



### Q1.What is the Problem statement ?

 Wafer Fault Detection : Wafer is the Kind of Industrial electronic sensor, Which contains about 500 sub-sensors . To monitor each and Every sensor is very difficult task to maintenance team.( it’s takes lots of time to find the Faulty sensors , till then we need to stop the production line..) . so overall loss of company is increases as the maintenance time increases because of stopped production line & hence productivity decreases.

In this project , we’re receiving the Sensors dataset , & we are predicting it ( Faulty / healthy) using Machine learning concept , so that faulty sensor will easily catch & maintenance time will reduces ,in that way company loss due to (time/stopped prod line) also reduces.

## Q2 : Which tools You’ve used?

 Language : Python

Libraries : Sklearn , pandas , numpy , .git , matplotlib , seaborn Clouds : AWS (EC2)

Storage : Mongo DB , SQL lite3 IDE : pycharm / Vscode WebFrame work : Flask

Front end Laguages : HTML , CSS etc

### Q3. How you can classify your project wrt Batch learning and Online Learning?

 This project is the type of batch learning , in this project we are training the model in offline and once we satisfied with the performance of the model , then we deployed it on server for serving the predictions.

### Q4 .Will your model training happen on fly ?

 No , because of Batch learning

### Q5.Why Not Online Learning , Now ?

 The Biggest problem in Online learning system is Bad Data ,

If we feed the bad data to the online learning system and trained on it

Then the system performance is gradually decreases. And it’s very harmful to our system and for that we need to design the Entire pipeline most of the time it’s very tricky & time Consuming task ( required lot’s of experiment )….

Bad data will come Due to the Malfunctions in the sensors ,

And as Our project is based on the sensors there will be high chance of getting such

data ( corrupted data) …

Reason 2 : The Online Learning system will be suitable for the problems ( where data is changing rapidly Like( stock market , business order predictions..) ,

In this Case system suppose to keep update it self..

In our case ,data isn’t much dependent on the Outside factor (like market positions

, economical crises and etc) and hence we decided to use the Batch learning system..

### Q6: What type of problem you will be facing in future with this project ?

 It’s very Good Question …

1. **As we move ahead , we’ll be receiving more and more data ,and to make the**

system more & more Robust we need to train the system on (Old data + new data )

1. **As the Data Volume will high , it’ll take more time to Learn & Costly , at that time requirement of resources will increases ..**
2. **To store the Old data & new data , we’ll be requiring the High storage Capacity ..**
3. **To train the model on the Such Huge dataset , System configuration need to be updated in future ..**
4. **So there will be chance that increase in Handling Charges….**
5. **If we want to train model on new data , we need to replace entire system with**

new system from scratch ( it’ll be Painfull)

### Q7:How you can get rid from Above Problem?

 Yah, Well !!!!

I think we need to Change as per Environment ….

As we’re Facing the Such Challenges ( Improving the system Configuration and Build New Infrastructure is not a good option , Because it’s a repetative Problem , It’ll last till end & increases rapidely) , we need to think new ….

1. **We can shift to Online Learning system , but we need to take Pain of Maintaining the Data Pipeline , to make sure that Good Input Data..**
2. **It’ll be Good Option to Build the pipeline Which avoid the Corrupted data , using**

our past learning knowledge , so that our Repetative problem will be solved ….

### Q7: What kind Of data You’re receiving , size / Volume & in which Form ?

 We’re receiving the Sensors data in batch Format..

No cols : 500 ( Number of Sub sensors )

No Rows : approx. 1000 ( total number of sensors/wafer )

Format of Data : ‘CSV’ ( comma Separated Values)

Q8: What Type Of Storage You’ve used ?

 Initially all the Raw training Files are Kept in the ‘Training Dataset Folder’ We’re receiving the Raw training Data from that dir ..

Training dir → train Data Validation → data processing → Stacking the all Good data together → SQL lite3

- We have used SQL lite 3.

Q9 : Why SQL lite3 database ?

 As we have data in the Form of Rows and Columns Hence we’ve used SQLLite3

database.

