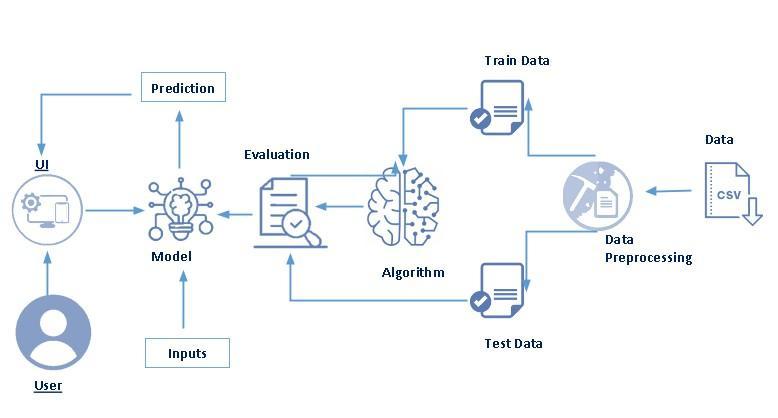
Wafer Fault Prediction on Sensors data

# Project Description:

* In electronics, a wafer (also called a slice or substrate) is a thin slice of semiconductor, such as a crystalline silicon (c-Si), used for the fabrication of integrated circuits and, in photovoltaics, to manufacture solar cells. The wafer serves as the substrate(serves as foundation for contruction of other components) for microelectronic devices built in and upon the wafer.
* It undergoes many microfabrication processes, such as doping, ion implantation, etching, thin-film deposition of various materials, and photolithographic patterning. Finally, the individual microcircuits are separated by wafer dicing and packaged as an integrated circuit.

# Technical Architecture:



# Pre requisites:

**To complete this project, you must required following software’s, concepts and packages**

* **Anaconda navigator and pycharm:**
  + Refer the link below to download anaconda navigator
  + Link : <https://youtu.be/1ra4zH2G4o0>
* **Python packages:**
  + Open anaconda prompt as administrator
  + Type “pip install numpy” and click enter.
  + Type “pip install pandas” and click enter.
  + Type “pip install scikit-learn” and click enter.
  + Type ”pip install matplotlib” and click enter.
  + Type ”pip install scipy” and click enter.
  + Type ”pip install pickle-mixin” and click enter.
  + Type ”pip install seaborn” and click enter.
  + Type “pip install Flask” and click enter.

# Prior Knowledge:

You must have prior knowledge of following topics to complete this project.

* **ML Concepts**
  + Supervised learning: <https://www.javatpoint.com/supervised-machine-learning>
  + Unsupervised learning: <https://www.javatpoint.com/unsupervised-machine-learning>
  + KNN: <https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning>
  + Logistic Regression: [https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-logistic-regression/](https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/)
  + Evaluation metrics: <https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/>
* **Flask Basics** : <https://www.youtube.com/watch?v=lj4I_CvBnt0>

# Project Objectives:

By the end of this project you will:

* Know fundamental concepts and techniques used for machine learning.
* Gain a broad understanding about data.
* Have knowledge on pre-processing the data/transformation techniques on outlier and some visualization concepts.

# Project Flow:

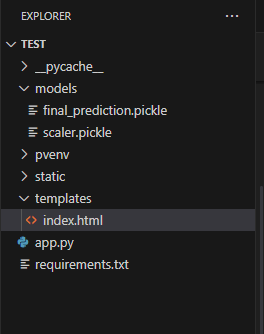
* User interacts with the UI to enter the input.
* Entered input is analyzed by the model which is integrated.
* Once model analyses the input the prediction is showcased on the UI

To accomplish this, we have to complete all the activities listed below,

* Data collection
  + Collect the dataset or create the dataset
* Visualizing and analyzing data
  + Univariate analysis
  + Bivariate analysis
  + Multivariate analysis
  + Descriptive analysis
* Data pre-processing
  + Checking for null values
  + Handling outlier
  + Handling categorical data
  + Splitting data into train and test
* Model building
  + Import the model building libraries
  + Initializing the model
  + Training and testing the model
  + Evaluating performance of model
  + Save the model
* Application Building
  + Create an HTML file
  + Build python code

# Project Structure:

Create the Project folder which contains files as shown below­:



* We are building a flask application which needs HTML pages stored in the templates folder and a python script app.py for scripting.
* model.pkl is our saved model. Further we will use this model for flask integration.

