Employee Performance Prediction

Solution Template

# Project Description:

In this project we are going to analyse and predict the performance of employees in an organization on the basis of various factors, including, but not limited to, individual and domain specific characteristics, nature and level of schooling, socioeconomic status and different psychological factors.

Here we have used Supervised learning techniques namely Support Vector Machines, Random Forest, Naive Bayes, Neural Networks and Logistic Regression which considers these factors and provides insights into

the performance and commitment of employees. The employees are classified into 3 output classes indicating the level of their performance from low to high. In this research paper, 10-fold validation

technique is used to ensure the correctness of the prediction by the above-mentioned techniques. Support Vector Machines prove to be the most efficient in terms of accuracy. The result is accentuated by the high validation score obtained by the same.

**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 required :

* + 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.

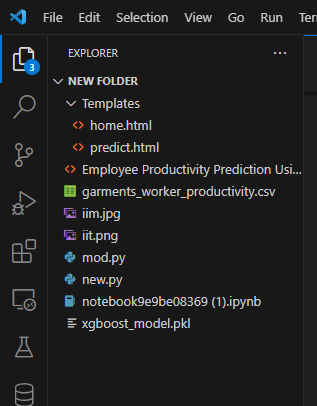
# Project Objectives:

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 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
  + Correlation analysis
  + Descriptive analysis
* Data pre-processing
  + Checking for null values
  + Handling Date & department column
  + 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:

* We are building a flask application which needs HTML pages stored in the templates folder and a python script app.py for scripting.
* gwp.pkl is our saved model. Further we will use this model for flask integration.
* Training folder contains Employee\_Prediction.ipynb , model file.
* Ibm folder contains IBM deployment files.

# Milestone 1: Data Collection

## Acquiring the necessary dataset is a pivotal step in machine learning, as the algorithms heavily rely on the quality of the data for training. For this project, the dataset "garments\_worker\_productivity.csv" was obtained from kaggle.com. To access the dataset, you can visit the provided link or explore other popular sources like the UCI repositor

## 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 garments\_worker\_productivity.csv data. This data is downloaded from kaggle.com. Please refer the link given below to download the dataset.

Link: [Productivity Prediction of Garment Employees | Kaggle](https://www.kaggle.com/datasets/utkarshsarbahi/productivity-prediction-of-garment-employees)

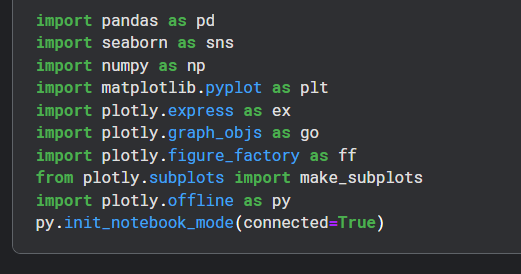
# Milestone 2: Visualizing and analysing the data

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**

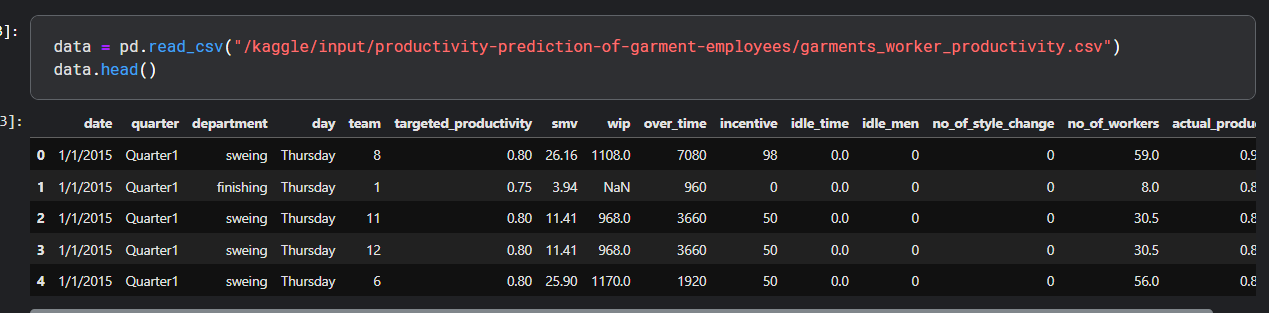
Import the necessary libraries as shown in the image.



## Activity 2: Read the Dataset

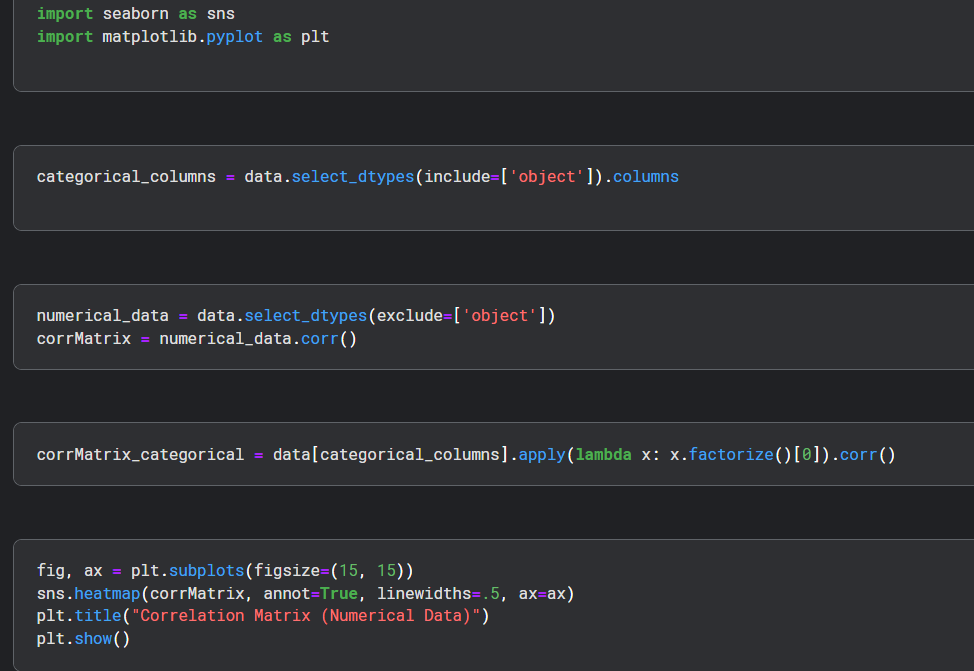
Our dataset format might be in .csv, excel files, .txt, .json, etc. We can read the dataset with the help of pandas.