## Q10 : How you can improve the Database system ?

 We can Store the initial Raw training files in S3 buckets .. and then by importing the dataset from S3 bucket we can do the Further operations …

 Once the Dataset is Filtered , we can store this Good dataset to the Good\_data\_dir in S3 bucket & utilize this data for further training..

 And Bad data to Archive Folder..

### Q11. What type Of Data Validation You’re Performing ?

 In data Validation Part we have Separate rules for Training and Testing Dataset in the form of Json Schema as per Data sharing Agreement ..

In Training Validation : we’re Checking the File name , number of columns , Type of columns , Date stamp , time stamps etc …

In Testing Validation : we’re Checking the File name , number of columns , Type of

columns , Date stamp , time stamps, % of missing Values etc …

Q12: Why Data Validation ?

I order to maintain the Flow Of ml system .. we need to make sure that the data attributes and types are same eg.( File name , number of columns , Type of columns

, Date stamp , time stamps etc …)

Ans : ML is continuous learning system , we need to train it continuously ( periodic learning on new + Old data ) ..

Then it’s necessary that Our Old data And New data should be Align ,

So that our training will happen successfully …

If it’s not align , Our whole system will be going for toss ….

( It’s the precautions ) ..

Q13 : What type of Preprocessing You have Done ?

 Handling the Missing Values , Feature scaling …

( all the features have Numerical data , Hence Part of Categorical data Handling is Absent)

Handling Missing Values : KNN imputer Feature Scaling : Standard Scaler

### Q14 : Why Not Outlier Detection ?

 The data we’re getting have certain range.. ( it’s production/sensors data)

After observing the ( pd.describe()) , we found that there is not much difference in Mean and median ,( it means that there is no Extreme Values are present in dataset)

The dataset having slight variation is nothing but the Faulty Sensor data..

And to detect this data we’re using multiple algorithms ..

## Q15 :What are the Different types of Missing Value Imputation Techniques?

 1.Replacing it with mean , median , mode

 2.Knn Imputer

 **3.Imputation using most frequent( From sklearn.impute import SimpleImpute)**

## Q16 : What are the Different types of Outlier Detection Techniques ,?

1. **Box plot**
2. **Using Imperial Formulae**
3. **IQr techniques**
4. **Db scan Clustering**
5. **5 star methods…**
6. **Visualizing the Scatter plot**

**Q17 : What are the Different Task you perform in Feature Engineering**

**?**

 1.Handling the Missing Values

 2.Outlier detection

 3.Handling Categorical Data

 4.rescaling Operation ( If required)

 5.Variable Transformation

 Creating new Features.

## Q18 :Various Methods of the Feature Selection Techniques?

 1.Pearson Correlation 2.Chi2 test (k-best) 3.Backtracking method

4.Feature importance methods (tree based) 5.Annova

6.variance threshold methods 7.Fisher Score

1. **BackWord Elemination**
2. **info Gain (mutual\_info\_class) 10.Lasso Reg( )**

## Q19: What is the Size of the Data ?

 Data is present in the Form of batches , Around(12-14) csv files Each file consist around 150 records of sensors data..

Each sensors Have 590 Sub-sensors Total size = 16.00mb

# Q20 : Why not You’re using PCA (as you have 590 cols) ?

 Actually , it’s sensor based dataset ,

Each Column is representing the Particular sensor and the presence of every sensor information is very important to detect the Faulty sensor ..

All sub-sensors data is not correlated with each other ( all are independent ) and

**single Faulty sub-sensor can make whole wafer Faulty ..**

**And hence to Avoid the data loss we’re not using PCA here.…**

If we use PCA here there will be chance of losing some important information , which can affect Our prediction in bad manner…

## Q21 : Is There Any Chance Of Overfitting or Curse Of Dim’s ?

 There are to types of data : 1. Wide data 2.Long data

And our dataset belongs to Wide data Category ( large cols , less rows)

Obviously , whenever we have large number of cols , the chance of Overfitting also increases , to avoid the Overfitting We can reduces the Number of Cols , use PCA But sadly we can’t apply those technique in this project ..

