selected by plotting the elbow plot, and for the dynamic selection of the number of clusters, we are using "KneeLocator" function. The idea behind clustering is to implement different algorithms

To train data in different clusters. The Kmeans model is trained over preprocessed data and the model is saved for further use in prediction.

**Data Insertion into the Database**

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

**Model Selection and Training**

After clusters are created, we find the best model for each cluster. We are using three algorithms, "SVM", "Random Forest" and "XGBoost". For each cluster, both the algorithms are passed with the best parameters derived from GridSearch. We calculate the AUC scores for both models and select the model with the best score. Similarly, the model is selected for each cluster. All the models for every cluster are saved for use in prediction

**Prediction Data Description**

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

Apart from prediction files, we also require a "schema" file from client which contains all the relevant information about the training files such as:

Name of the files, Length of Date value in File Name, Length of Time value in File Name, Number of Columns, Name of the Columns and their data type.

**Data Validation**

In this step, we perform different sets of validation on the given set of training files.

* Name Validation- We validate the name of the files on the basis of given Name in the schema file. We have created a regex pattern as per the name given in schema file, to use for validation. After validating the pattern in the name, we check for 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, we move such files to "Good\_Data\_Folder" else we move such files to "Bad\_Data\_Folder".
* Number of Columns - We validate the number of columns present in the files, if it doesn't match with the value given in the schema file then the file is moved to "Bad\_Data\_Folder".
* Name of Columns - The name of the columns is validated and should be same as given in the schema file. If not, then the file is moved to "Bad\_Data\_Folder".
* Data type of columns - The data type of columns is given in the schema file. This is validated when we insert the files into Database. If data type is wrong then the file is moved to "Bad\_Data\_Folder".
* Null values in columns - If any of the columns in a file has all the values as NULL or missing, we discard such file and move it to "Bad\_Data\_Folder".

**Data Insertion in Database**

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

**Prediction**

* Data Export from Db - The data in the stored database is exported as a CSV file to be used for prediction.
* Data Preprocessing
* Check for null values in the columns. If present, impute the null values using the KNN imputer.
* Check if any column has zero standard deviation, remove such columns as we did in training.
* Clustering - KMeans model created during training is loaded, and clusters for the preprocessed prediction data is predicted.
* Prediction - Based on the cluster number, the respective model is loaded and is used to predict the data for that cluster.
* Once the prediction is made for all the clusters, the predictions along with the Wafer names are saved in a CSV file at a given location and the location is returned to the client.

**Deployment**

The model was deployed on Amazon Web Service (AWS)

**Question and Answers**

**Q1. Tell me about your project.**

A client has a production line in which different wafer sensors are deployed. Whenever there is a fault in any of the wafer, they have to manually check all the wafers to verify which of the wafer is faulty which leads to stopping the entire production line for the wafer to be replaced.

By using different technological stack, the client has asked us to create a model which successfully identify which wafer is faulty which automatically prevent the entire production line from stoppage

**Q2. What is the source of the data?**

The data for training is provided by the client in multiple batches and each batch contain multiple files

**Q3. After the file validation, what did you do with the incompatible file or file that didn’t pass the validation?**

Files that didn’t satisfy the non-disclosure agreement signed with the client will be moved to the **Achieve folder** and a list of these files has been shared with the client and is moved to the bad data folder.

**Q5. How logs are managed?**

We are using different logs as per the steps that validation phase and modeling like file validation log, data insertion, model training log, prediction log etc

**Q6. What are the techniques used for data preprocessing?**

* Removing unwanted attributes
* Visualizing the relationship between independent variables with each other and output variables
* Removing columns with zero standard deviation
* Cleaning data and imputing of null values where necessary
* Converting categorical data into numeric values
* Removal of outliers
* Scaling the data

**Q7. How training was done or what model was used?**

Single model was not used on the entire training data. We found out implicit patterns in the data and the training data was divided into different clusters/group. Clusters are then added to the training data. We looped over these clusters to segregate the data.

We then prepared individual model for individual clusters and see which particular model performs well. Algorithms like SVM, XGBoost, and Random Forest were used and based on AUC, final model was used for each cluster and we saved the model

**Q8. How was prediction done?**

The testing file was shared by the client. We performed the same life cycle till the data is clustered. On the basis of cluster number model is loaded and perform prediction. In the end we get the accumulated data of predictions

**Q9. What was the data type?**

The data used in the training of this model is a combination of both numerical and categorical values

**Q10. What was the team size and distribution?**

The team consist of people with different technological skills and disciples. The team consisted of

* 1 Project manager
* 1 Technical Architect
* 1 Project Lead
* 3 Data Scientist
* 2 Dev-Ops Engineers
* 2 UIUX/Analytic developer

**Q11. How were you maintaining failure cases?**

Suppose the model is not able to make correct prediction for a wafer sensor, then the data get stored in the database. There will be a report triggered to the support team with all the failure scenarios where they can inspect the cause of failure. Once we have sufficient number of sensor data, we can label and include those data while retraining the model for better model performance.

