

Text  Description automatically generated with low confidence

**SmartLearn Forecast: Machine Learning Models for Anticipating Academic Outcomes**

SmartInternz

[www.smartinternz.com](http://www.smartinternz.com/)

**SMART LEARN FORECAST: MACHINE LEARNING MODELS FOR ANTICIPATING ACADEMIC OUTCOMES**

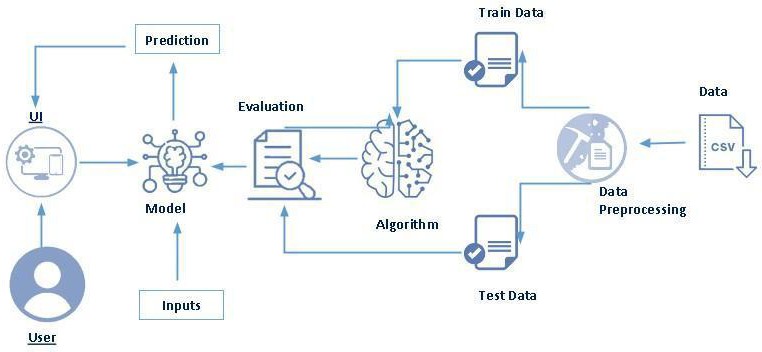
# Project Description:

The Smart Learn Forecast project is a cutting-edge initiative aimed at transforming the landscape of academic forecasting through the implementation of sophisticated machine learning models. This project is meticulously crafted to anticipate and categorize academic outcomes, empowering educational institutions to make informed decisions based on predictive analytics.

The dataset used in this project will be collected from various sources such as online surveys, social media platforms, and other publicly available data sources. The data will be pre-processed and cleaned to ensure quality and eliminate any noise or missing values. Once the data is cleaned, it will be split into training and testing sets. Several machine learning models will be built and evaluated on the training data to determine the best-performing model. The models to be explored include logistic regression, decision trees, k-nearest neighbors, support vector machines, random forests, and gradient boosting. After selecting the best-performing model, it will be used to predict the productivity status of the. The model's performance will be evaluated based on various metrics such as cross-validation score accuracy, precision, recall, and F1 score.

The project's output will be a web-based application that allows airline companies to predict passenger satisfaction and identify areas of improvement. The application will provide real-time insights to the airline companies, which can be used to improve their services and enhance passenger experience.

# 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 seaborn” 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 Flask” and click enter.

**Project Flow:**

* User interacts with the UI to enter the input.
* Entered input is analyzed by the model which is integrated.
* Once the model analyses the input the prediction is showcased on the UI To accomplish this, we have to complete all the activities listed below,
* Define Problem / Problem Understanding
  + Specify the business problem
  + Business requirements
  + Literature Survey
  + Social or Business Impact.
* Data Collection & Preparation
  + Collect the dataset
  + Data Preparation
* Exploratory Data Analysis
  + Descriptive statistical
  + Visual Analysis
* Model Building
  + Training the model in multiple algorithms
  + Testing the model
* Performance Testing & Hyperparameter Tuning
  + Testing model with multiple evaluation metrics
  + Comparing model accuracy before & after applying hyperparameter tuning
* Model Deployment
  + Save the best model
  + Integrate with Web Framework
* Project Demonstration & Documentation
  + Record explanation Video for project end-to-end solution
  + Project Documentation-Step by step-by-step project development procedure

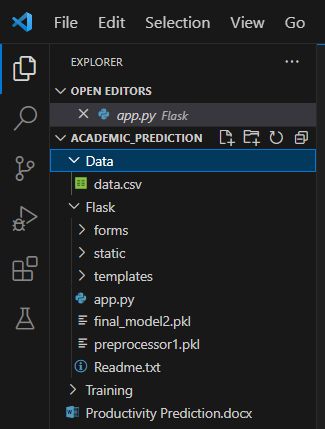
# Prior Knowledge:

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

* ML Concepts
* Supervised learning: <https://www.javatpoint.com/supervised-machine-learning>
* Decision tree: [https://www.javatpoint.com/machine-learning-decision-tree-classification-](https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm) [algorithm](https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm)
* Random forest: <https://www.javatpoint.com/machine-learning-random-forest-algorithm>
* KNN: <https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning>
* Logistic regression: https://www.javatpoint.com/logistic-regression-in-machine-learning
* Support Vector Machine: https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm
* Gradient Boosting: https://www.geeksforgeeks.org/ml-gradient-boosting/
* Naïve Bayes: https://www.geeksforgeeks.org/naive-bayes-classifiers/
* Evaluation metrics: [https://www.analyticsvidhya.com/blog/2019/08/11-important-model-](https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/) [evaluation-error-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 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.
* Final\_model1.pkl will be our saved model. Further we will use this model for flask integration.
* Training folder contains a model training file.

# Milestone 1: Define Problem / Problem Understanding

## Activity 1: Specify the business problem

Refer Project Description

## Activity 2: Business requirements

## The business requirements for the Smart Learn Forecast model aim to revolutionize educational planning and student support systems. The key objectives include:

## Personalized Academic Planning:

## Predicting academic outcomes to facilitate personalized academic planning for students.

## Providing insights into individual learning paths and recommending suitable courses based on predicted outcomes.

## Early Intervention Strategies:

## Identifying students at risk of academic challenges early on to implement targeted intervention strategies.

## Enhancing student success rates through timely support and guidance.

## Activity 3: Literature Survey (Student Will Write)

## The literature survey for the Smart Learn Forecast project involves an in-depth exploration of existing studies and research in the field of academic outcome prediction. Key aspects of the survey include:

## Review of Predictive Models:

## Analyzing various predictive models used in academic outcome prediction, understanding their strengths, limitations, and applicability in diverse educational contexts.

## Data Sources and Features:

## Investigating the sources of data used in academic prediction models and identifying the key features considered in predicting student outcomes.

## Activity 4: Social and Business Impact.

## Social Impact

## Empowering Students:

## Optimizing academic planning fosters a positive learning environment, empowering students to make informed decisions about their educational journey.

## Diversity and Inclusion:

## Early identification of potential challenges supports a more inclusive educational system, addressing diverse learning needs and promoting equal opportunities.

## Business Impact:

## Enhanced Institutional Reputation:

## Implementation of advanced predictive models contributes to an institution's reputation for innovation and commitment to student success.

## Resource Optimization:

## Efficient resource allocation based on predicted outcomes ensures optimal utilization of educational resources.

## Decision Support:

## Data-driven decision-making supports educational leaders in addressing challenges promptly and adapting strategies for continuous improvement.

## Top of Form

# Milestone 2: Data Collection & Preparation

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

## Activity 1: Collect the dataset

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

In this project we have used .csv data. This data is downloaded from kaggle.com. Please refer to the link given below to download the dataset.

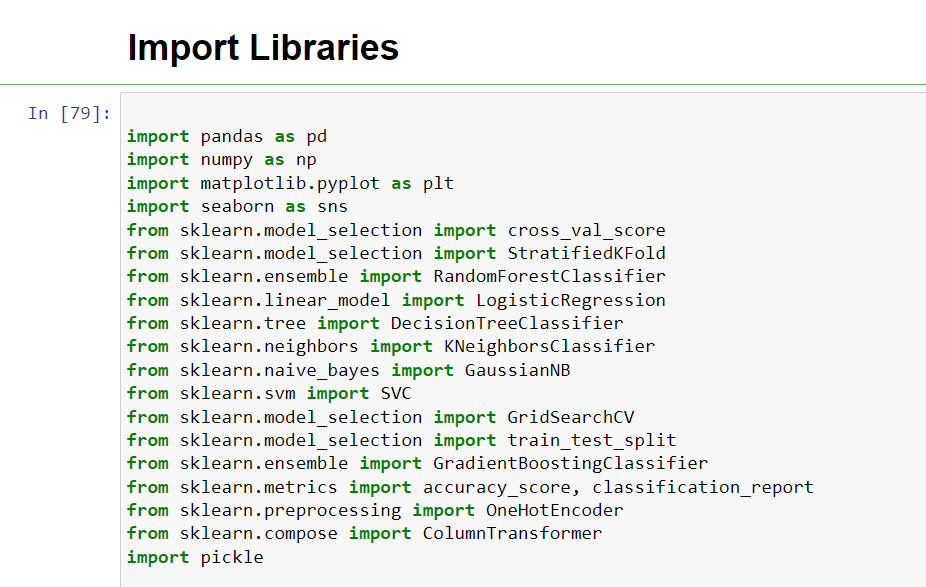
Link: <https://www.kaggle.com/code/abhisri03/garment-regression/input>

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

**Note:** There are a 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 1.1: Importing the libraries

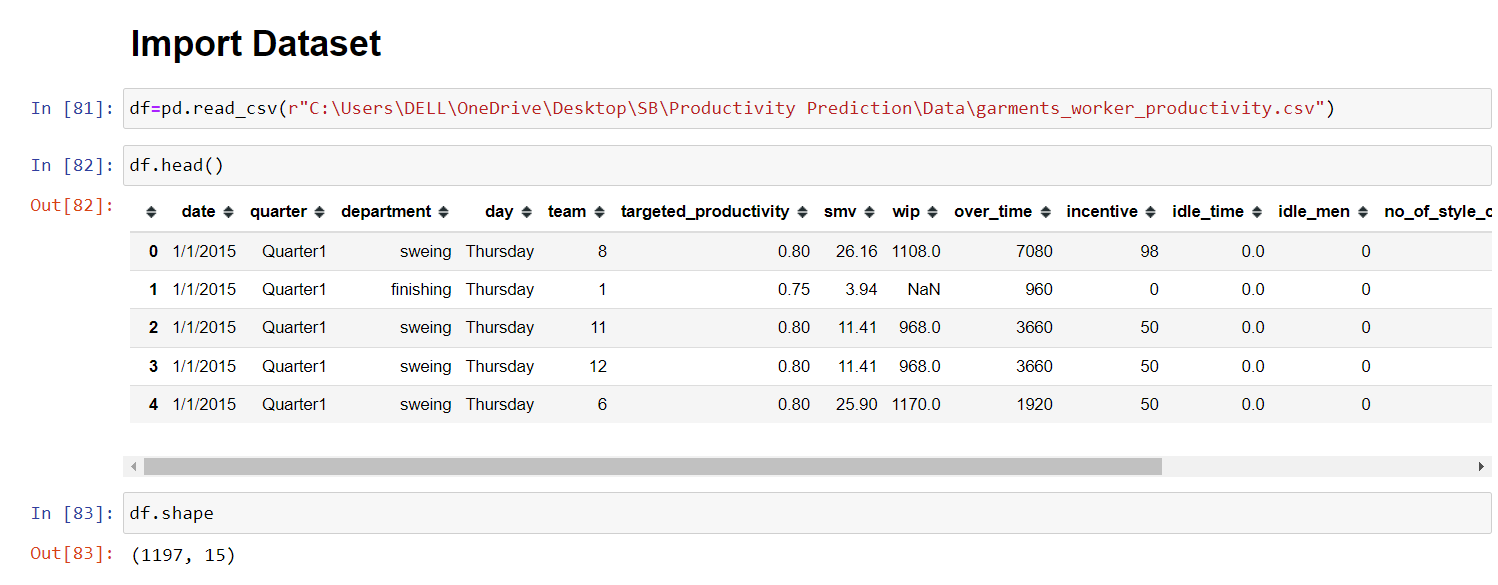
Import the necessary libraries as shown in the image.



## Activity 1.2: 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 the csv file.



## Activity 2: Data Preparation

As we have understood how the data is, let's pre-process the collected data.