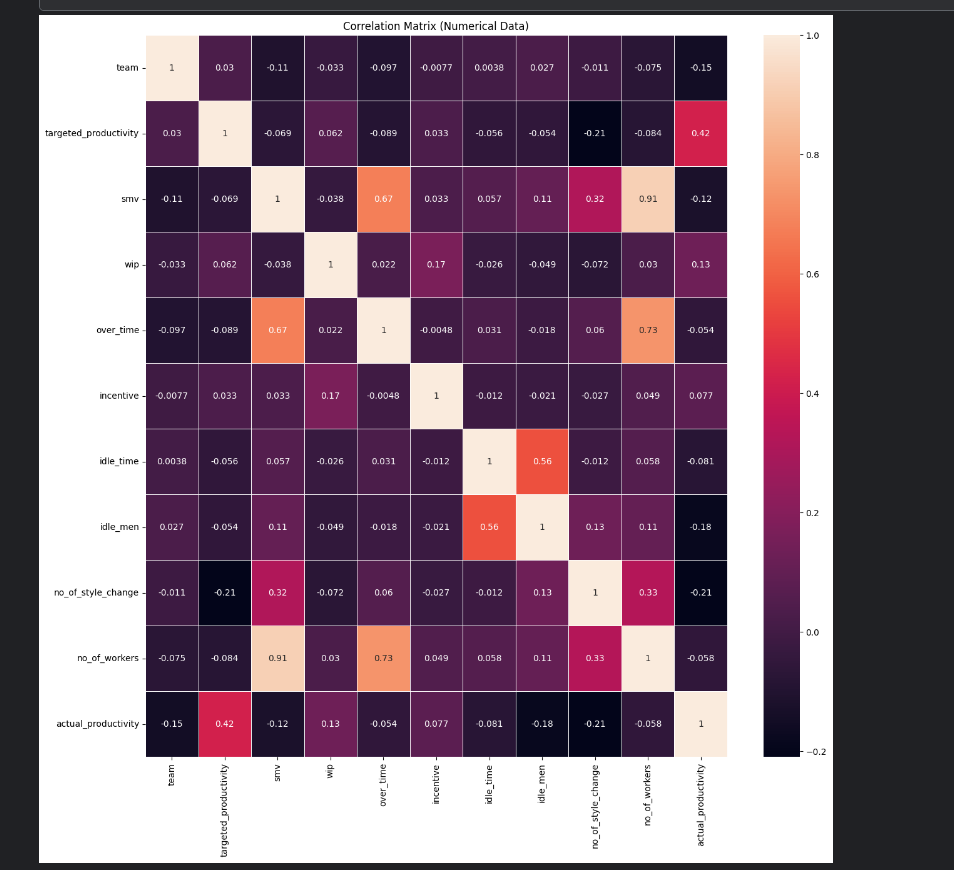
In pandas we have a function called read\_csv() to read the dataset. As a parameter we have to give the directory of csv file.



## Activity 3: Correlation analysis

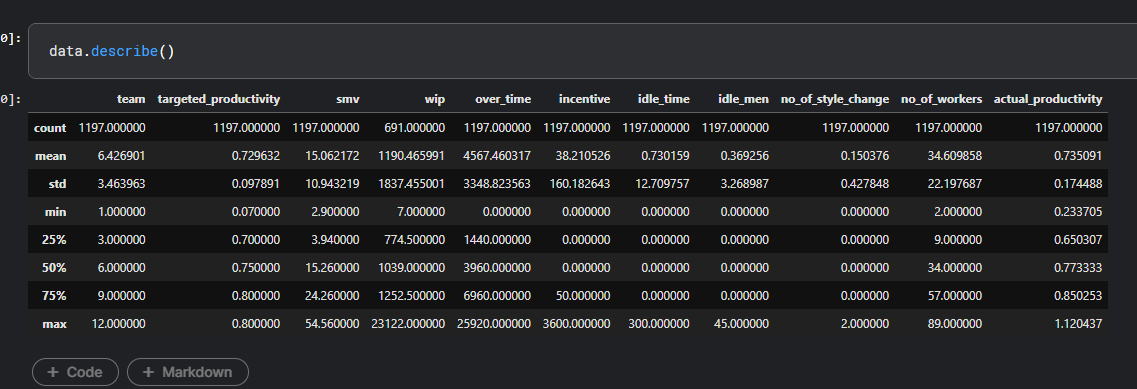
In simple words, A correlation matrix is simply a table which displays the correlation coefficients for different variables. The matrix depicts the correlation between all the possible pairs of values in a table. It is a powerful tool to summarize a large dataset and to identify and visualize patterns in the given data.





## Activity 4: 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 3: 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

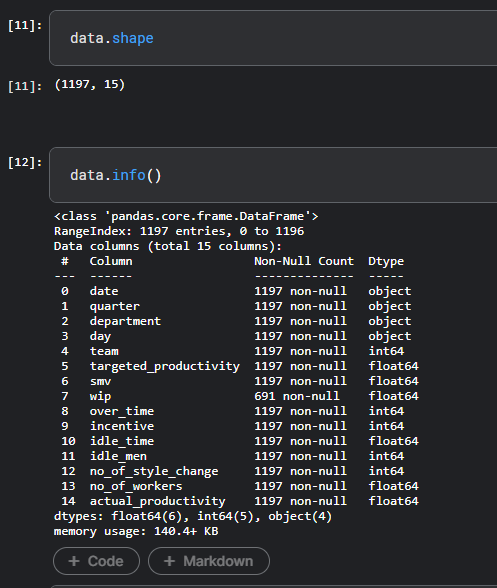
randomness so we need to clean the dataset properly in order to fetch good results. This activity includes the following steps.