**Q12. What is the kind of automation for data processing?**

A full-fledged ETL pipeline is in place for data extraction processing and loading into the database

**Q13. Have you used any scheduler?**

Yes!

A scheduler was used to retrain the model after a combine period of twenty days.

**Q14. How are you monitoring job?**

There are logging set-ups done. We regularly monitor the logs to see for any error scenarios. For fatal errors, we had email notifications in place. Whenever a specific error code, which has been classified as a fatal error occurs, email gets triggered to the concerned parties.

**Q15. What were your roles and responsibilities in the project?**

I participated in both core and non-core data science area particularly the data validation, transformation, model selection and dash boarding

**Q16. What was your day to day task?**

My day to day tasks involved completing the task assigned to me by the project manager. Some of these task include

* Attending the scrum meetings,
* Participating in design discussions and requirement gathering,
* Doing the requirement analysis,
* Data validation,
* Unit test for the models

**Q17. In which area you have contributed the most?**

I contributed the most to Data transformation and model training areas. Also, we did a lot of brainstorming for finding and selecting the best algorithms for our use cases. After that, we identified and finalized the best practices for implementation, scalable deployment of the model, and best practices for seamless deployments as well

**Q18.In which technology you are most comfortable?**

I have worked across diverse areas in the industry using different technological stack ranging from Machine Learning, Deep Learning and NLP. With my experience and exposure, I love working in Machine Learning and Deep Learning

**Q19.How you rate yourself in big data technology?**

Being someone that is hungry for growth across the tech industry, Big data is the area I am working seriously on to become proficient. I’m currently learning SPARK

**Q20. In how many projects you have already worked?**

Giving a precise figure will be difficult but I have worked in various small and large scale projects e.g Deep learning, Computer Vision, NLP projects, Chatbot building, Machine learning with regression, and classification problems.

**Q21. How would you rate yourself in distributed computation?**

I will rate myself 7

**Q22. In which part of machine learning have you already worked on?**

I have worked on both supervised and unsupervised machine learning approaches and building different models using the as per the requirement of the client/user

**Q23. What are the areas of machine learning algorithms that you already have explored?**

I have explored various machine learning algorithms like Linear Regression, Logistic Regression, L1 and L2 Regression, Polynomial Regression, Multi Linear Regression, Decision Trees, Random Forests, Extra Trees Classifier, PCA, XGBoost, CAT Boost, ADA Boost, Gradient Boosting, Light Boost, K-Means, K-Means ++, LDA, QDA, KNN, SVM, SVR, Naïve Bayes, Agglomerative clustering, DBSCAN, Hierarchical clustering and so on

**Q24. How much time did your model take to get trained?**

With a training batch size of 41 and a laptop with system configuration of 64GB RAM using NVIDIA Pascal Titan GPU, the entire training took 45 minutes

**Q25. At what frequency are you retraining and updating your model?**

The model gets retrained every 20 days.

**Q26. How would you rate yourself in machine learning?**

On the scale of 10, I will rate myself 8

**Q27. Why are you leaving your current organization?**

Being in the organization was a blessing but what we do is becoming monotonous and boring. The quest to explore and seek for knowledge, skills and a fresh challenge is the reason behind my job search

**Q28. What kind of challenges have you faced in your project?**

The biggest challenges I faced came during the data preprocessing stage. Ensuring that the unclean data is cleaned before being supplied to the model was challenging task. Some of the task I performed are:

* Filling the null values using KNN imputer
* Check if any column have zero standard deviation
* Converting categorical data to numeric values
* Removal of outliers
* Experimentation between PCA and Clustering techniques before fitting the training data into the model. Clustering technique gave higher degree of accuracy and therefore was used

**Q29. In which mode have you deployed your model?**

I have deployed the model both in cloud environments as well in the on-premise ones based on the client and project requirements

**Q30. How were you doing deployment?**

The mechanism of deployment depends on the client's requirement. For example, some clients want their models to be deployed in the cloud, and the real-time calls they take place from one cloud application to another. On the other hand, some clients want an on-premise deployment, and then they do API calls to the model. Generally, we prepare a model file first and then try to expose it through an API for predictions/classifications. The mechanism in which he API gets called depends on the client requirement