# Milestone 1 : Define Problem/Problem Understanding

## Activity 1: Specify the business problem

* Wafers are predominantly used to manufacture solar cells and are located at remote locations in bulk and they themselves consist of few hundreds of sensors. Wafers are fundamental of photovoltaic power generation, and production thereof requires high technology. Photovoltaic power generation system converts sunlight energy directly to electrical energy.
* The motto behind figuring out the faulty wafers is to obliterate the need of having manual man-power doing the same. And make no mistake when we're saying this, even when they suspect a certain wafer to be faulty, they had to open the wafer from the scratch and deal with the issue, and by doing so all the wafers in the vicinity had to be stopped disrupting the whole process and stuff and this is when that certain wafer was indeed faulty, however, when their suspicion came out be false negative, then we can only imagine the waste of time, man-power and of course, cost incurred.
* Data fetched by wafers is to be passed through the machine learning pipeline and it is to be determined whether the wafer at hand is faulty or not apparently obliterating the need and thus cost of hiring manual labour.

## Activity 2: Business requirements

The machine learning model must meet the following requirements:

* Accuracy: The model must be able to predict whether a wafer has defects with an accuracy of at least 95%.
* Speed: The model must be able to predict whether a wafer has defects in real time so that defective wafers can be removed from the production line immediately.
* Scalability: The model must be scalable to handle large volumes of sensor data.
* Deployment: The model must be deployed in a production environment so that it can be used to predict whether wafers have defects in real time.

## Activity 3: Literature Survey

Wafer fault detection is a critical task in the semiconductor manufacturing process. Defects in wafers can lead to the failure of integrated circuits, which can be costly and time-consuming to replace. Manually inspecting wafers for defects is a tedious and error-prone process. Machine learning can be used to develop models that can accurately predict whether a wafer has defects based on the sensor data collected during the manufacturing process.

Recent advances in machine learning have led to the development of a variety of new methods for wafer fault detection. Some of the most promising methods include:

* Deep learning: Deep learning models can learn complex patterns in the sensor data that are associated with wafer defects. Deep learning models have been shown to achieve state-of-the-art results on a variety of wafer fault detection tasks.
* Transfer learning: Transfer learning can be used to train machine learning models on wafer fault detection tasks using pre-trained models that have been trained on other tasks. This can save time and computational resources, and it can also help to improve the performance of the model.
* Ensemble learning: Ensemble learning combines the predictions of multiple machine learning models to produce a more accurate prediction. Ensemble learning methods have been shown to be effective for wafer fault detection tasks, especially when the sensor data is noisy or incomplete.

## Activity 4: Social or Business Impact.

**Social Impact**

* Improved product quality: By reducing the number of defective wafers that are produced, machine learning can help to improve the quality of products that are made with semiconductors. This can lead to fewer product recalls and improved consumer satisfaction.
* Reduced costs: By reducing the number of defective wafers that are produced, machine learning can help to reduce the costs associated with semiconductor manufacturing. This can lead to lower prices for consumers and increased profits for businesses.
* Increased safety: Machine learning can help to improve the safety of semiconductor manufacturing by detecting defects that could lead to accidents. This can help to protect workers and prevent environmental damage.

**Business Impact**

* Increased productivity: Machine learning can help to increase the productivity of semiconductor manufacturing by automating the wafer inspection process. This can free up workers to focus on other tasks and can lead to increased production output.
* Reduced waste: Machine learning can help to reduce waste by detecting defects in wafers early in the manufacturing process. This can help to prevent defective wafers from being processed further, which can save time and money.
* Improved customer satisfaction: By reducing the number of defective wafers that are produced, machine learning can help to improve customer satisfaction. This is because customers are less likely to receive defective products if the wafers used to make those products have been carefully inspected.

# Milestone 2: Data Collection

ML depends heavily on data, It is most crucial aspect that makes algorithm training possible. So this section allows you to download the required dataset.

**Activity 1: Download the dataset**

There are many popular open sources for collecting the data. Eg: kaggle.com, UCI repository, etc.

In this project we have used Mental Attention State Detection Using EEG data. This data is downloaded from kaggle.com. Please refer the link given below to download the dataset.

Link: <https://www.kaggle.com/datasets/manavpatel571/wafer-dataset>

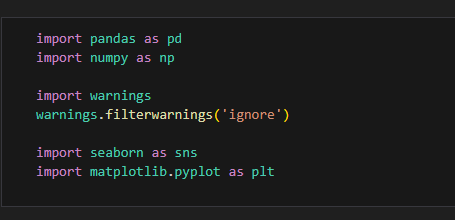
# Milestone 3: Visualizing and analysing the data

As the dataset is downloaded. Let us read and understand the data properly with the help of some visualization techniques and some analysing techniques.