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

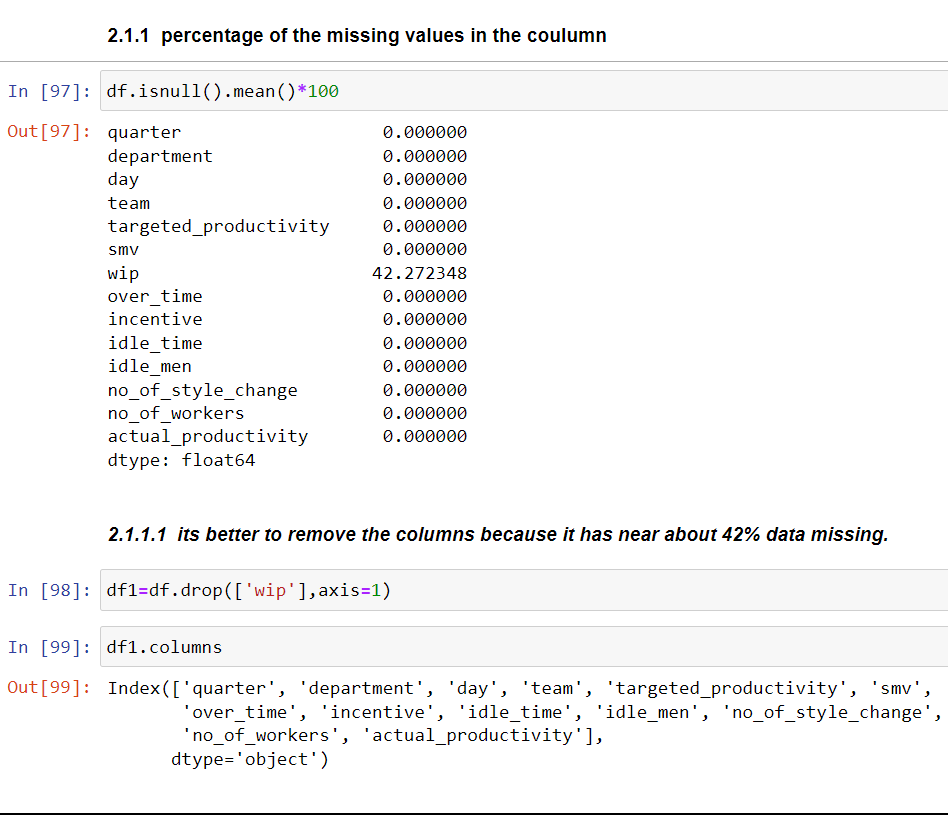
* Handling missing values
* Handling categorical data
* Handling Outliers

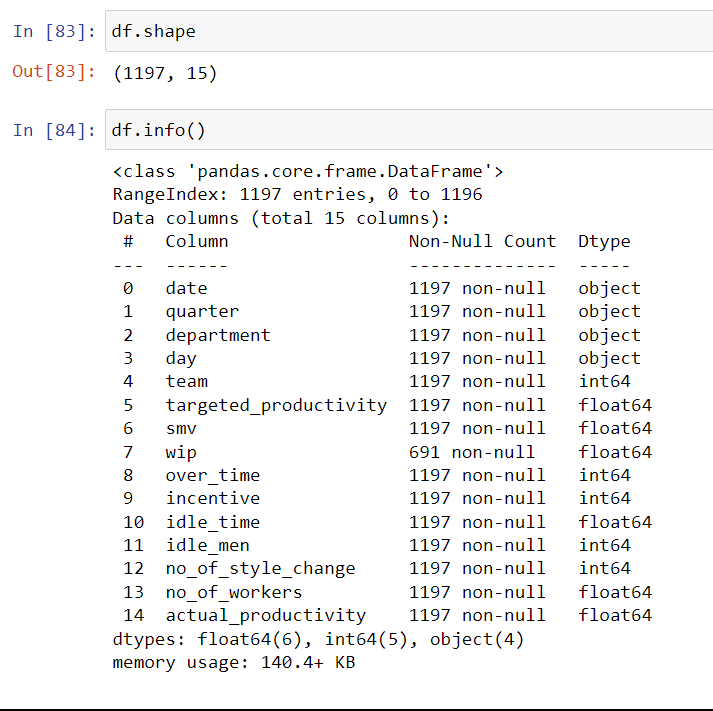
Note: These are the general steps of pre-processing the data before using it for machine learning. Depending on the condition of your dataset, you may or may not have to go through all these steps.

## Activity 2.1: Handling missing values

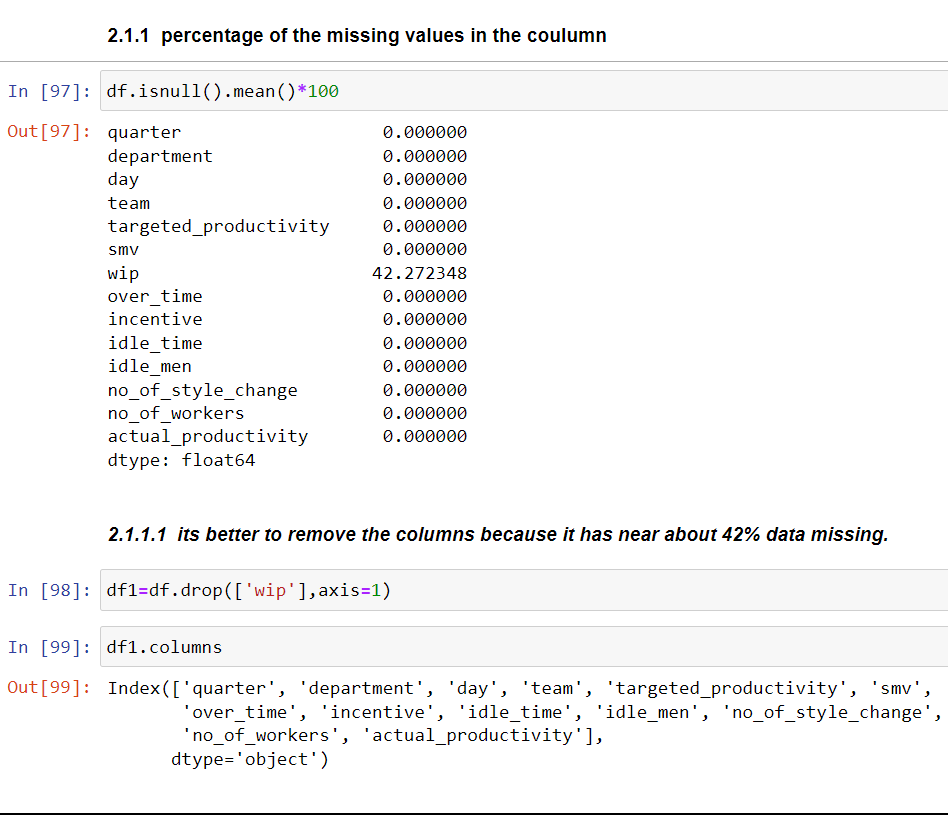
* Let’s find the shape of our dataset first. To find the shape of our data , To check percentage of missing values present in the dataset we will use df.isnull().mean()\*100
* the df.shape method.

To find the data type, df.info() function is used.





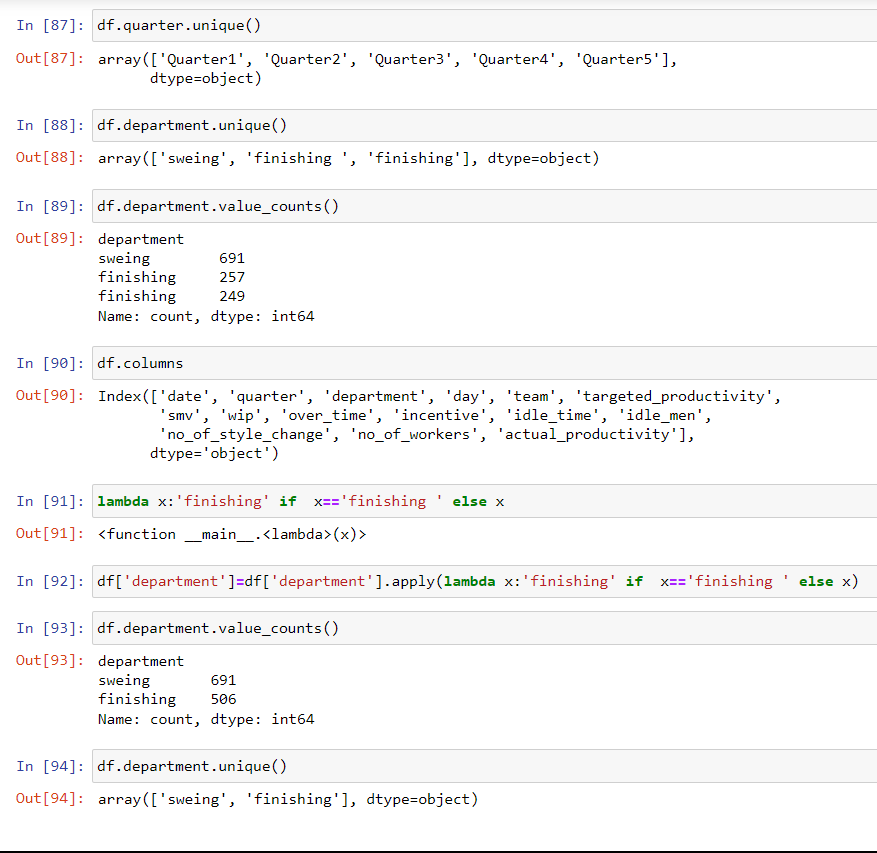
* + For checking the null values, df.isnull() function is used. To sum those null values we use .sum() function. From the below image we found that there are no null values present in our dataset. So we can skip handling the missing values step.



## Activity 2.2: Handling Categorical Values

As we can see our dataset has categorical data we must convert the categorical data to integer encoding or binary encoding.

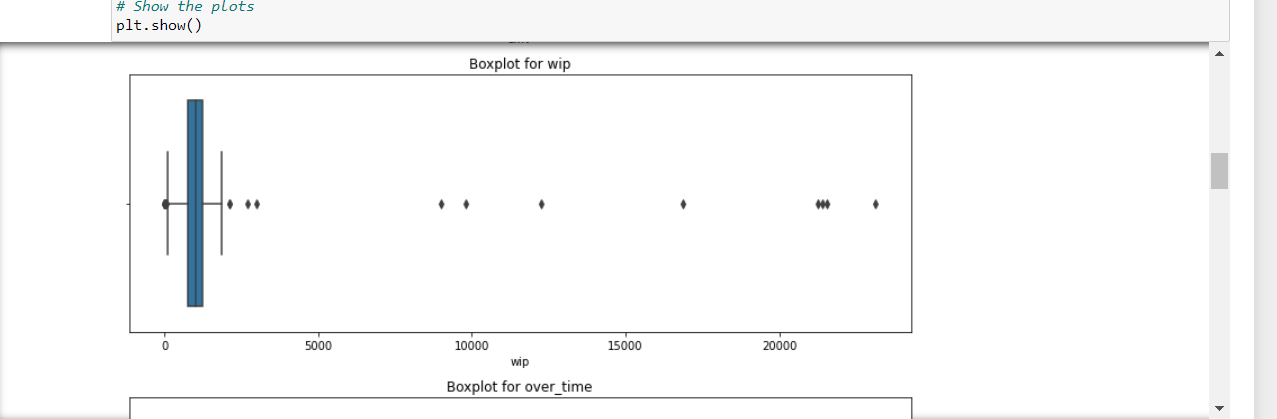
To convert the categorical features into numerical features we use encoding techniques. There are several techniques but in our project we are using manual encoding with the help of list comprehension.