* Handling missing values
* Handling Date & department column
* Handling categorical data
* 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

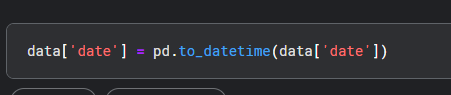
* Let’s find the shape of our dataset first, To find the shape of our data, data.shape method is used. To find the data type, data.info() function is used.



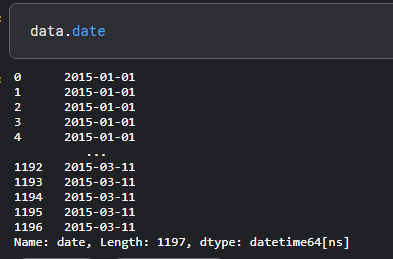
* For checking the null values, data.isnull() function is used. To sum those null values we use .sum() function to it. From the below image we found that in our dataset there is one feature which has high number of null values. So we drop that feature.

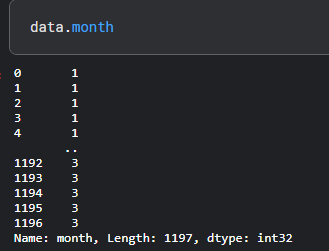
## Activity 2: Handling Date & department column

* Here what we are doing is converting the date column into datetime format.



* Then converting date column to month (month index) & transferring the values into a new column called month. As we have the month column now we don’t need date, so we will drop it.

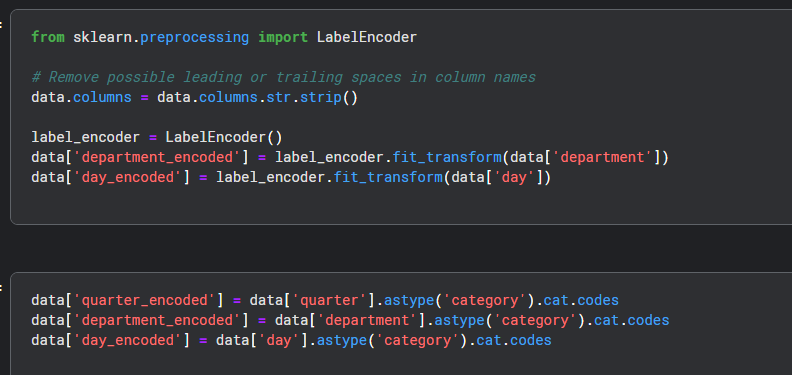




* From below image we can see that in department column the values are slit into 3 categories Sweing, finishing, finishing. Finishing class is repeating twice, so we will merge them into 1.

## Activity 3: Handling Categorical Values

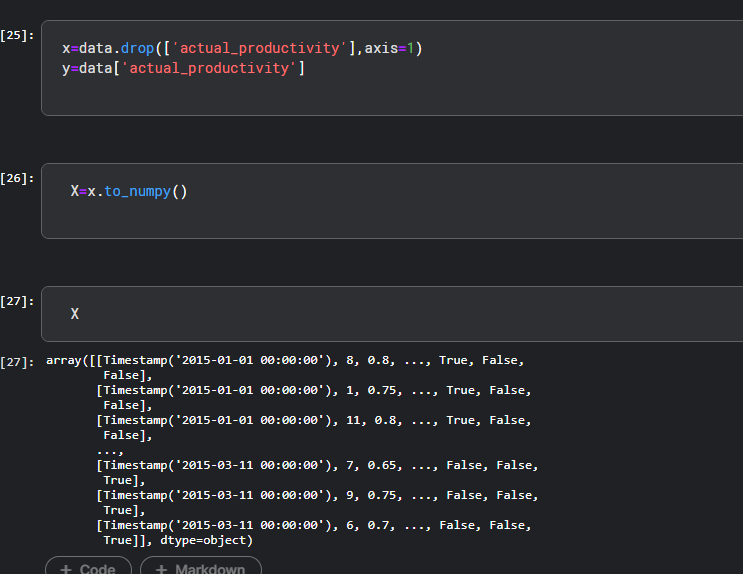
**Remove leading or trailing spaces in column names:**pythonCopy codedata.columns = data.columns.str.strip()This line ensures that any leading or trailing spaces in the column names of the DataFrame data are removed. This step is a good practice to prevent potential issues with column names.**Label Encoding:**pythonCopy codelabel\_encoder = LabelEncoder()data['department\_encoded'] = label\_encoder.fit\_transform(data['department'])data['day\_encoded'] = label\_encoder.fit\_transform(data['day'])LabelEncoder is a preprocessing technique in machine learning to convert categorical labels into numerical representations.For each unique value in the 'department' column, fit\_transform assigns a unique numerical label to it and creates a new column 'department\_encoded' in the DataFrame.Similarly, for the 'day' column, a new column 'day\_encoded' is created with numerical labels.The purpose of this encoding is to provide a way for machine learning algorithms to work with categorical data, as many algorithms require numerical input. However, it's important to note that label encoding introduces an implicit ordinal relationship between the categories, which may not always be appropriate for all types of categorical data.So, after this encoding, the 'department' and 'day' columns are replaced with their corresponding numerical representations in the new columns 'department\_encoded' and 'day\_encoded'.



## Activity 4: 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. After that x is converted into array format then passed into a new variable called X.

Here X and y variables are created. On X variable, data 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 4: 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 Regression algorithms. The best model is saved based on its performance.