As the Number of Features increases the model Complexity also increases & model need to learn more from this features and hence we’ll end up with the Overfitting .. There May be Some reasons behind that , ( multi co-linearity , when Training and test data is not matching , unproper selections of Hyper-params , un-filtered data & etc)..

# CURSE-OF\_Dims :

**Curse of dimensionality happens when we have lot’s of features and less rows , and during Training Model is not able to identify the relation/pattern between input and Output Features.**

**Q22 : How you’re Fighting With Curse Of Dimensionality ?**

To avoid The Curse Of Dims we’re using K-mean Clustering Technique ,

Which Help us to Grouped the Data based on there similar characteristic , so that model will be able to identify the Pattern Easily ..

After that we’ll be Applying the Different Different algorithms to Each group & find which

one is performing Better for which group , and after Validating the model based on

decided Performance matrix , we’ll be finalized the model ..

## Q20 : What are different Cross-Validation Techniques , Which one you used in your project & Why ?

1. K-fold , Leave One Leave out , Stratified k-fold , Grid search cv , Hold Out Grid Search Cv : To find out the Best Parameters for model ..

**k-fold :** *- helps to overcome the Overfitting , gives the clear idea about the model Performance*

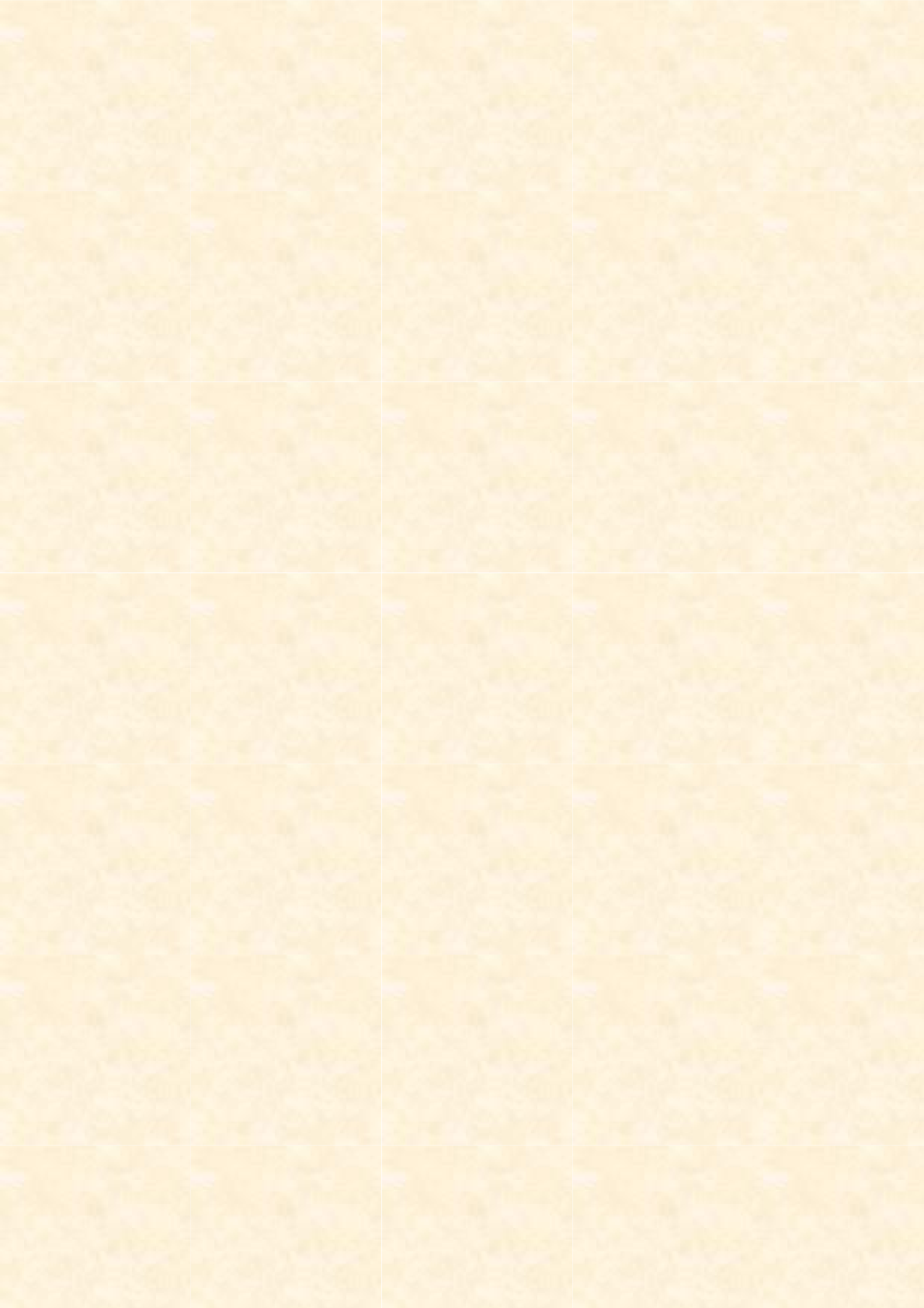
*- If the Range between highest score and lowest score is high , then it'll the sign of overfitting ,*