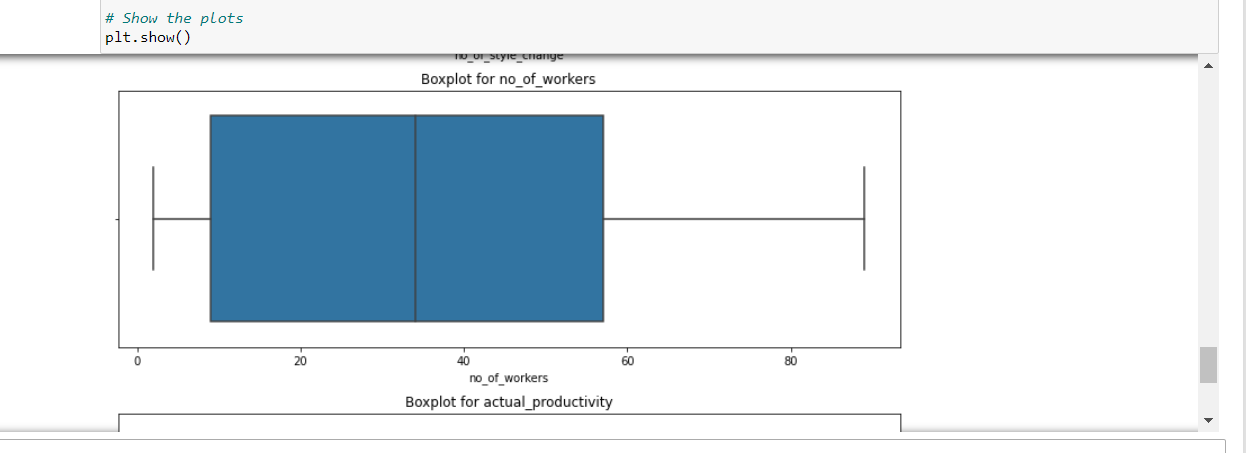


## Activity 2.3: Handling Outliers

With the help of boxplot, outliers are visualized. And here we are going to find upper bound and lower bound of every column.



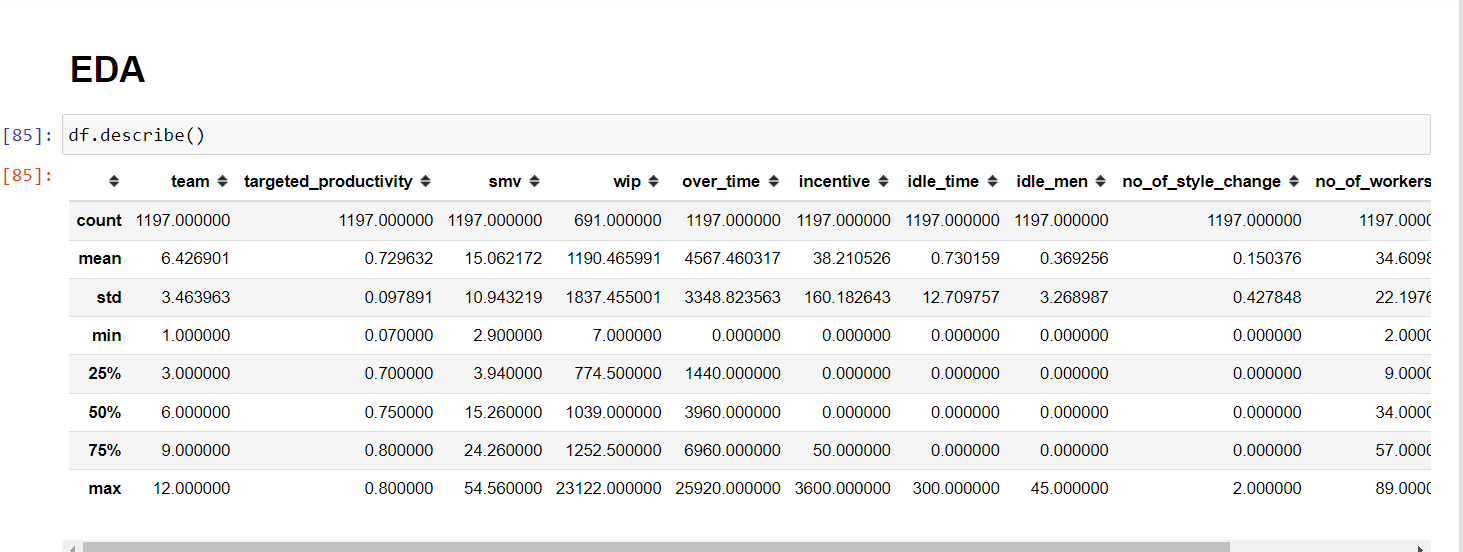




# Milestone 3: Exploratory Data Analysis

## Activity 1: Descriptive statistical

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.



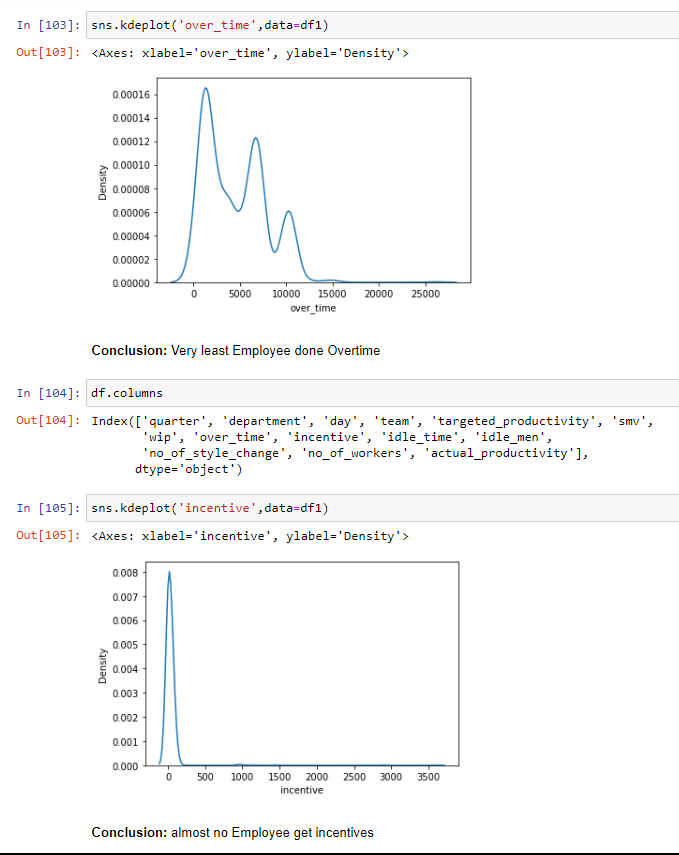
## Activity 2: Visual analysis

Visual analysis is the process of using visual representations, such as charts, plots, and graphs, to explore and understand data. It is a way to quickly identify patterns, trends, and outliers in the data, which can help to gain insights and make informed decisions.

## Activity 2.1: Univariate analysis

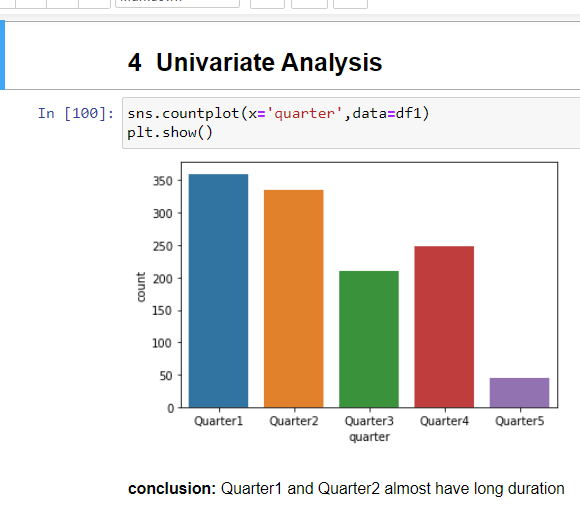
In simple words, univariate analysis is understanding the data with single feature. Here we have displayed two different graphs such as kdeplot(density plot) and countplot.

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



In our dataset we have some categorical features. With the countplot function, we are going to count the unique category in those features. We have created a dummy data frame with categorical features. With for loop and subplot we have plotted this below graph.

From the plot we came to know, Most of the passengers are in neutral or dissatisfied category

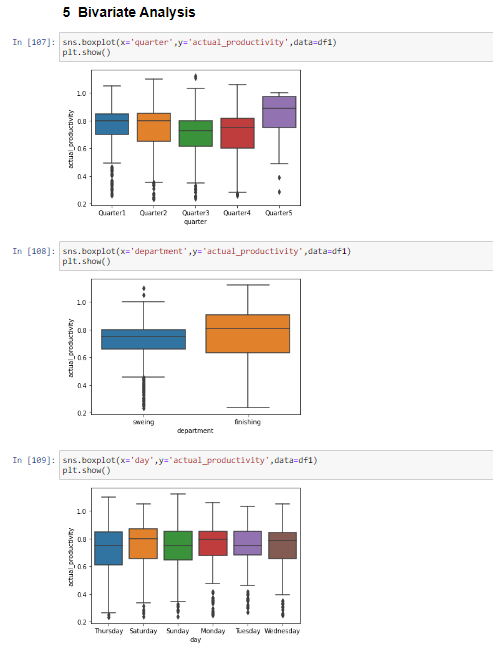




**Activity 2.2: Bivariate Analysis**

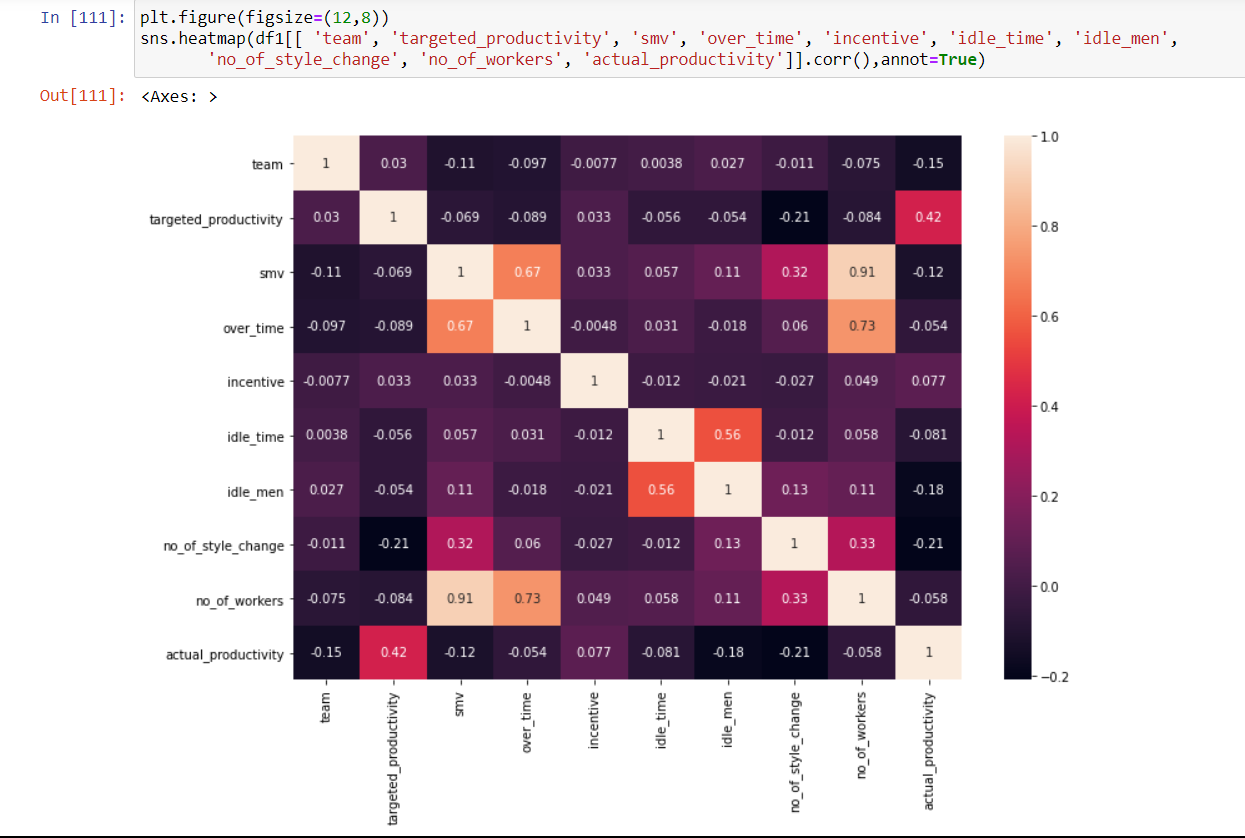
To find the relation between two features we use bivariate analysis. Categorical variables of the model

With the target variable actual\_productivity



## Activity 2.3: Multivariate analysis

In simple words, multivariate analysis is to find the relation between multiple features. Here we have used heatmap from package seaborn.