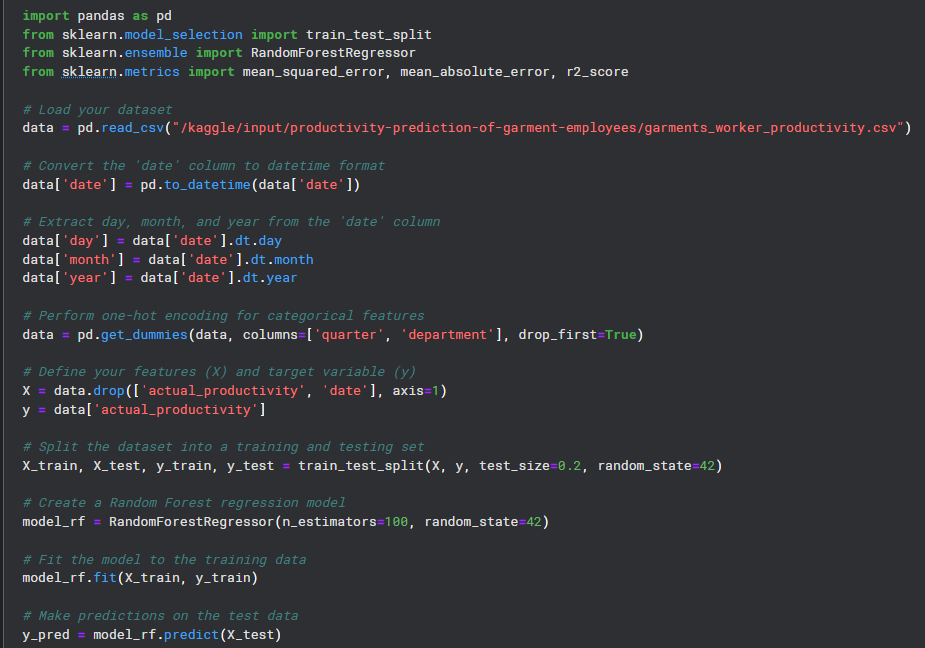
## Activity 1: Linear Regression model

Linear Regression has been initialized with the name model\_lr. Then predictions are taken from x\_test given to a variable named pred\_test. After that Mean absolute error, mean squared error & r2\_scrores are obtained.



## Activity 2: Random Forest model

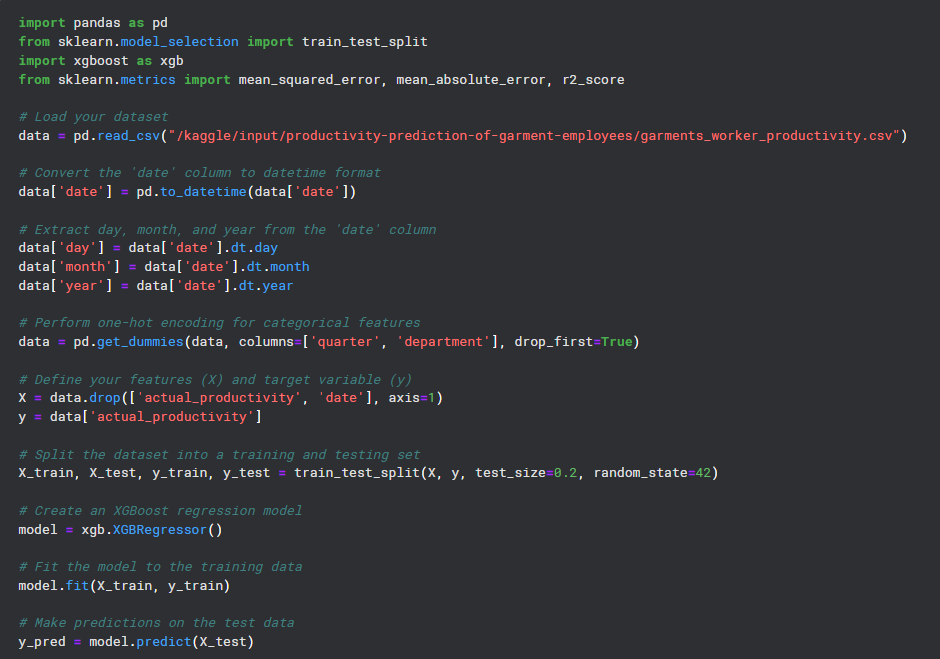
Random Forest has been initialized with the name model\_rf. Then predictions are taken from x\_test given to a variable named pred. After that Mean absolute error, mean squared error & r2\_scrores are obtained.



## Activity 3: Xgboost model

XGBoost has been initialized with the name model\_xgb. Then predictions are taken from x\_test given to a variable named pred3. After that Mean absolute error, mean squared error & r2\_scrores are obtained.

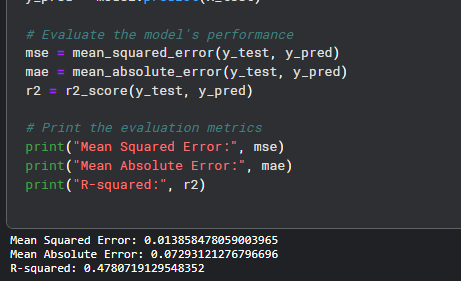
Now let’s see the performance of all the models and save the best model



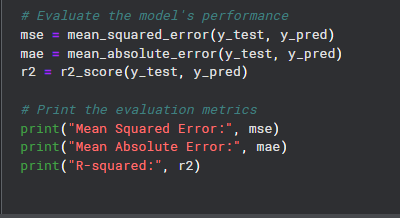
## Activity 4: Compare the model

For comparing the above three models MSE, MAE & r2\_scroes are used.

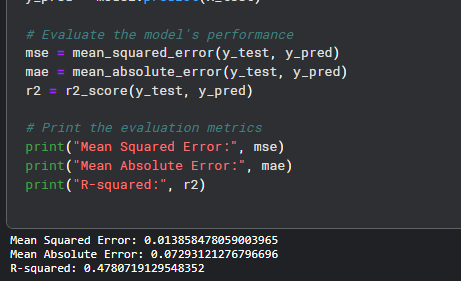
**Linear Regression:**

****

**Random Forest:**



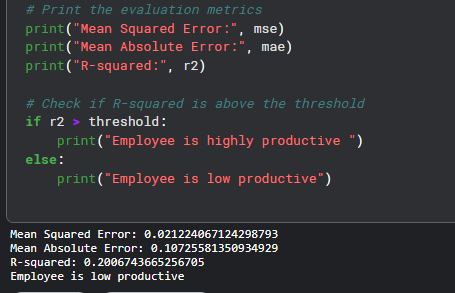
**XGBoost:**

****

After calling the function, the results of models are displayed as output. From the three model xgboost is performing well.

## Activity 5: Evaluating performance of the model and saving the model

From sklearn, metrics r2\_score is used to evaluate the score of the model. On the parameters, we have given y\_test & pred3. Our model is performing well. So, we are saving the model by pickle.dump().