*we need todo hyper-prameter tunning again*

**stratified K-Fold :** *helps to overcome the Overfitting , gives the clear idea about the model Performance*

* *if dataset is unbalanced , then we use strati fied K-fold validation for the model validation ,*
* *StratifiedKFold , ensures that the equal propo rtion of element should be choosen from each class whil e cross-validation*
* *If the Range between highest score and lowest score is high , then it'll the sign of overfitting ,*

*we need todo hyper-prameter tunning again*

Hold Out Validation Approach :

***''' It's simple train test split dataset , divides datas et in two part(train & test )***

***- data selected randomly , as we change the Rand om\_state model Accuracy will change it's the***

***major drawback of this approch..***

***'''***

## In this project we have used , Grid Search cv , cross val score , hold out , stratified K-fold..

**Q21 : Which Performance Matrix You have Used in Project & Why , Talk more about the Various Performance Matrixes ?**

###  We’re reducing the FN .. ( trying to improve the sensitivity)

**No any faulty sensor will be detected as Normal …..**

**If we’re predicting (normal sensor as Faulty ) up to Some extent it’s Digestible , because the maintenance person can go at particular sensor Location & cross check it….**

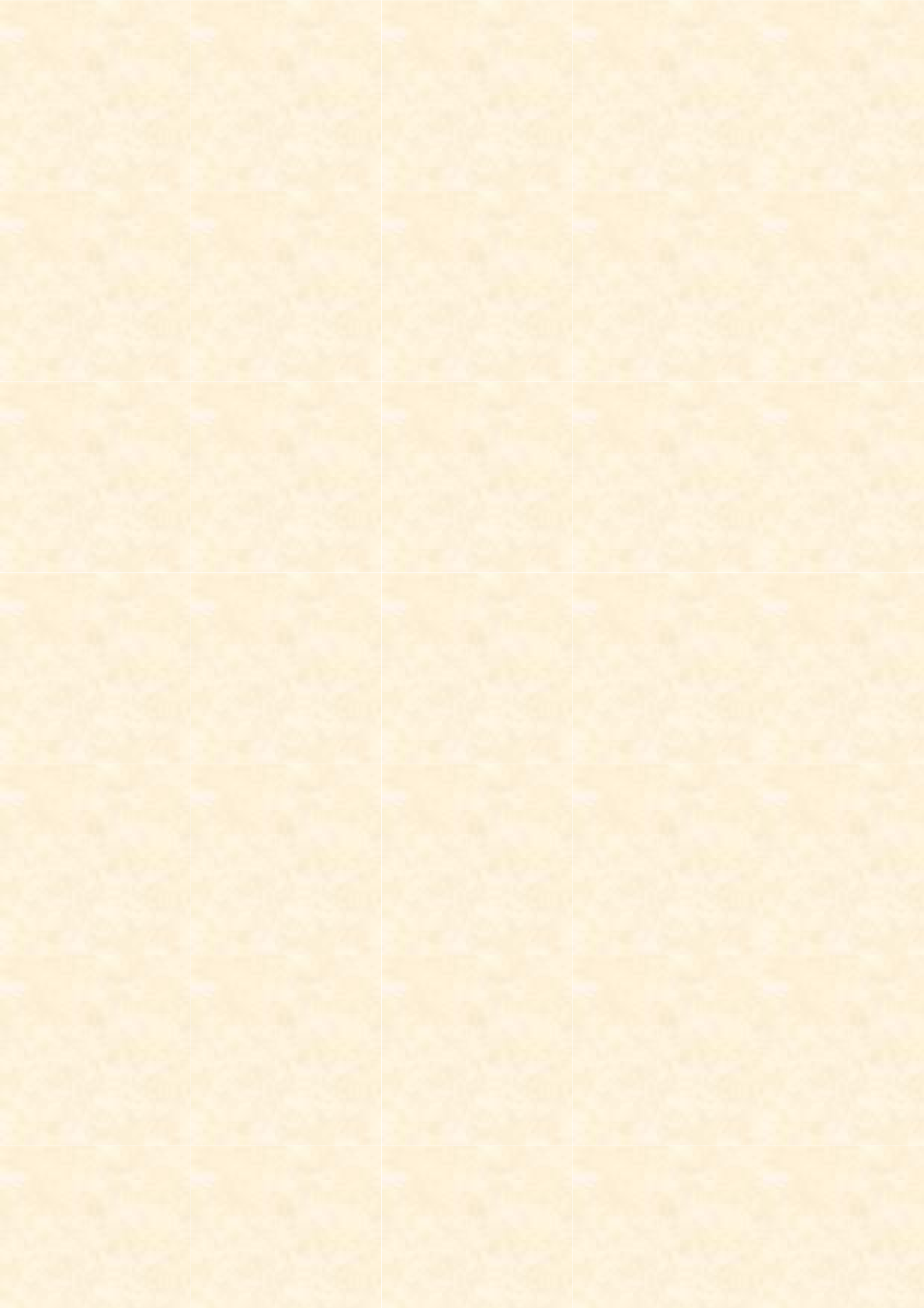
**And if we’re predicting the Faulty sensor as Normal Sensor ,**

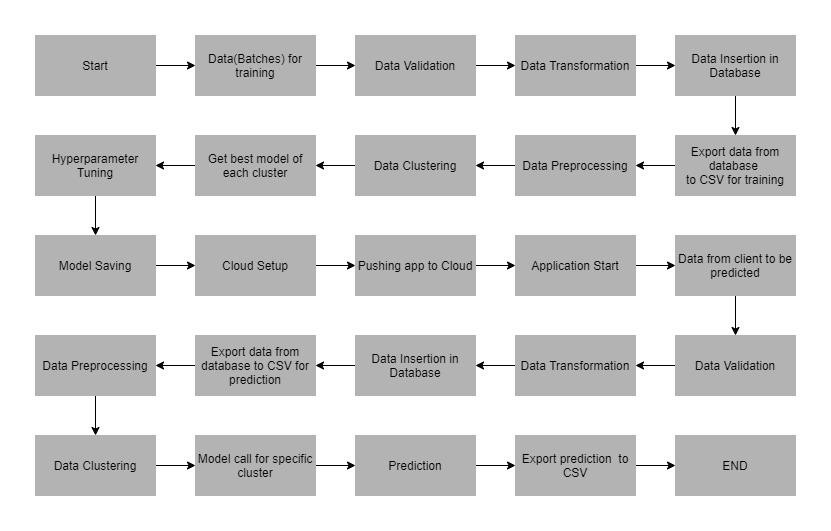
**Then it’s again very difficult to track the Faulty sensor & it’s very time Consuming task…. And it’ll not be suitable for Our Project title…**