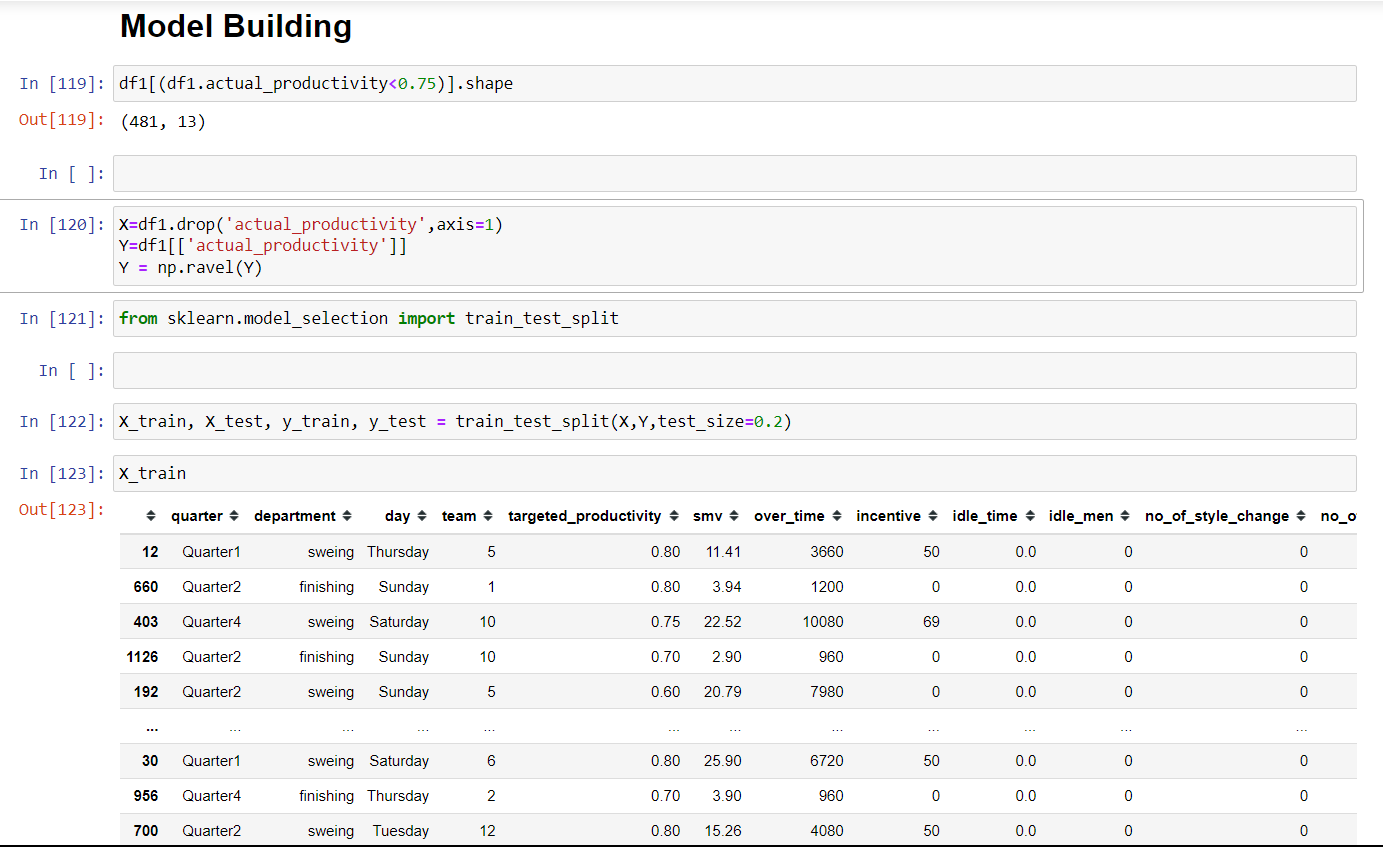


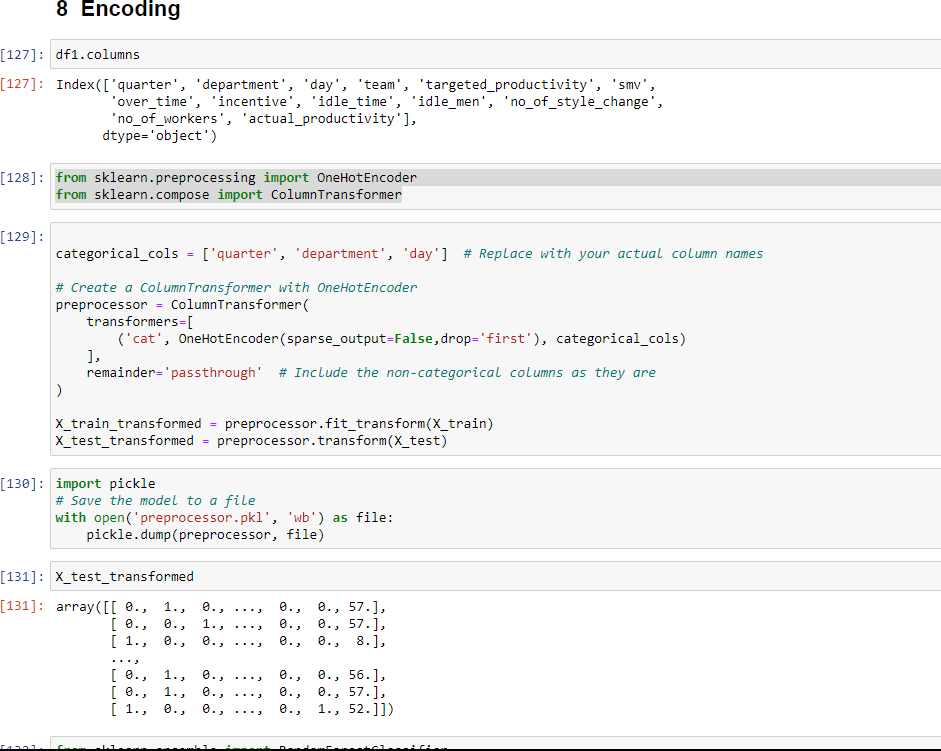
**Splitting data into train and test**

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.







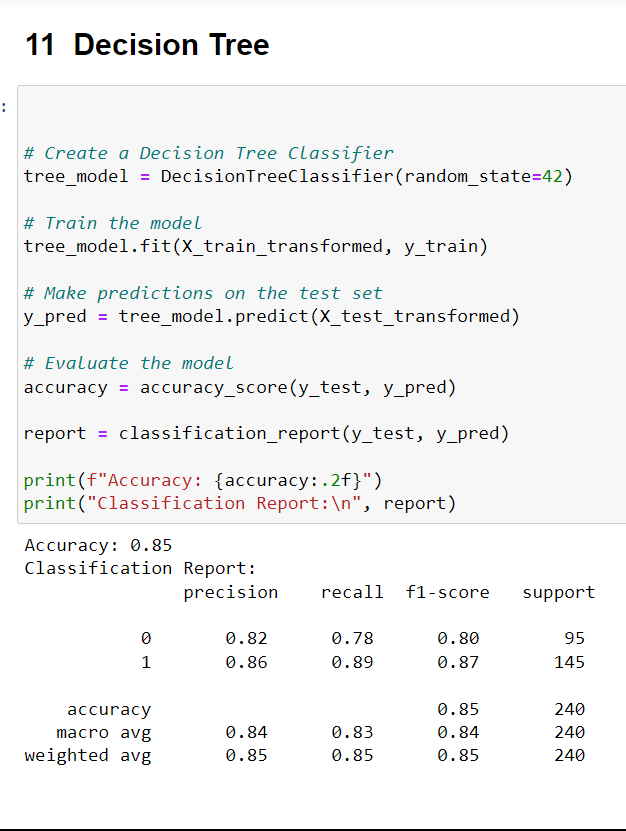
# Milestone 4: Model Building

## Activity 1: Training the model in multiple algorithms

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 four classification algorithms. The best model is saved based on its performance.

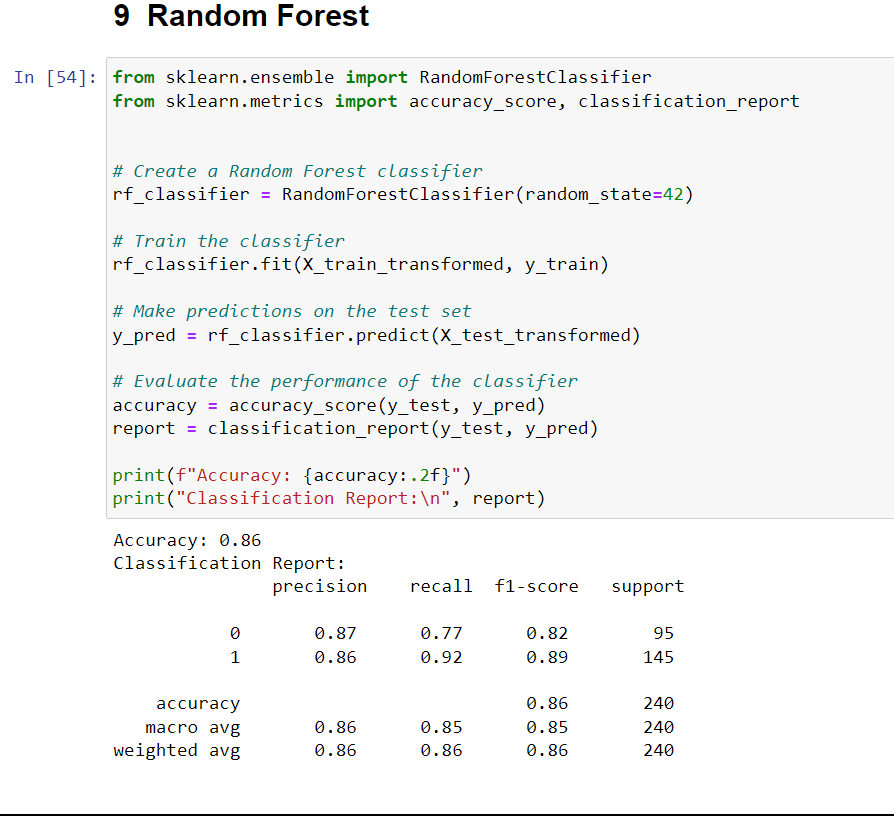
## Activity 1.1: Decision tree model

A function named decisionTree is created and train and test data are passed as the parameters. Inside the function, DecisionTreeClassifier algorithm is initialized and training data is passed to the model with the .fit() function. Test data is predicted with .predict() function and saved in a new variable. For evaluating the model, a confusion matrix and classification report is done.



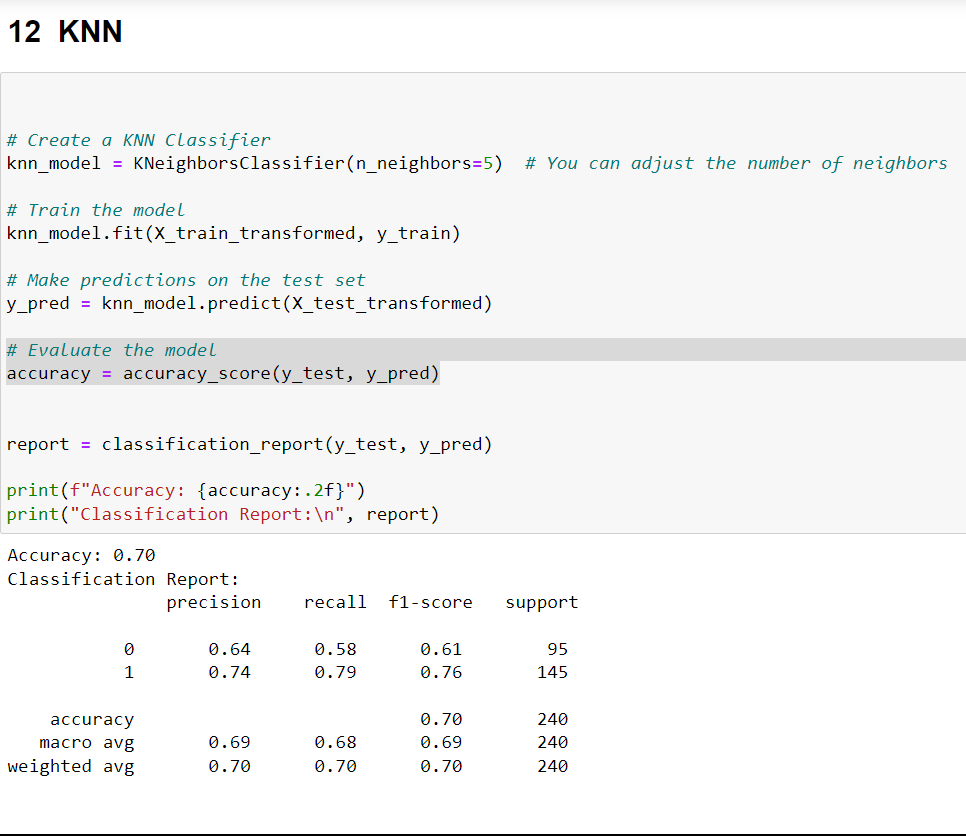
## Activity 1.2: Random forest model

A function named randomForest is created and train and test data are passed as the parameters. Inside the function, RandomForestClassifier algorithm is initialised and training data is passed to the model with .fit() function. Test data is predicted with predict() function and saved in a new variable. For evaluating the model, a confusion matrix and classification report is done.



## Activity 1.3: KNN model

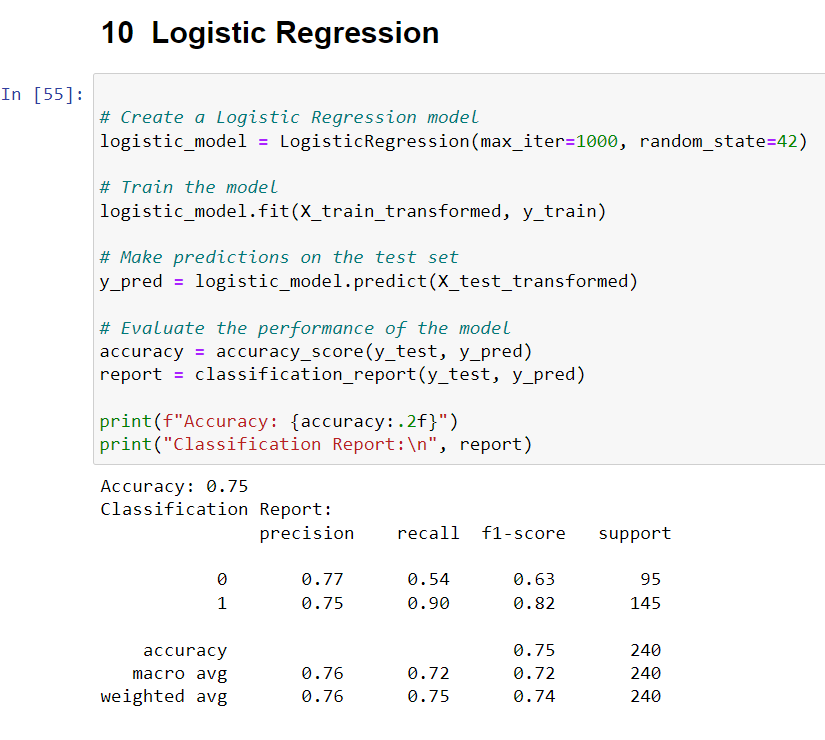
A function named KNN is created and train and test data are passed as the parameters. Inside the function, KNeighborsClassifier algorithm is initialized and training data is passed to the model with .fit() function. Test data is predicted with .predict() function and saved in new variable. For evaluating the model, confusion matrix and classification report is done.



4

## Activity 1.4: Logistic Regression

A function named Logistic regression is created and train and test data are passed as the parameters. Inside the function, Logistic Regression Classifier algorithm is initialized and training data is passed to the model with .fit () function. Test data is predicted with .Predict () function and saved in new variable. For evaluating the model, confusion matrix and classification report is done.



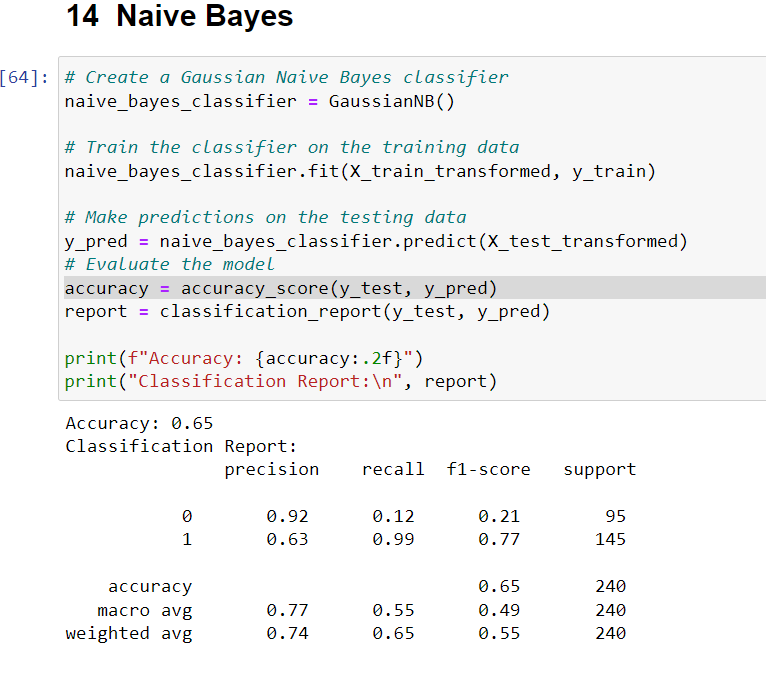
## Activity 1.5 Support Vector Machine

A function named SVC() is created and train and test data are passed as the parameters. Inside the function, Logistic Regression Classifier algorithm is initialized and training data is passed to the model with .fit () function. Test data is predicted with .Predict () function and saved in new variable. For evaluating the model, confusion matrix and classification report is done.

## 

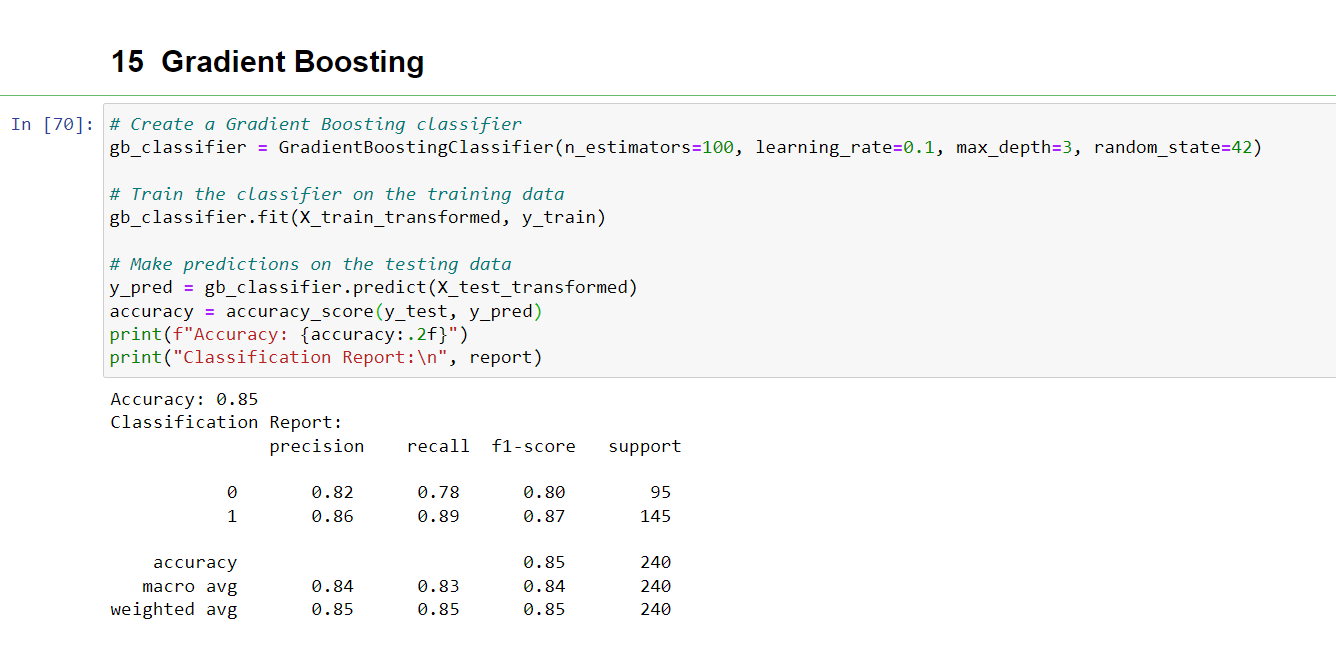
## Activity 1.6 Naïve Bayes

A function named GaussianNB() is created and train and test data are passed as the parameters. Inside the function, Logistic Regression Classifier algorithm is initialized and training data is passed to the model with .fit () function. Test data is predicted with .Predict () function and saved in new variable. For evaluating the model, confusion matrix and classification report is done.



## Activity 1.7 Gradient Boosting

A function named GaussianNB() is created and train and test data are passed as the parameters. Inside the function, Logistic Regression Classifier algorithm is initialized and training data is passed to the model with .fit () function. Test data is predicted with .Predict () function and saved in new variable. For evaluating the model, confusion matrix and classification report is done.



# Milestone 5: Performance Testing & Hyparameter Tuning

## Activity 1: Testing model with multiple evaluation metrics

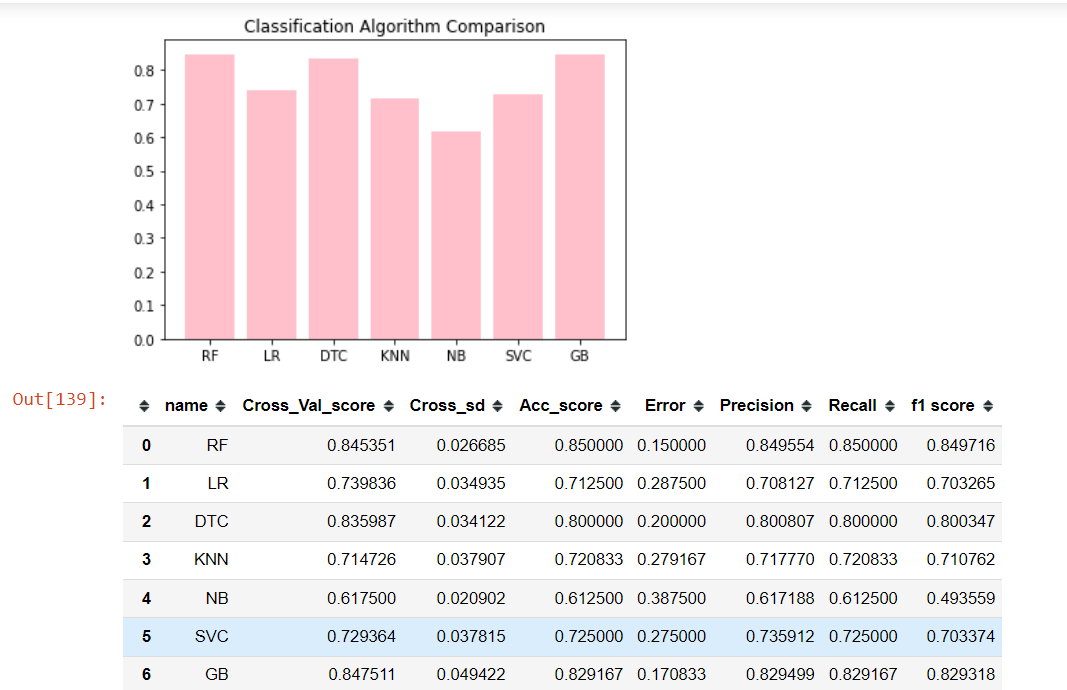
Multiple evaluation metrics means evaluating the model's performance on a test set using different performance measures. This can provide a more comprehensive understanding of the model's strengths and weaknesses. We are using evaluation metrics for classification tasks including accuracy, precision, recall, support and F1-score.

## Activity 1.1: Compare the model

For comparing the above four models, the compareModel function is defined.

After calling the function, the results of the models are displayed as output. From the above model random forest classifier is performing well. From the below image, we can see the accuracy of the models. Even the confusion matrix also has nearly the same results, here random forest is selected and evaluated with cross-validation. Additionally, we can tune the model with hyperparameter tuning techniques.

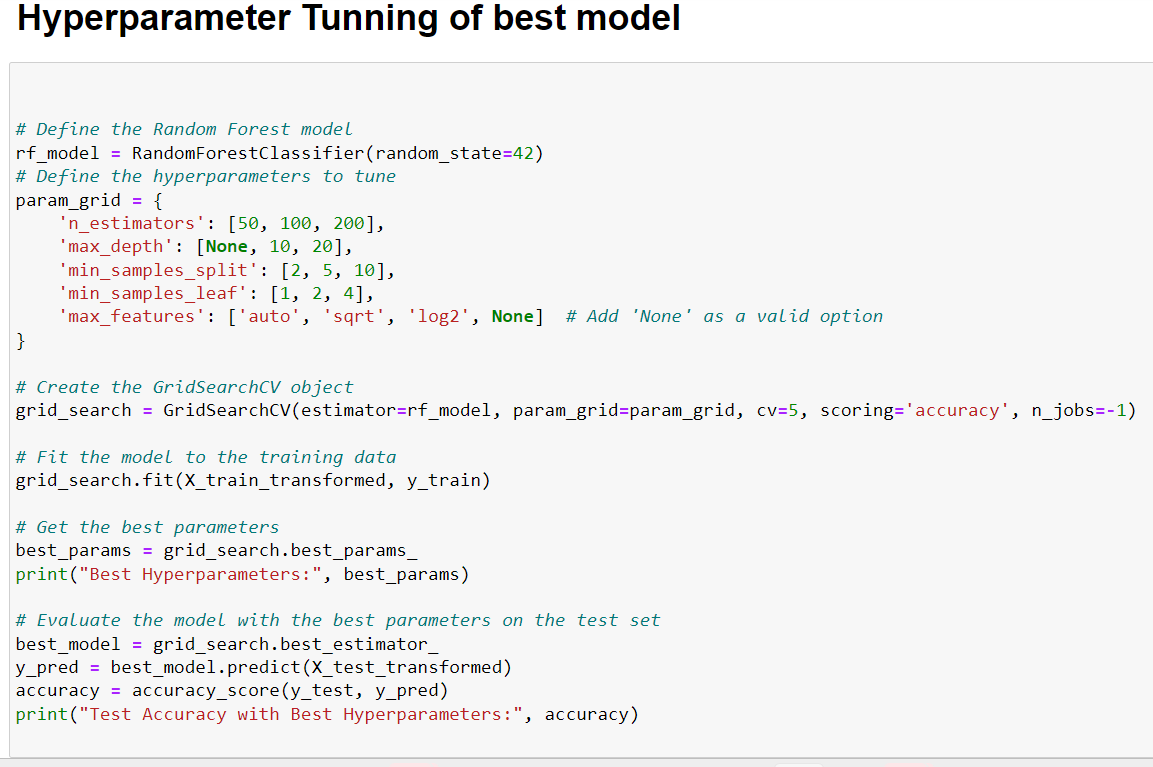


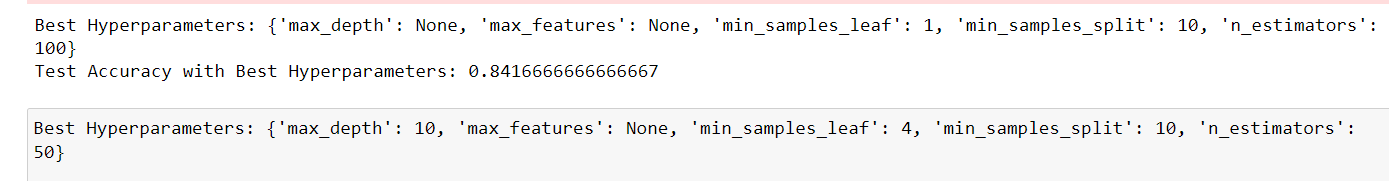


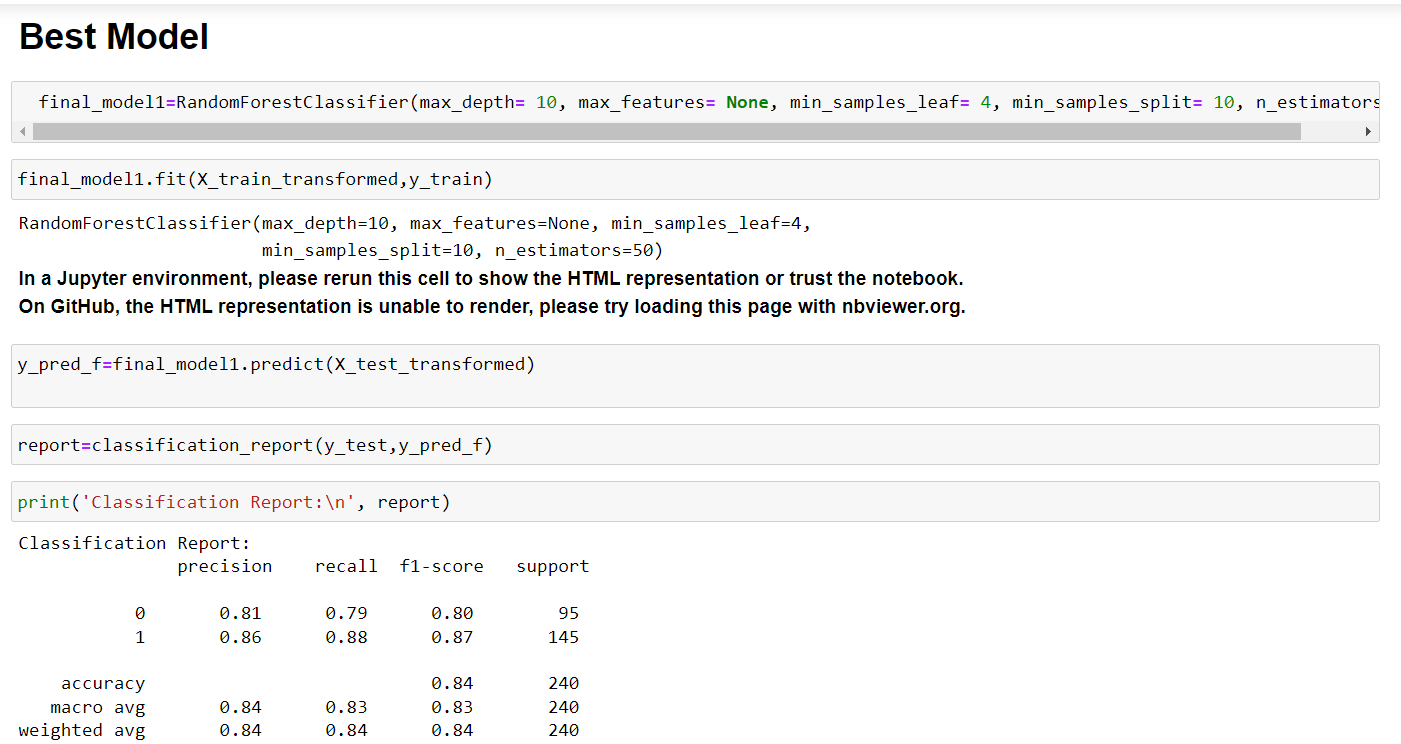
After calling the function, the results of models are displayed as output. From the all models random forest is performing well

## Activity 2: Comparing model accuracy before & after applying hyperparameter tuning.

Evaluating performance of the model with Hyper parameter tuning







# Milestone 6: Model Deployment

## Activity 1: Save the best model

Saving the best model after comparing its performance using different evaluation metrics means selecting the model with the highest performance and saving its weights and configuration. This can be useful in avoiding the need to retrain the model every time it is needed and also to be able to use it in the future.



## Activity 2: Integrate with Web Framework

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
* Run the web application

## Activity 2.1: Building Html Pages:

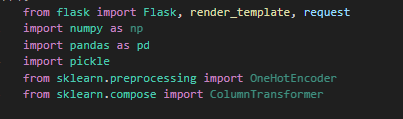
For this project create HTML files namely

* index.html
* inner-page.html
* portfolio-details.html
* blog-single.html
* portfolio-details.html

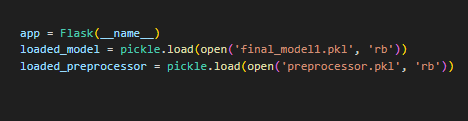
and save them in the templates folder. Refer this [link](https://drive.google.com/drive/folders/1K-C2uvRstV8x6bwsOY_BWBvXdzjbbfPF?usp=share_link) for templates.

## Activity 2.2: Build Python code:

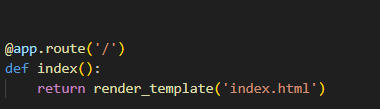
Import the libraries



Load the saved model. Importing the 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 a declared constructor to route to the HTML page which we have created earlier.

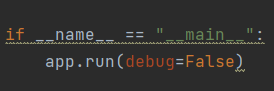
In the above example, ‘/’ URL is bound with the home.html function. Hence, when the home page of the web server is opened in the browser, the html page will be rendered. Whenever you enter the values from the html page the values can be retrieved using POST Method.

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 be rendered to the text that we have mentioned in the submit.html page earlier.

Main Function:



## Activity 2.3: Run the web application

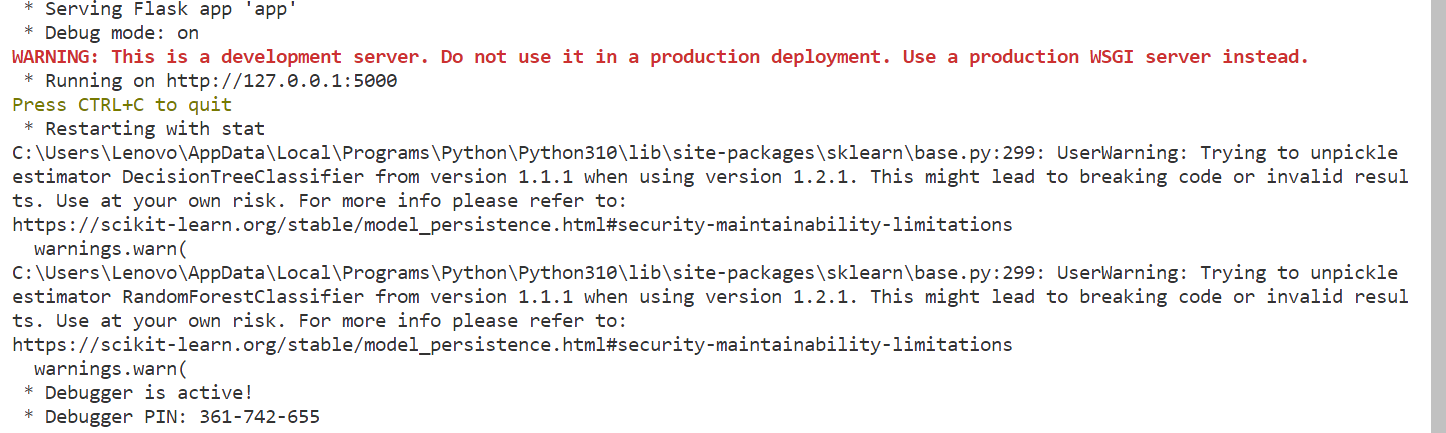
Open anaconda prompt from the start menu

Navigate to the folder where your python script is.

Now type “python app.py” command

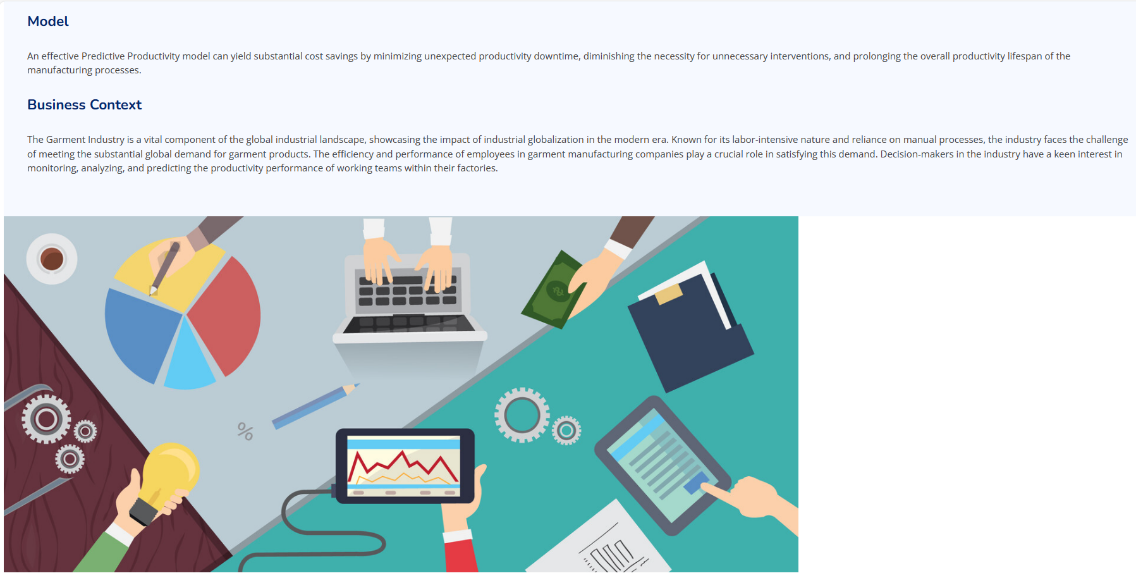
Navigate to the localhost where you can view your web page.

Click on the predict button from the top left corner, enter the inputs, click on the submit button, and see the result/prediction on the web.

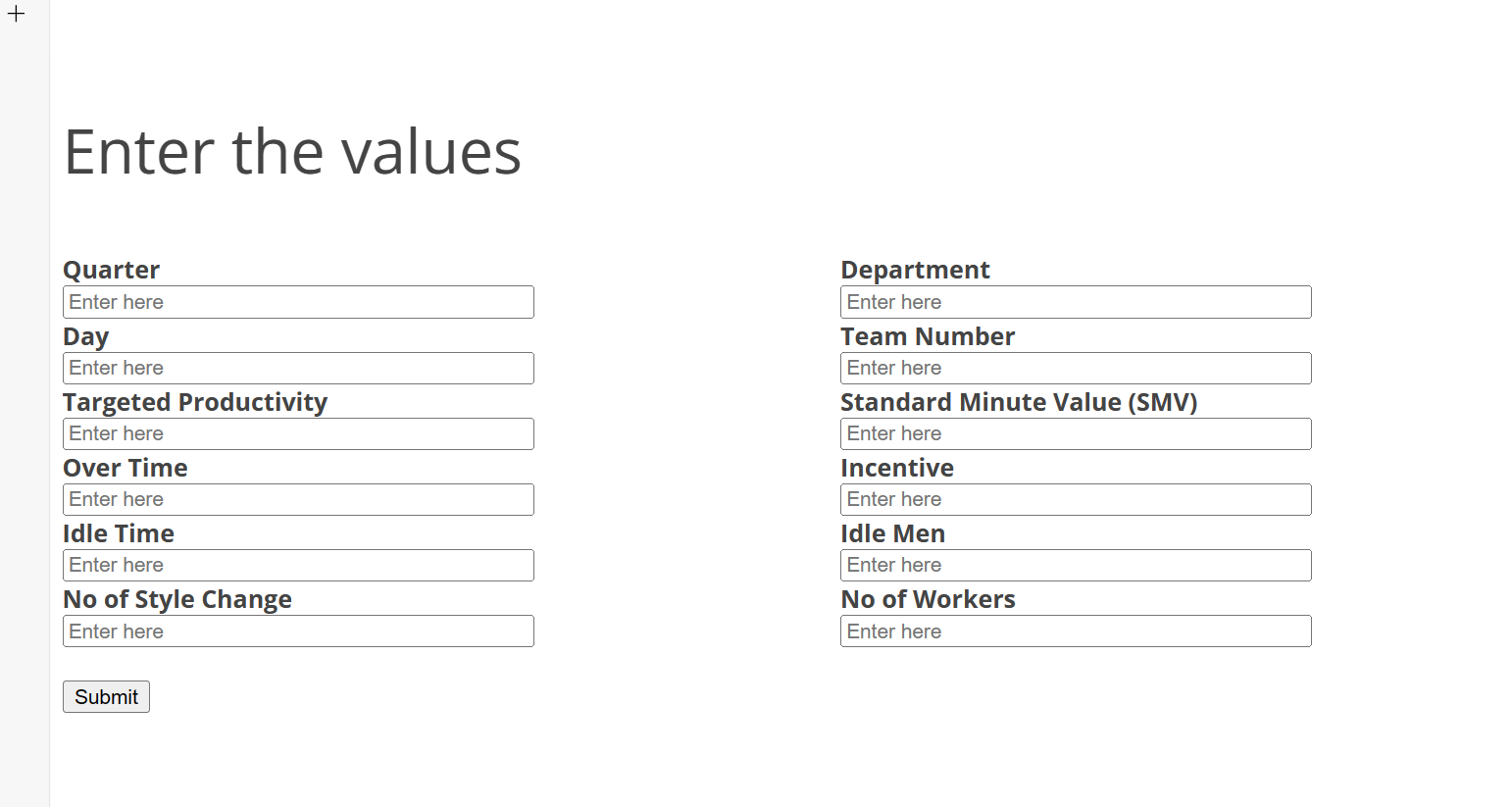


Now, Go to the web browser and write the localhost url (http://127.0.0.1:5000) to get the below result



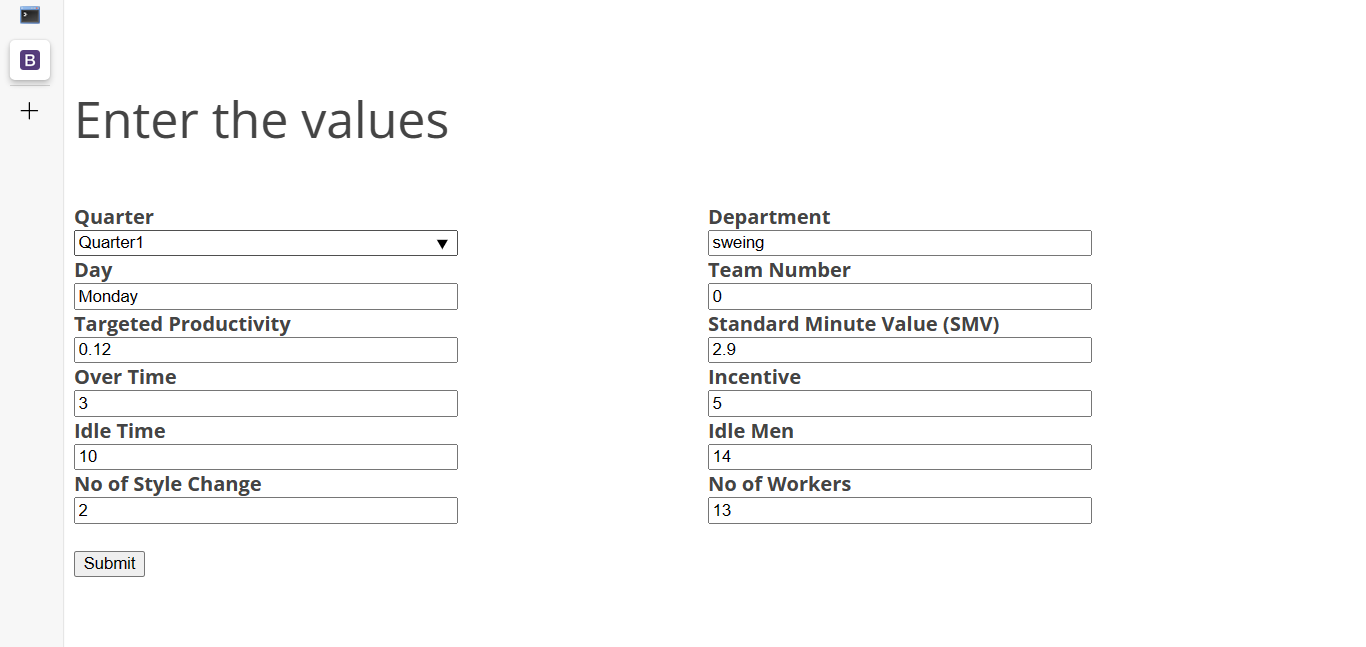


Now when you click on predict button from top right corner you will get redirected to predict.html



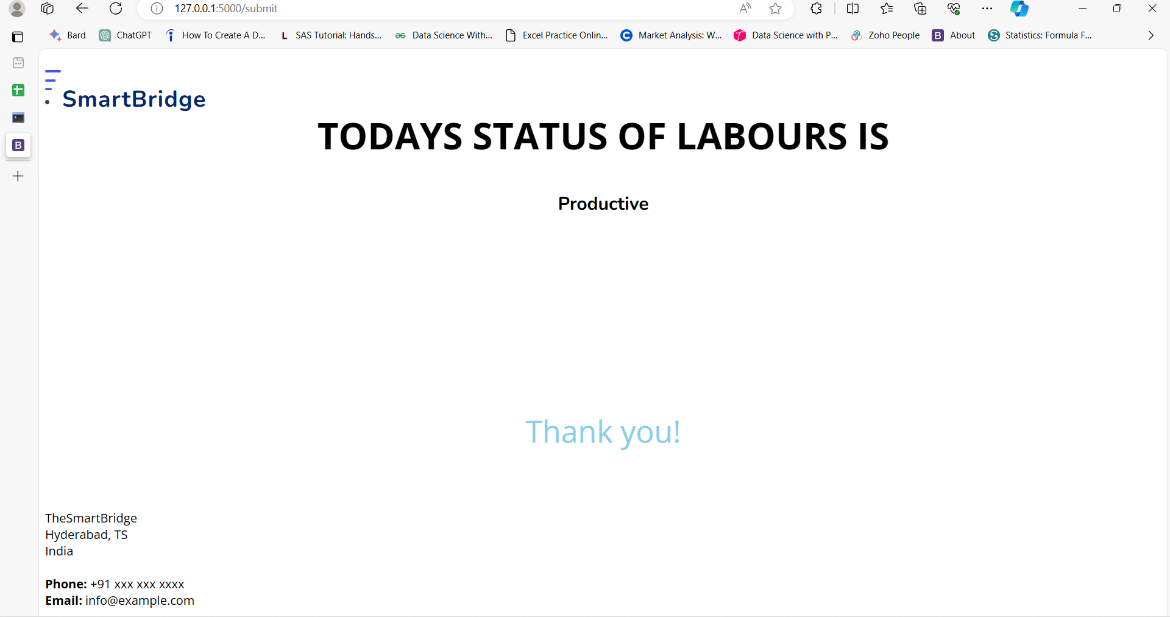
Now when you click on the ‘ Productivity Prediction’ button from the top console bar you will get redirected to the prediction page

**Input 1:**

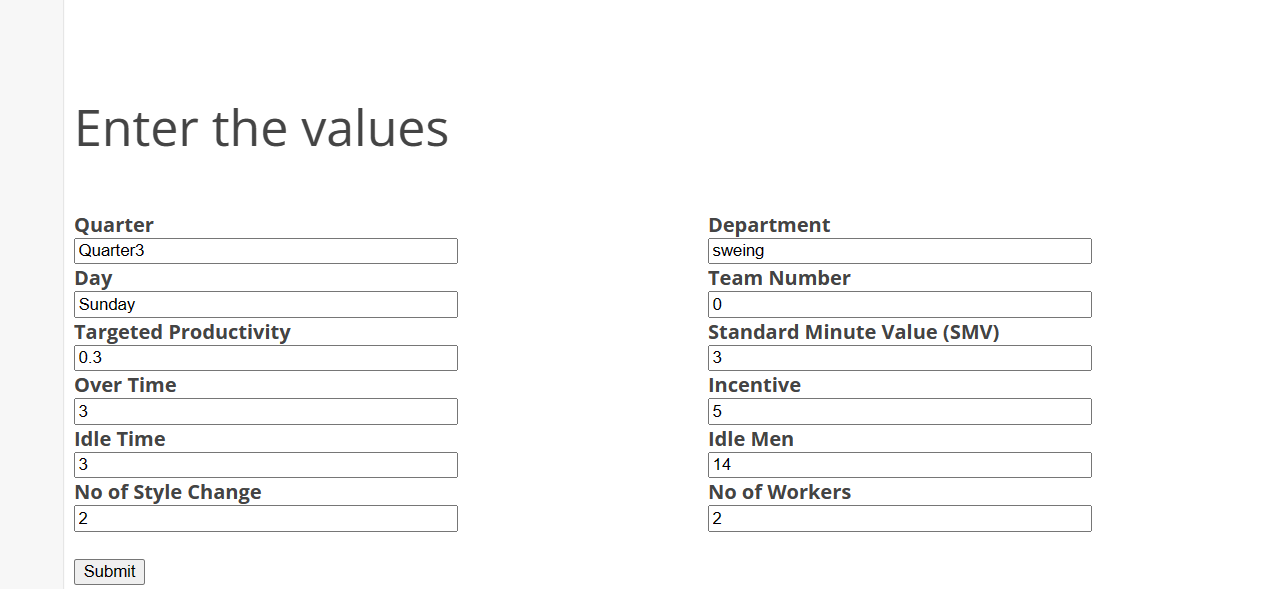


Lets look at what our prediction file looks like when you click the ‘Productivity prediction’ button in the top console bar below you will get redirected to the Enter values page After inputting the values and clicking on the submit button you will get output. We will see 3 Inputs and Outputs.

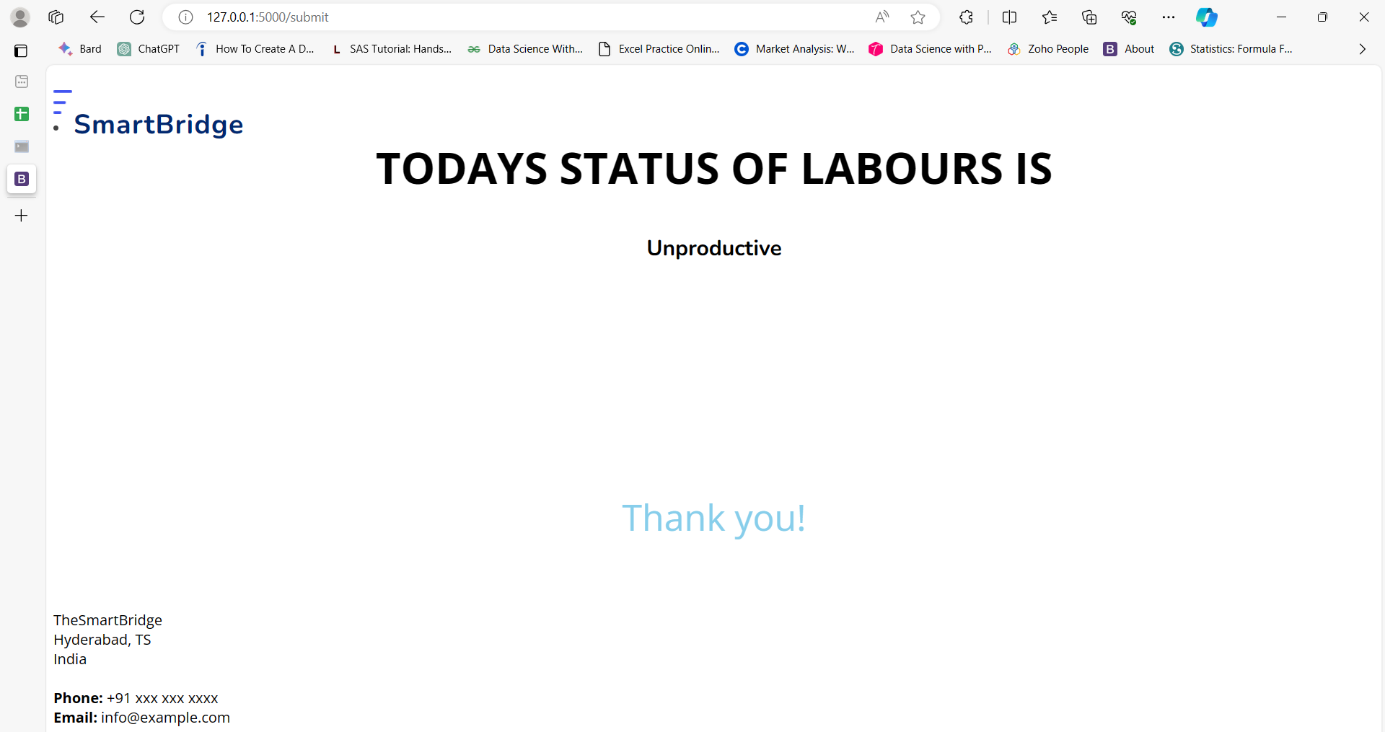
**Output 1:**

****

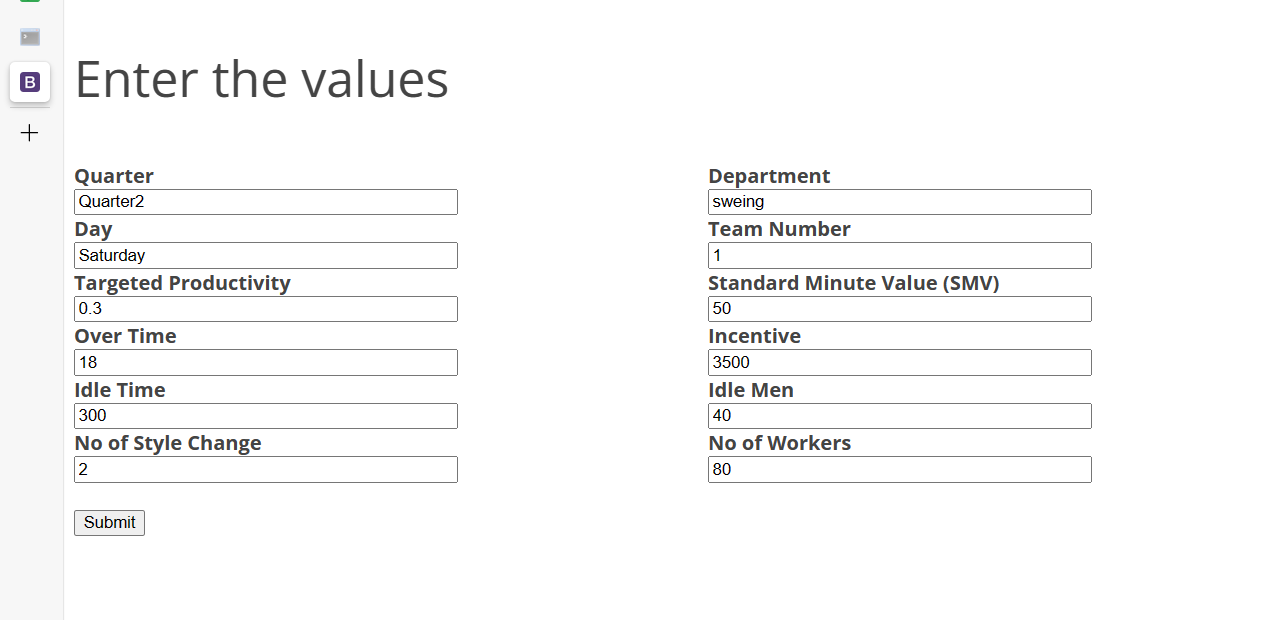
**Input 2:**

****

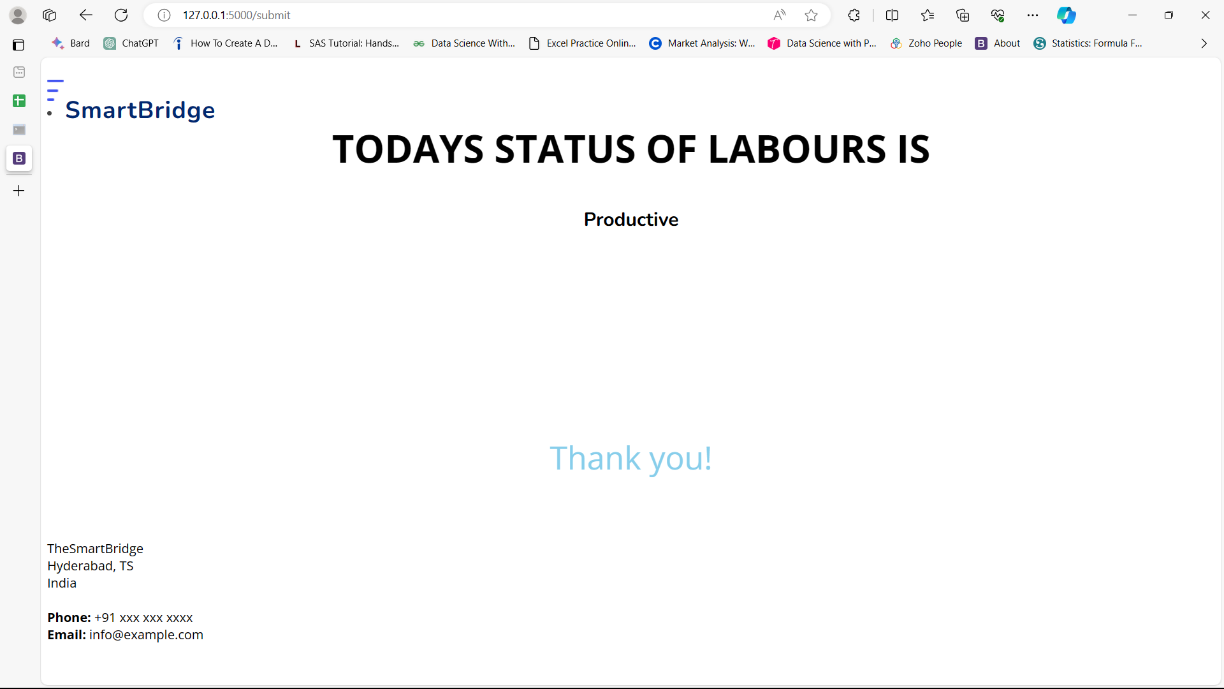
**Output 2:**



**Input 3:**



**Output 3:**

****

**Milestone 7: Project Demonstration & Documentation**

Below mentioned deliverables to be submitted along with other deliverables

## Activity 1:- Record explanation Video for project end to end solution

**Activity 2:- Project Documentation-Step by step project development procedure**

Create document as per the template provided