# Milestone 5: 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 server side script

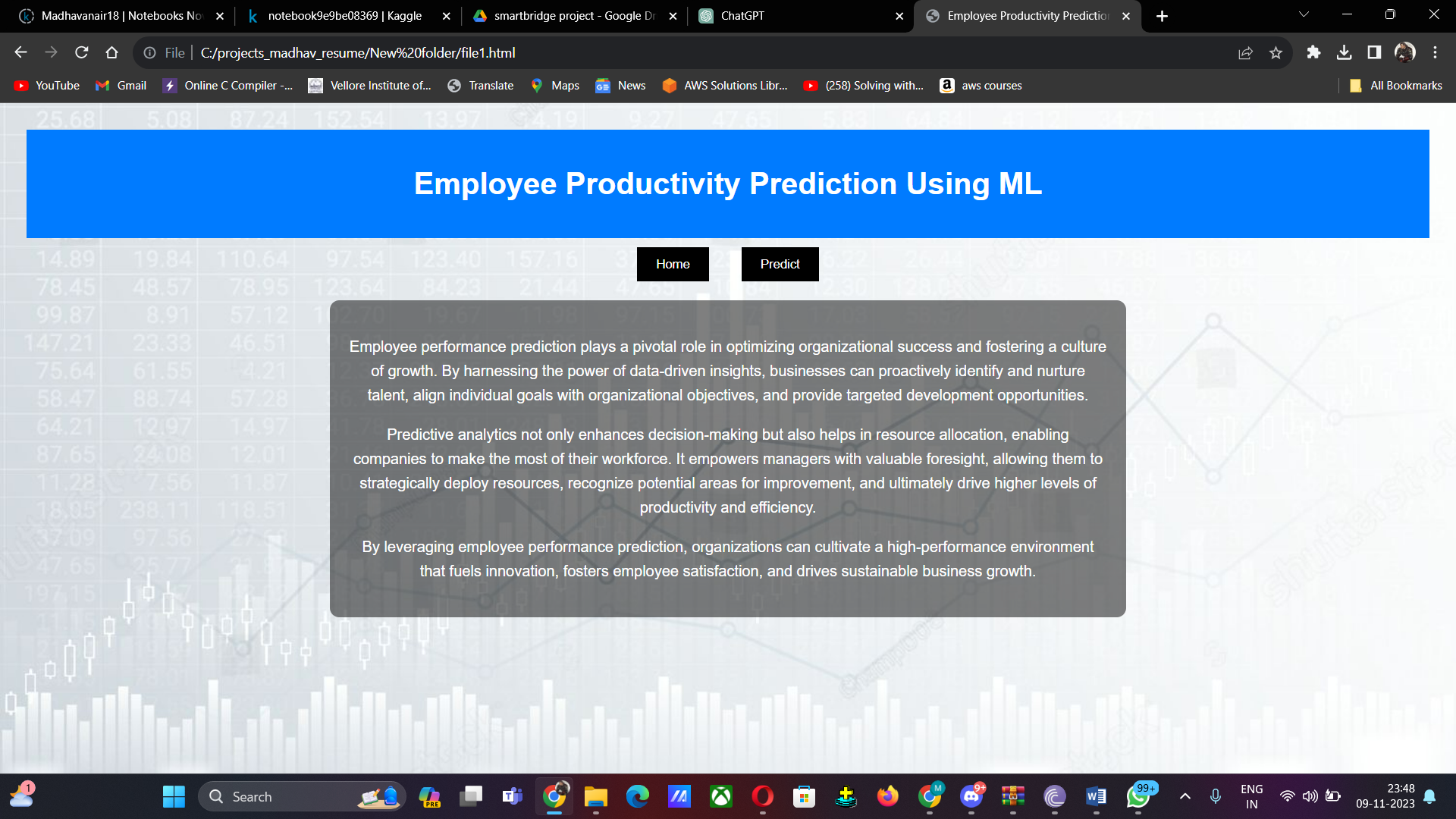
## Activity1: Building Html Pages:

For this project create three HTML files namely

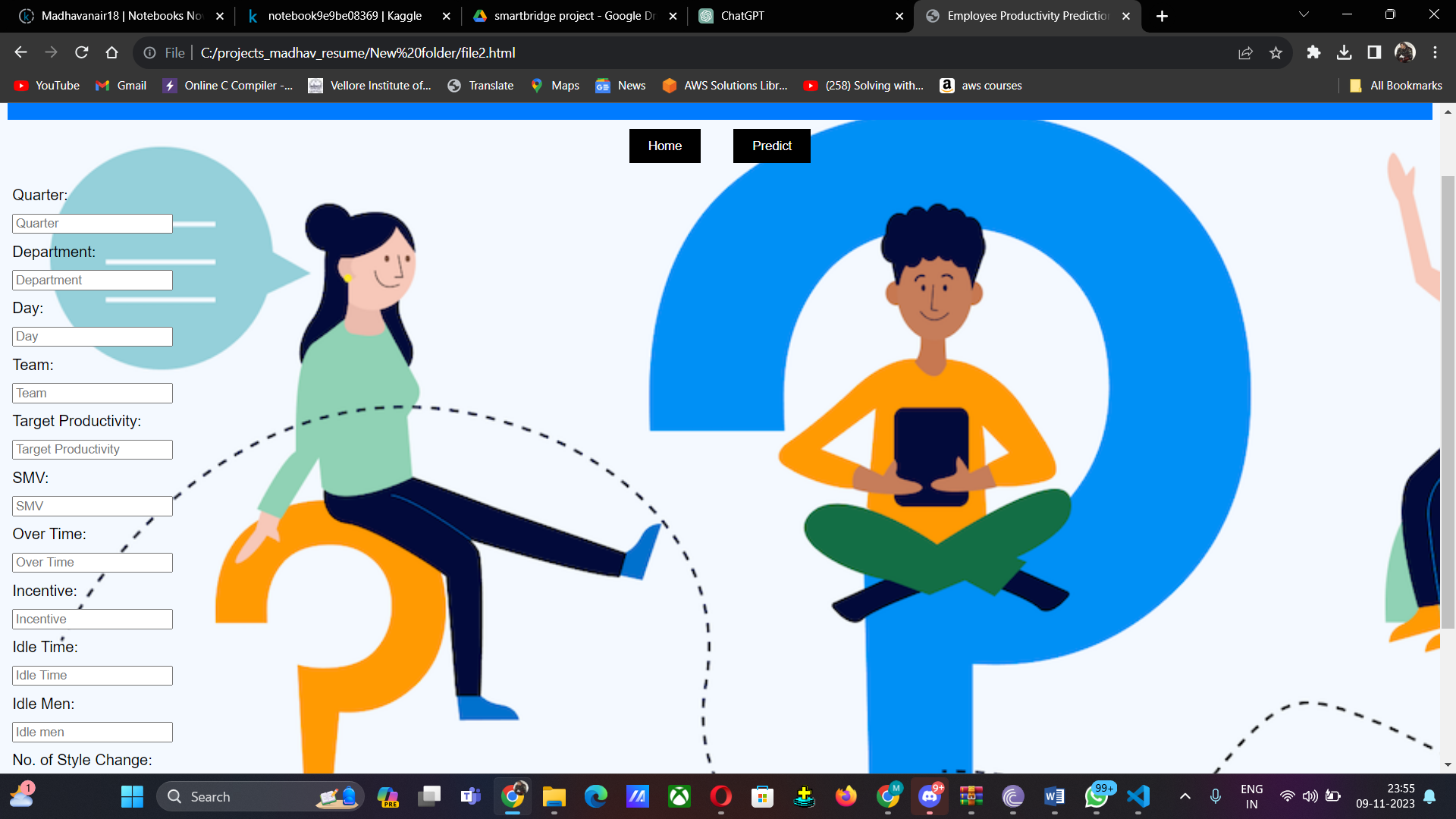
* about.html
* home.html
* predict.html
* submit.html