**Note: There is n number of techniques for understanding the data. But here we have used some of it. In an additional way, you can use multiple techniques.**

**Activity 1: Importing the libraries**

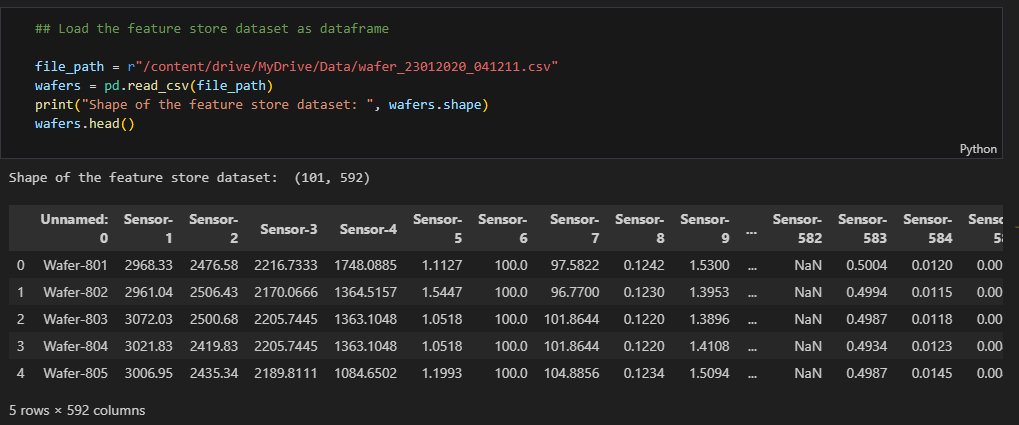
Import the necessary libraries as shown in the image.



**Activity 2: Read the Dataset**

Our dataset format might be in .csv, excel files, .txt, .json, etc. We can read the dataset with the help of pandas.

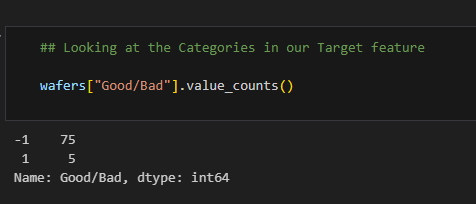
In pandas we have a function called read\_csv() to read the dataset. As a parameter we have to give the directory of csv file.



**Activity 3: Univariate analysis**

In simple words, univariate analysis is understanding the data with single feature. Here we have displayed two different graphs such as distplot and countplot.

* Seaborn package provides a wonderful function distplot. With the help of distplot, we can find the distribution of the feature. To make multiple graphs in a single plot, we use subplot.



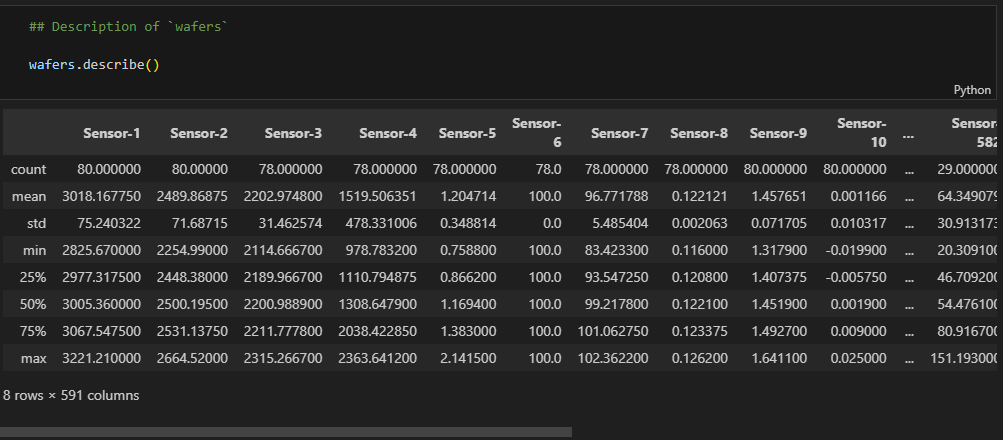
**Activity 4: Bivariate analysis**

To find the relation between two features we use bivariate analysis.

**Activity 5: Multivariate analysis**

In simple words, multivariate analysis is to find the relation between multiple features.

**Activity 6: Descriptive analysis**

****

Descriptive analysis is to study the basic features of data with the statistical process. Here pandas has a worthy function called describe. With this describe function we can understand the unique, top and frequent values of categorical features. And we can find mean, std, min, max and percentile values of continuous features.

# Milestone 4: Data Pre-processing

As we have understood how the data is lets pre-process the collected data.

The download data set is not suitable for training the machine learning model as it might have so much of randomness so we need to clean the dataset properly in order to fetch good results. This activity includes the following steps.

* Handling missing values
* Handling categorical data
* Handling outliers
* Scaling Techniques
* Splitting dataset into training and test set

Note: These are the general steps of pre-processing the data before using it for machine learning. Depending on the condition of your dataset, you may or may not have to go through all these steps.

**Activity 1: Checking for null values**

* Let’s find the shape of our dataset first, To find the shape of our data, df.shape method is used. To find the data type, df.info() function is used.
* For checking the null values, df.isnull() function is used. To sum those null values we use .sum() function to it. From the below image we found that there are no null values present in our dataset. So we can skip handling of missing values step.

Let’s look for any outliers in the dataset

**Activity 2: Handling outliers**

With the help distribution, outliers are visualized. And here we are going to find upper bound and lower bound of all features with some mathematical formula.

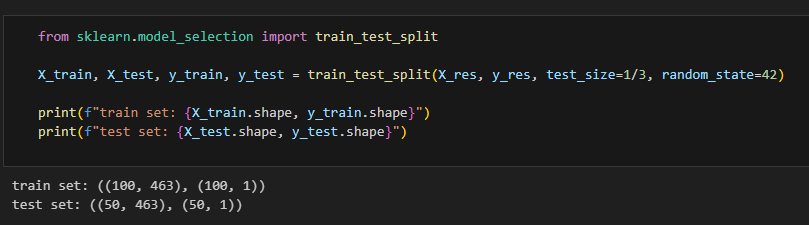
* From the below diagram, we could visualize that Distribution of feature has outliers or Not.



**Activity 3: Splitting data into train and test**

Now let’s split the Dataset into train and test sets. First split the dataset into x and y and then split the data set

Here x and y variables are created. On x variable, df is passed with dropping the target variable. And on y target variable is passed. For splitting training and testing data we are using train\_test\_split() function from sklearn. As parameters, we are passing x, y, test\_size, random\_state.



# Milestone 5: Model Building

Now our data is cleaned and it’s time to build the model. We can train our data on different algorithms. For this project we are applying Three classification algorithms KNN, Logistic Regression and Random Forest Classifier. The best model is saved based on its performance.

Link:

<https://colab.research.google.com/drive/1fZvnhf3_LfLhFqMqcaALnqZVpOdwGjK7?usp=sharing>

# Milestone 6: Application Building

In this section, we will be building a web application that is integrated to the model we built. A UI is provided for the uses where he has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI.

This section has the following tasks

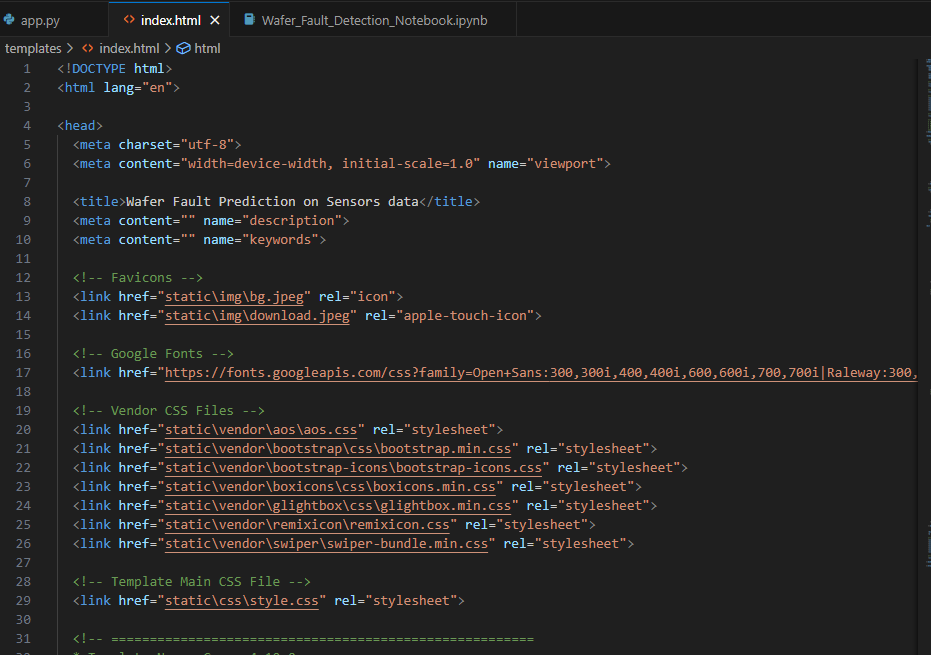
* Building HTML Pages
* Building serverside script

**Activity1: Building Html Pages:**

For this project create three HTML files namely

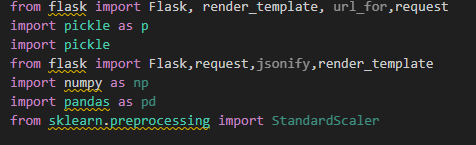
* Index.html

and save them in templates folder.

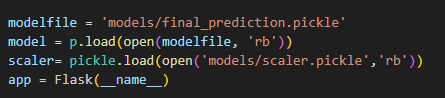


**Activity 2: Build Python code:**

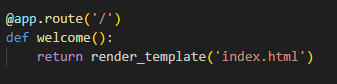
Import the libraries

****

Load the saved model. Importing flask module in the project is mandatory. An object of Flask class is our WSGI application. Flask constructor takes the name of the current module (\_\_name\_\_) as argument.



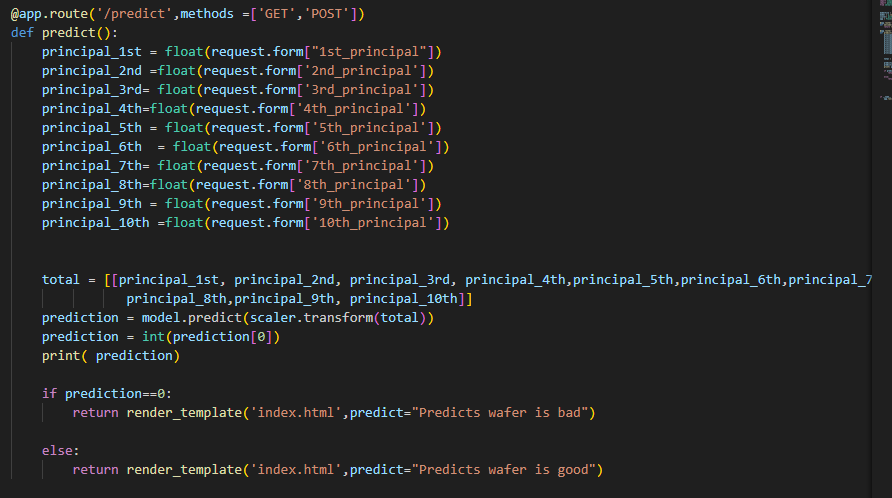
Render HTML page:



Here we will be using declared constructor to route to the HTML page which we have created earlier.

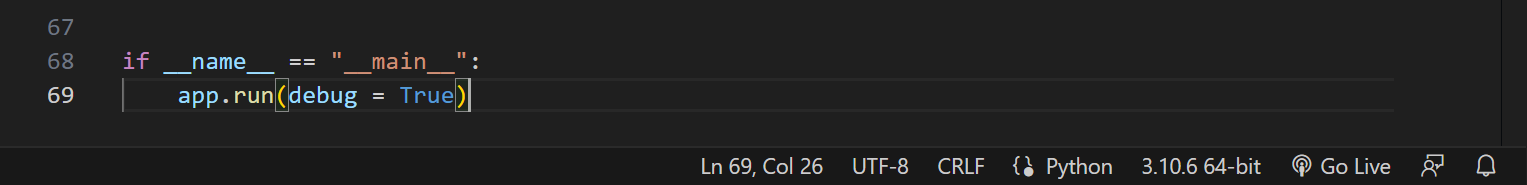
In the above example, ‘/’ URL is bound with home.html function. Hence, when the home page of the web server is opened in browser, the html page will be rendered. Whenever you enter the values from the html page the values can be retrieved using POST Method.

Retrieves the value from UI:



Here we are routing our app to predict() function. This function retrieves all the values from the HTML page using Post request. That is stored in an array. This array is passed to the model.predict() function. This function returns the prediction. And this prediction value will rendered to the text that we have mentioned in the submit.html page earlier.

Main Function:



**Activity 3: Run the application**

* Open anaconda prompt from the start menu
* Navigate to the folder where your python script is.
* Now type “python app.py” command
* Navigate to the localhost where you can view your web page.
* Click on the predict, enter the inputs, click on the submit button, and see the result/prediction on the web.

## Final Output :