**Confusion\_matrix : talk abt Conf matrix ( FP , FN , TP , TN ) Sensitivity OR recall = TP / ( TP + FN ) ,**

**Precision = TP / ( TP + FP )**

**In this project we’re also using Accuracy , ROC and AUC curve…**

**Q22 : What is the Structures & Pipeline Of your project ?**

 

 1.Data Validation Pipeline 1.Data Validation Pipeline 2.Data Preprocessing Pipeline

3.Database Insertion Pipeline 4.Clustering Pipeline

1. **Model Training pipeline ,**
2. **Model Selection pipeline( from model buckets) based on decided 7.Also Pipeline for the ( prediction Module)**

## Q24: How you’re making Coding part more Generalized ,and make maintanence Easy ? ( talk about Logs)

 In this project , all the part of Coding Done in Modular fasion ..

Having some Advantages : Reusability , Easy to Understand , Easy to debug , Easy to make changes in particular Pipeline..

Maintaining the Logs : so that we can track the Internal Operations ,and we’ll be getting the current as well as historical info about run in text format ,, based on that we can Analyze & Optimized the Code & processes..

## Q25: What kind of Cloud Services You’re using( Cloud You’re Using ) ?

 **We’re using AWS Cloud Services here…**

**Instance Type : EC2 instance ( Spot Instance)**

* 1. **Spot instance is cheap at cost , and Good when we required instance for shorter period of time..**
  2. **We’ll get about 90% discount**
  3. **Your instances can be Terminated at any time , because at some time someone can bid Higher price for the same instances , it’s the Disadvantages.**
  4. **It’s usefull when cost Effectiveness Comes into Picture ..**

Why AWS ? :::: provides world class security , having good and reliable services ,

Scalable , having Elastic property ( we can extend the limit of our services any time , storage Facility , (pay for use Policy )

## Q26: How Frequently You decided to train model ?

1. To keep the model update we need to retrain the model on Old + new data So that system will be robust and more Generalized…

Basically , the retraining of model is depends on how frequently your data is changing ( rate of Change of data ) …

If rate of Change of( frequency ) is very high , then we need to retrain the model Rapidly ( once a day , week , once a month )

On other hand :

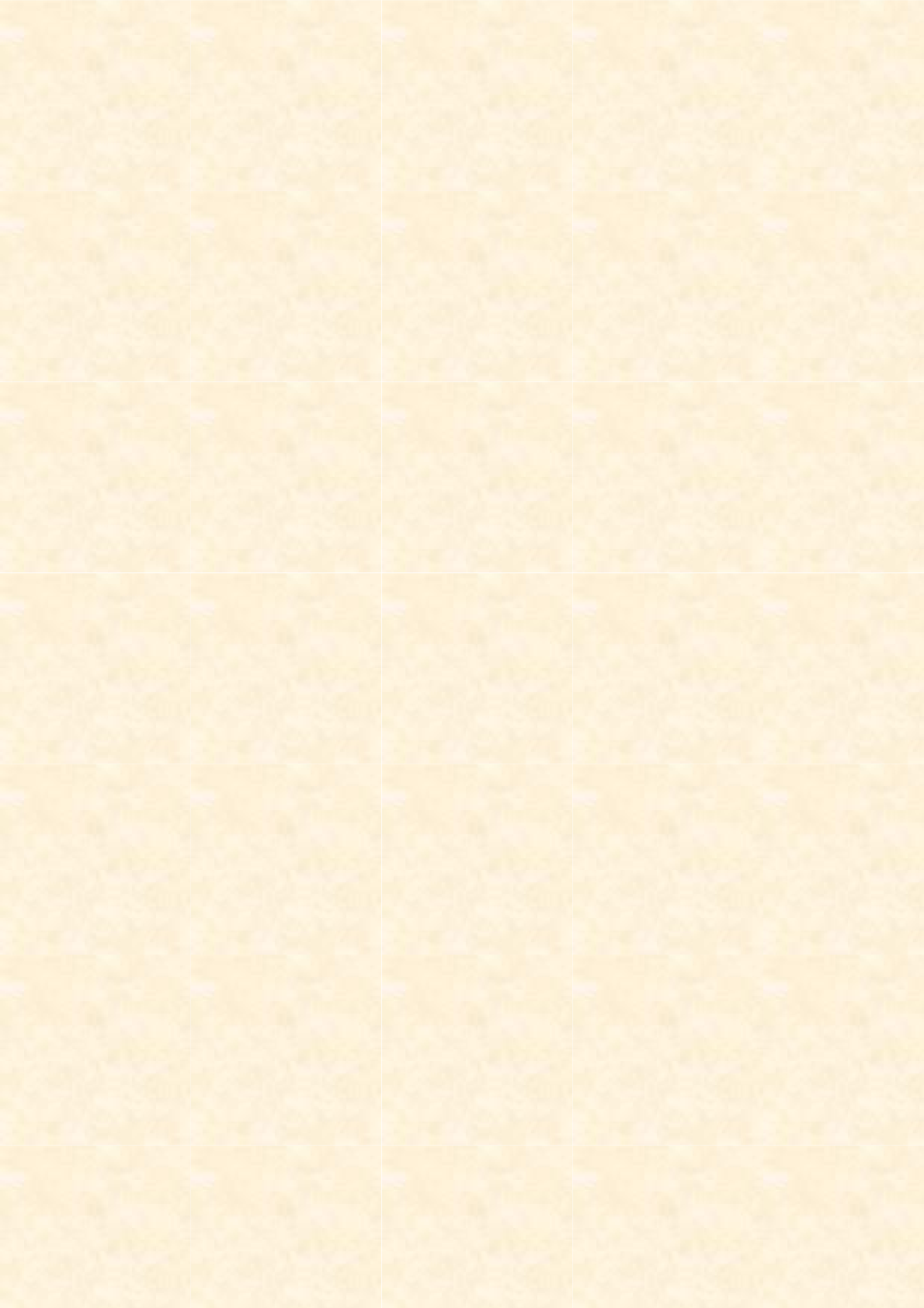
If rate of Change of( frequency ) is very less , then we not need to retrain the model Rapidly , we can retrain the model ( once a 2/3 months)

In our case , the Sensors data is not much depends on the Outside Factors , and hence very slightly changing day by day ..

And also if , industry is installing one more sensors ( or willing to change in electronic infrastructure ) then we need to modify our whole system also …

Hence , We can retrain the Our Models ( once 1 or 2 months)..

## 27 : What is the Team Size & Your role ?

 7 team member :

 Actually the role was not fixed , every week we get the task to perform and we

have to work on that task …

 Mostly I was active in installing data preprocessing pipeline , model creation pipeline and deployment part ;

## Q28 : Problem involved in Project Duration ?

**Every week New task / Not fixed role : I learn to be a dynamic**

**from this activity ….**

**Aligning the team members . Maintaining the co-ordination in team , Handling such wide dataset , EDA ,**

**deployment part , tuning the model with such high dimensions..**

**Q29 : Why not Feature Selection ?**

 In this project we have around 590 features ..

And Each and Every Feature is Contributing in prediction..

Every single Features contains the information of the One Sub-Sensor ,

And if we reject any columns there will be more important information loss will

happen , hence we’re not using the feature selection technique..

In some use cases Feature selection is not best option …

So we need to decide , how we can design a system / train model with that much

features….

## Q30 : What kind of Clustering you have done and Why (k-mean)?

For this project I have used K means clustering technique which is a non hierarchial based clustering because we can define number of clusters we want to distribute the dataset for our model.

**Q31 : Which Algorithms used in Project , Why ( details about Algos) ?**

 **K-mean clustering**

 **Random Forest Classifier**

 **XGB Classifier**

**( Explain the Operation of Each )**

**Why these classifiers only ?**

1. **we tried with all algos but Random Forest and XGB performing better ..**

**Random Forest and XGB classifier can Handle the Wide data ….**

**Mention the advantages of each**