and save them in templates folder.

Let’s see how our home.html page looks like:



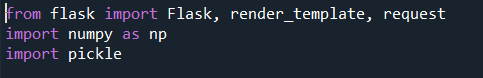
Now when you click on predict button from top right corner you will get redirected to predict.html Lets look how our predict.html file looks like:



Now when you click on submit button from left bottom corner you will get redirected to submit.html Lets look how our submit.html file looks like:

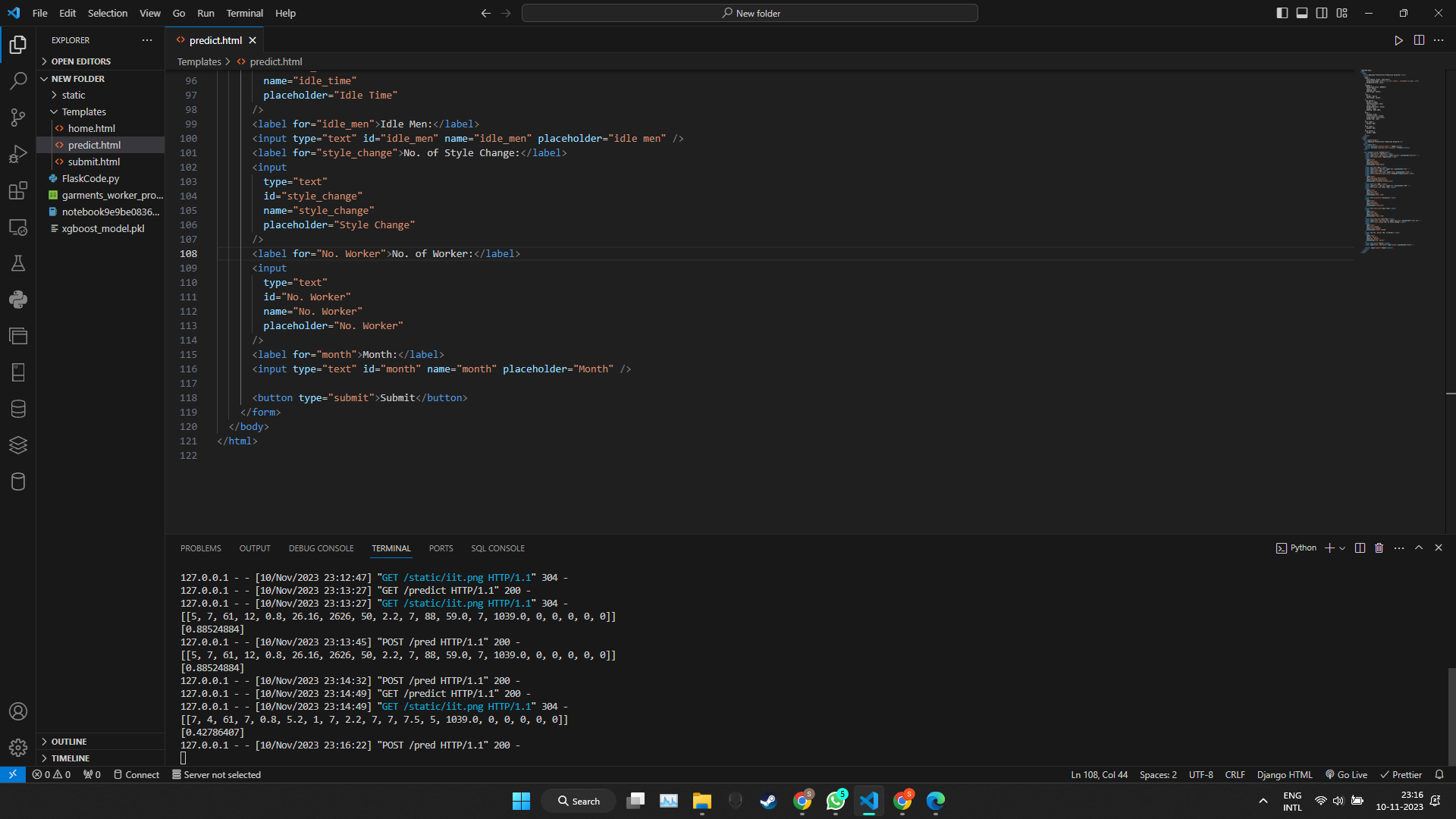
## Activity 2: Build Python code:

Import the libraries

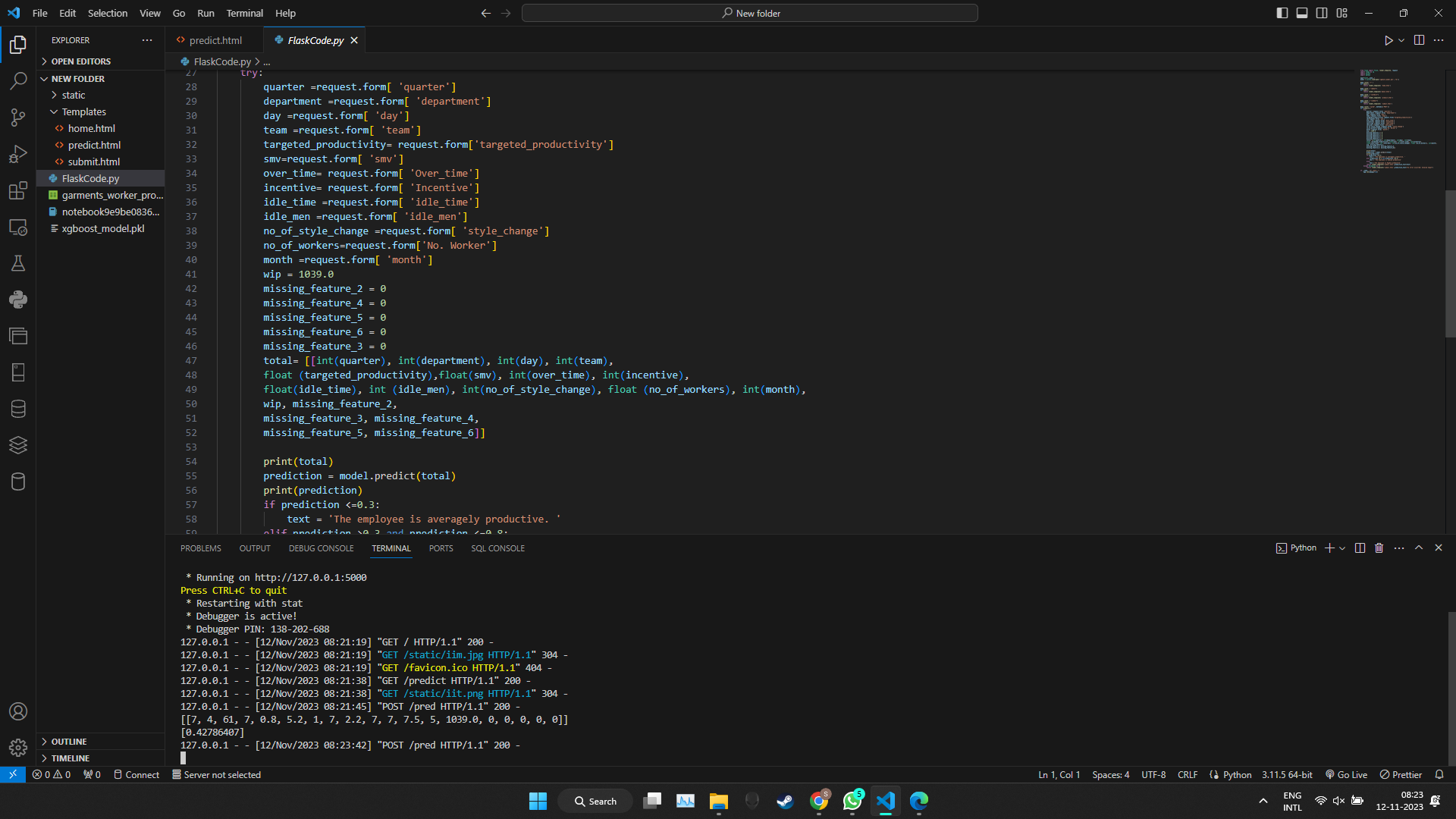


Load the saved model. Importing flask module in the project is mandatory. An object of Flask class is our WSGI application. Flask constructor takes the name of the current module ( name ) as argument

HTML page creation :

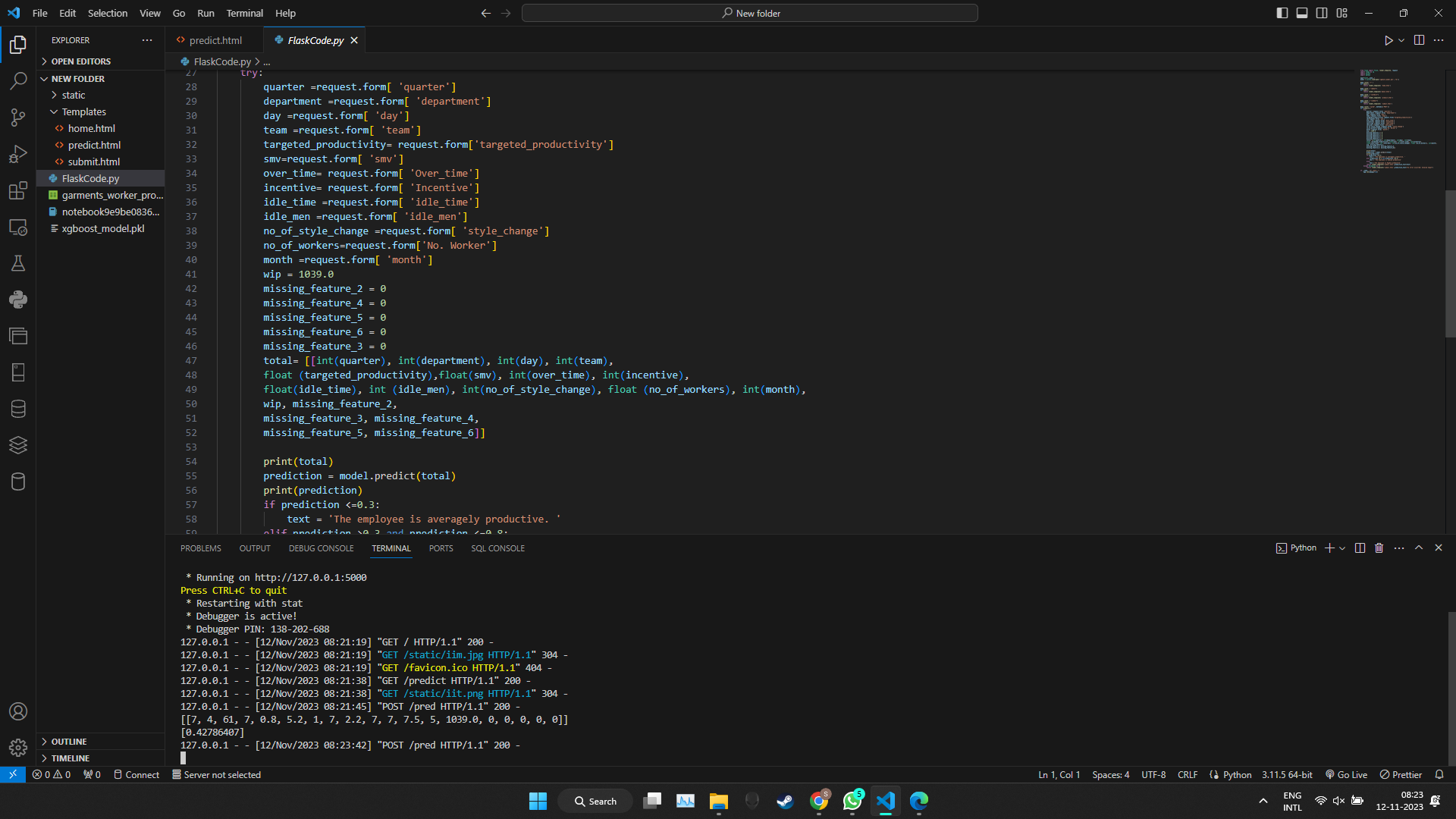


Flask app source code:



## 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 button from the top right corner, enter the inputs, click on the submit button, and see the result/prediction on the web.

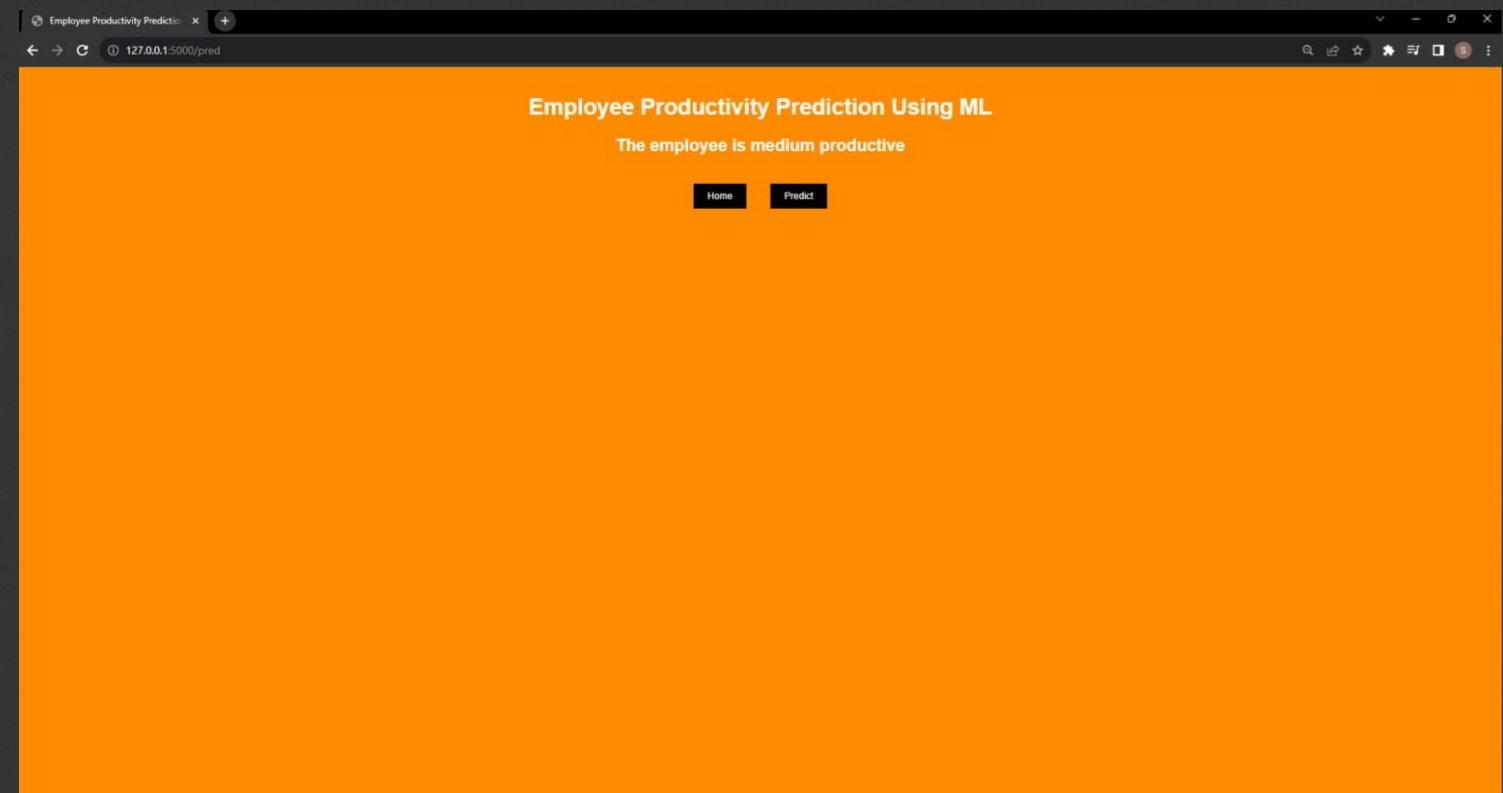


Input 2:

Input 1:



Output 1:





Output 2:

