

Text  Description automatically generated with low confidence

CONSUMER PRICE INDEX

PREDICTION USING

MACHINE LEARNING

SmartInternz

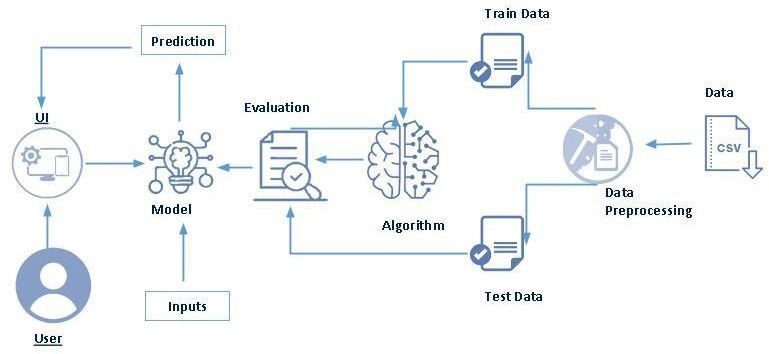
[www.smartinternz.com](http://www.smartinternz.com/)

**CPI Prediction Using Machine Learning**

Consumer Price Index:

Consumer Price Indices (CPI) measure changes over time in general level of prices of goods and services that households acquire for the purpose of consumption. CPI numbers are widely used as a macroeconomic indicator of inflation, as a tool by governments and central banks for inflation targeting and for monitoring price stability, and as deflators in the national accounts. CPI is also used for indexing dearness allowance to employees for increase in prices. The main purpose of predicting the Consumer Price Index (CPI) is to gain insights into future inflation trends and to make informed decisions in various areas, including economic policy, business planning, investment, and personal finance

# Technical Architecture:



**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,
* 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 & Hyper parameter Tuning
  + Testing model with multiple evaluation metrics
  + Comparing model accuracy before & after applying hyper parameter 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 project development procedure

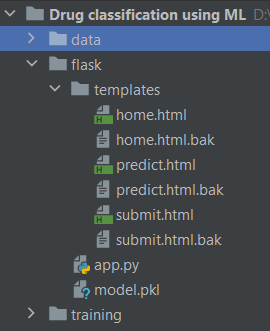
# Prior Knowledge:

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>
* 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>
* Xgboost: [https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-](https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/) [understand-the-math-behind-xgboost/](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-](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.
* Model.pkl is 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

Here are some potential business requirements for a Consumer Price Index (CPI):

* + Data Quality and Accuracy: Obtain and maintain reliable, up-to-date CPI data. Ensure data sources are reputable and follow standardized data collection practices. Implement data validation processes to maintain high data quality standards.
  + Transparency and Methodology: Clearly document the sources of CPI data, including the basket of goods and services. Disclose the methodology used for data calculation, including the formula and weightings. Make historical data and methodology readily available for analysis and research.
  + Accessibility and User-Friendly Interface: Develop an intuitive and user-friendly interface for easy access to CPI data. Offer various data formats and delivery methods to cater to diverse user needs. Provide user support and resources to help users understand and interpret the data.

.

* + Compliance, Security, and Privacy: Ensure compliance with relevant regulations and standards governing CPI calculation and reporting. Implement robust data security measures to protect sensitive CPI information. Safeguard user data and privacy by adhering to data protection regulations..
  + Customization and Feedback Mechanism: Allow users to customize data views, conduct analyses, and generate reports. Establish a feedback mechanism for users to report discrepancies, suggest improvements, and provide input. Continuously improve the CPI project based on user feedback and evolving user needs.

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

## A literature survey for a Consumer Price Index (CPI) analysis project involves researching and reviewing existing studies, articles, and publications related to CPI. This survey aims to:

## Current CPI Measurement: Explore existing CPI calculation methods, their advantages, and limitations.

## Data Sources: Investigate the sources of data used for CPI computation, considering data quality and reliability.

## Regression Models: Examine various regression models applied to CPI analysis, highlighting their variables and effectiveness.

## Research Findings: Summarize significant findings from previous research on CPI, including insights and correlations.

## Methodologies and Techniques: Analyze the methodologies and techniques employed in CPI analysis, such as time series models or machine learning approaches.

## Challenges and Limitations: Identify limitations and challenges, like data quality issues and seasonality, and how researchers address them.

## Research Gaps and Focus: Highlight areas where the current research falls short and explain how your project intends to contribute and improve CPI analysis.

## Activity 4: Social or Business Impact.

**Social Impact**: - A CPI regression project has the potential to empower policymakers and economists with valuable insights into inflation trends. By accurately predicting changes in the Consumer Price Index, it can inform the development of more effective economic policies, such as inflation control measures and minimum wage adjustments. This, in turn, can positively impact the overall economic well-being of a nation's citizens and contribute to a more stable and equitable society.

Business Model/Impact: - A CPI regression project can have a significant impact on businesses and industries by providing them with essential insights into consumer price trends. This data enables businesses to make informed decisions regarding pricing strategies, resource allocation, and investment choices. It can help companies adapt to changing economic conditions, optimize their operations, and remain competitive in the market. Thus, a CPI regression project contributes to more effective business planning and sustainable growth within various industries..

# 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.

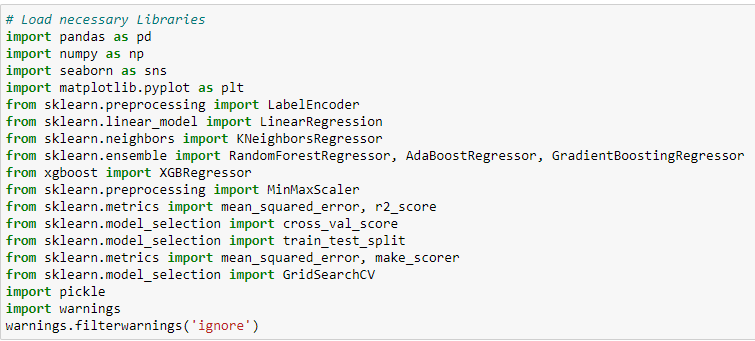
Link: https://www.kaggle.com/datasets/fiq423ubf/all-india-consumer-price-index-ruralurban/data

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

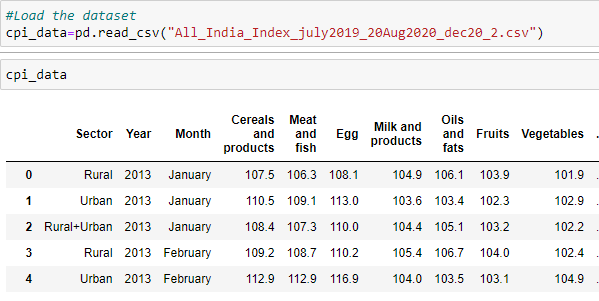
Import the necessary libraries as shown in the image.



## 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.

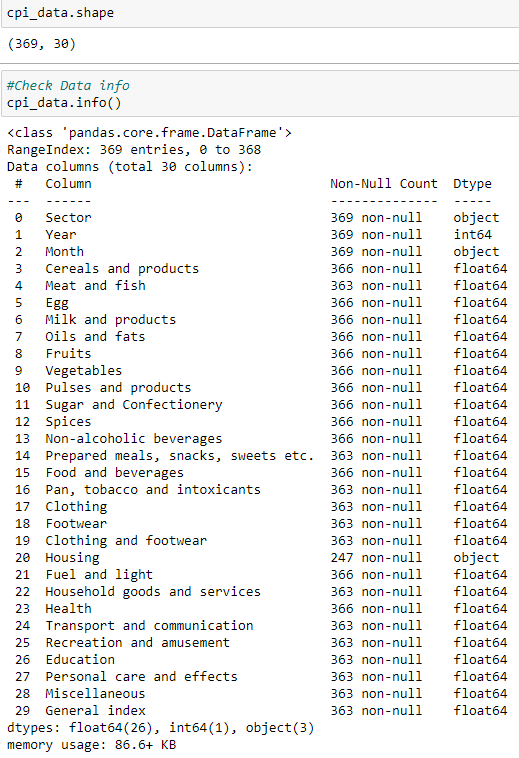
The download data set is not suitable for training the machine learning model as it might have so much randomness so we need to clean the dataset properly in order to fetch good results. This activity includes the following steps.

* Handling missing values
* Handling categorical data
* Handling Outliers

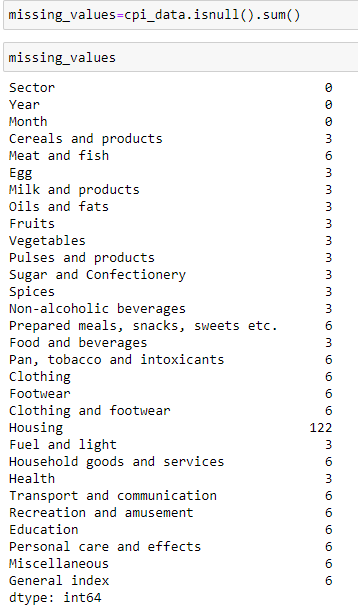
Note: These are the general steps of pre-processing the data before using it for machine learning. Depending on the condition of your dataset, you may or may not have to go through all these steps.

## Activity 2.1: Handling missing values

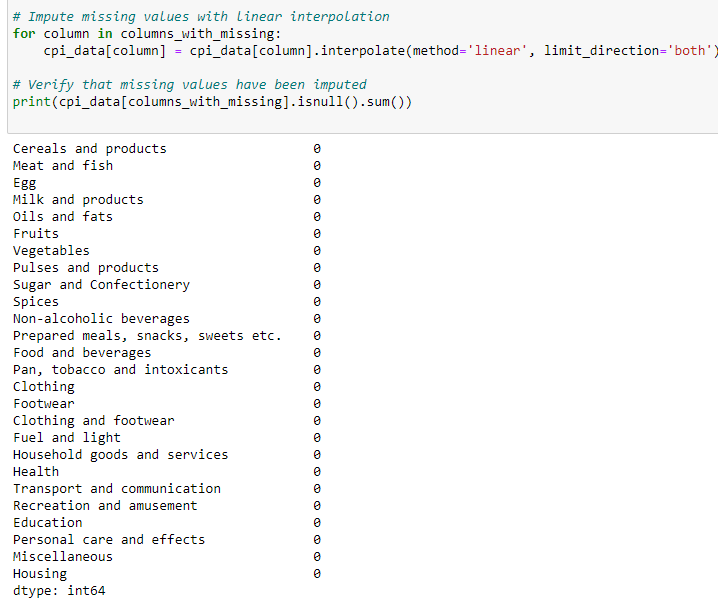
* Let’s find the shape of our dataset first. To find the shape of our data, the df.shape method is used. To find the data type, df.info() function is used.



* For checking the null values, df.isnull () function is used. To sum those null values we use .sum () function. From the below image we found that there are null values present in our dataset.



* For handling missing values here we used interpolation method because we observed less variance between missing values.

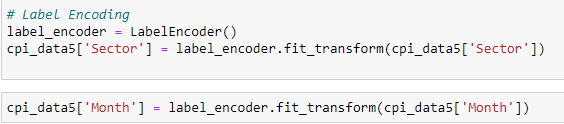


## Activity 2.2: Handling Categorical Values

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 Label encoding .

* + In our project, categorical features are Month, and Sector. With label encoding is done.



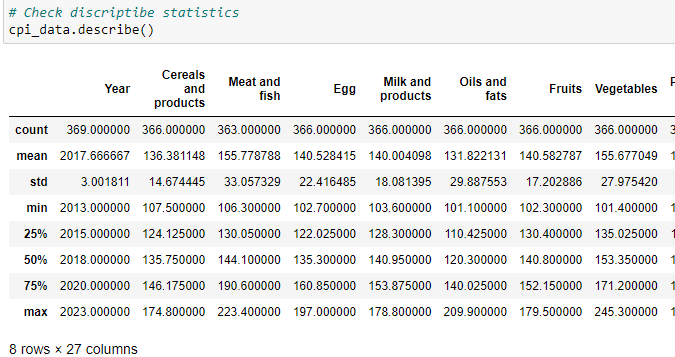
## Activity 2.3: Handling Imbalance Data

In our data set there is no imbalance data. So we skip this step.

# 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.



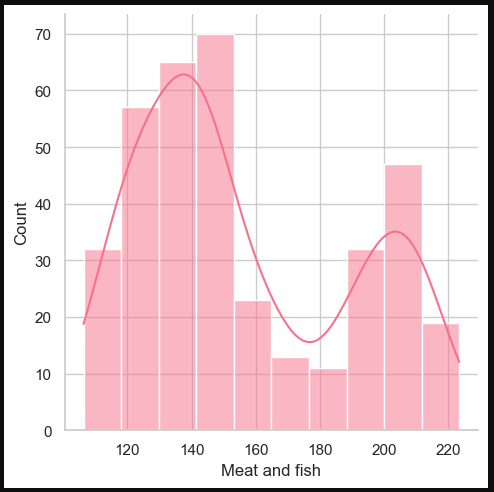
## Activity 2: Visual analysis

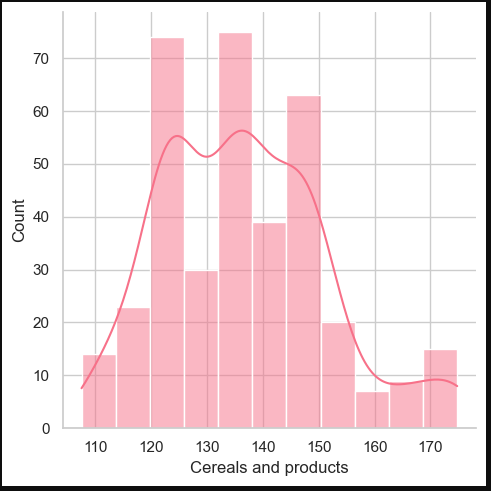
Visual analysis is the process of using visual representations, such as charts, plots, and graphs, to explore and understand data. It is a way to quickly identify patterns, trends, and outliers in the data, which can help to gain insights and make informed decisions.

## Activity 2.1: Univariate analysis

In simple words, univariate analysis understands the data with single feature. Here we have displayed two different graphs such as distplot and count plot.

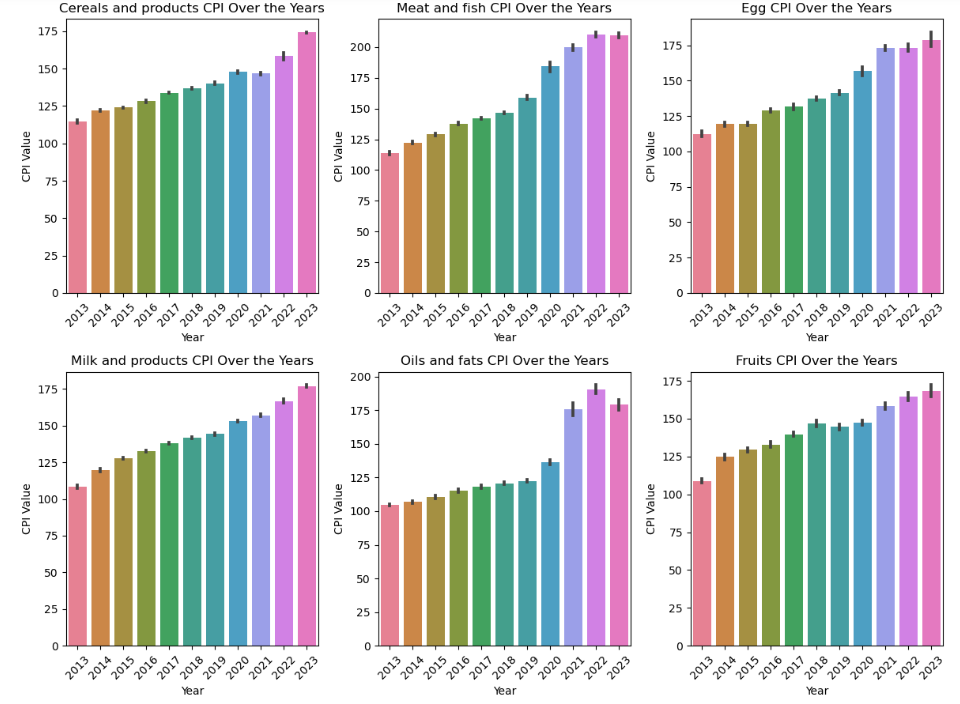
Seaborne package provides a wonderful function distplot. With the help of distplot, we can find the distribution of the feature. To make multiple graphs in a single plot, we use subplot.



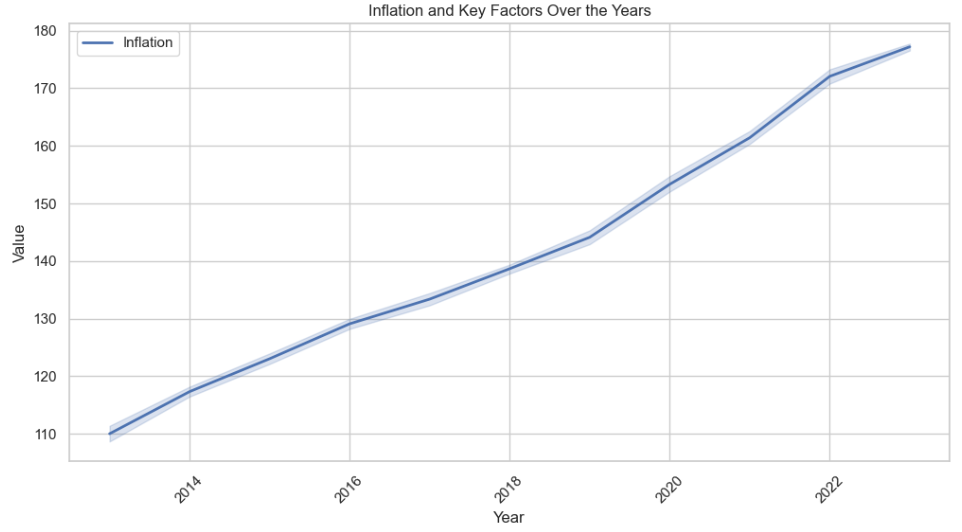


## Activity 2.2: Bivariate analysis

To find the relation between two features we use bivariate analysis. Here we are visualizing the relationship between Each feature with year. For example



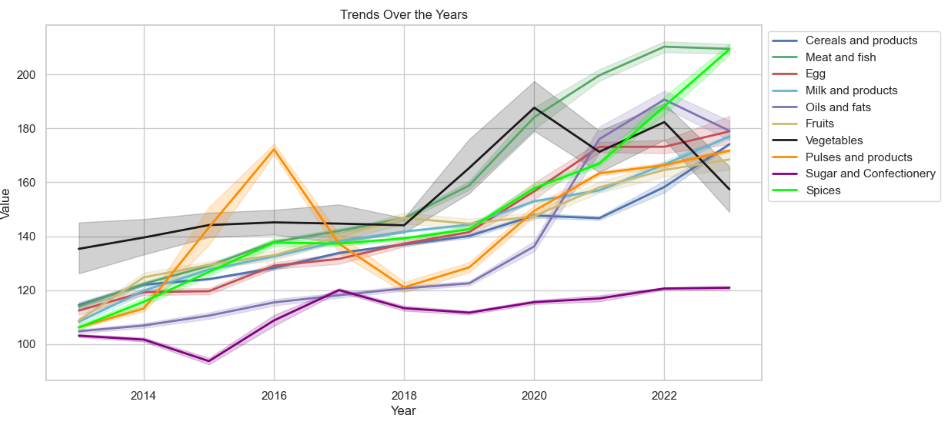
* Here we used grouped bar plot and we observe that each product cpi is increasing every year in some it is increasing rapidly and in some steadily increasing.



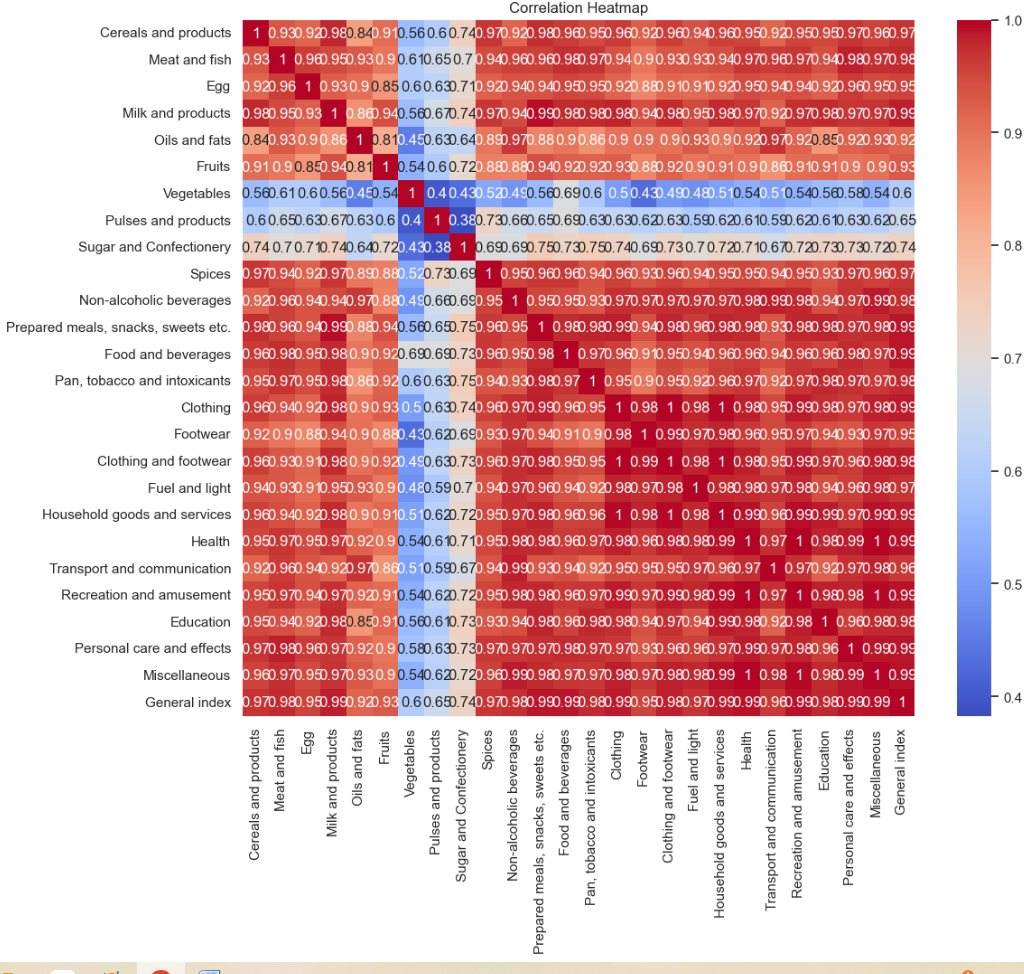
* Here we used line plot between general index (Target variable) and year to see inflation over the years.

## Activity 2.3: Multivariate analysis

In simple words, multivariate analysis is to find the relation between multiple features. Here we have used line plot from seaborne package.by using we can compare inflation between multi products with respect to year.



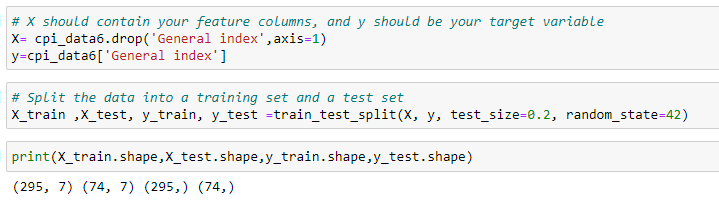
We use heat map also to check multicollinearity.



**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.



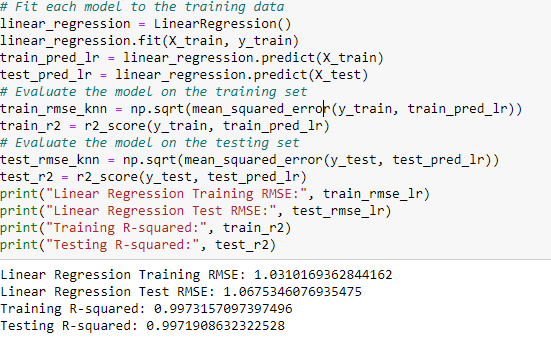
# Milestone 4: Model Building

## 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 three regression algorithms. The best model is saved based on its performance.

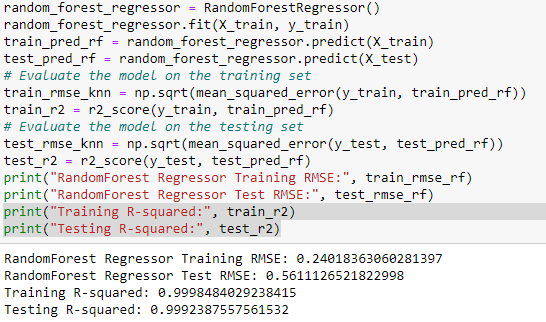
## Activity 1.1: Linear Regression model

A function named linear regression created and train and test data are passed as the parameters. Inside the function, Lasso regression 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, Calculates the root mean squared error (RMSE) for the training set. Calculates the R-squared Score for the training set. Calculates the RMSE for the testing set. Calculates the R-squared score for the testing set. Finally, it prints the training RMSE, testing RMSE, training R-squared, and Testing R-squared.



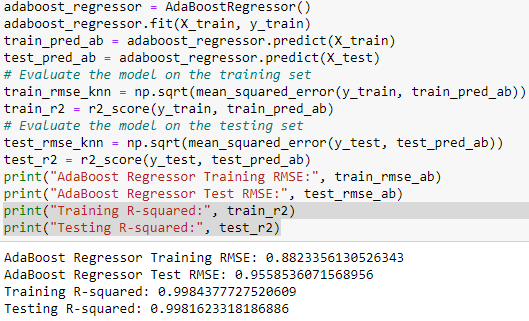
## Activity 1.2: Random forest model

A function named Random Forest Regression created and train and test data are passed as the parameters. Inside the function, Lasso regression 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, Calculates the root mean squared error (RMSE) for the training set. Calculates the R-squared Score for the training set. Calculates the RMSE for the testing set. Calculates the R-squared score For the testing set. Finally, it prints the training RMSE, testing RMSE, training R-squared, and Testing R-squared .



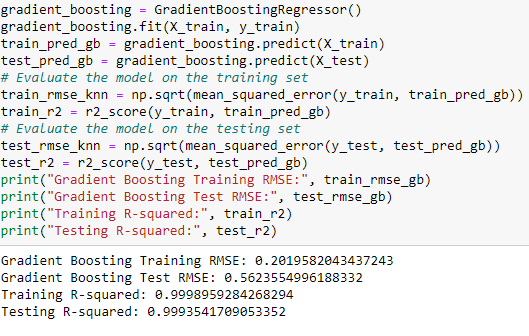
## Activity 1.3: Adaboost

A function named Adaboost created. and train and test data are passed as the parameters. Inside the function, Lasso regression 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, Calculates the root mean squared error (RMSE) for the training set. Calculates the R-squared Score for the training set. Calculates the RMSE for the testing set. Calculates the R-squared score for the testing set. Finally, it prints the training RMSE, testing RMSE, training R-squared, and Testing R-squared.



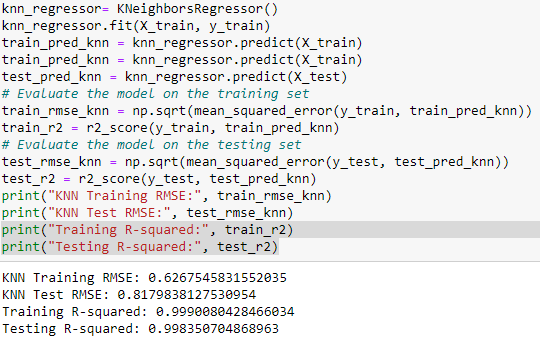
## Activity 1.4: Gradiant\_boosting

A function named Gradiant\_boosting created. and train and test data are passed as the parameters. Inside the function, Lasso regression 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, Calculates the root mean squared error (RMSE) for the training set. Calculates the R-squared Score for the training set. Calculates the RMSE for the testing set. Calculates the R-squared score for the testing set. Finally, it prints the training RMSE, testing RMSE, training R-squared, and Testing R-squared.



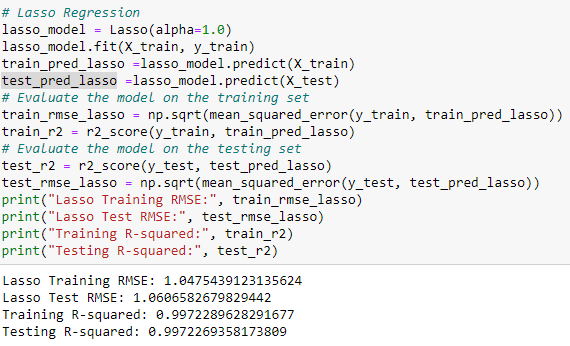
## Activity 1.5: KNN model

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, Calculates the root mean squared error (RMSE) for the training set. Calculates the R-squared Score for the training set. Calculates the RMSE for the testing set. Calculates the R-squared score for the testing set. Finally, it prints the training RMSE, testing RMSE, training R-squared, and Testing R-squared.



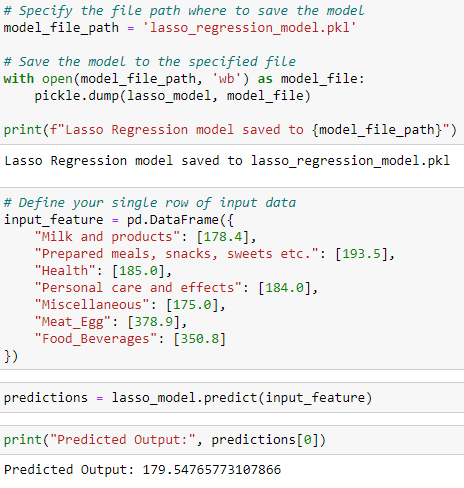
## Activity 1.6: Lasso Regression

A function named Lasso model is created and train and test data are passed as the parameters. Inside the function, Lasso regression 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, Calculates the root mean squared error (RMSE) for the training set. Calculates the R-squared Score for the training set. Calculates the RMSE for the testing set. Calculates the R-squared score For the testing set. Finally, it prints the training RMSE, testing RMSE, training R-squared, and Testing R-squared .



## Activity 2: Testing the model

Here we have tested with Lasso Regression algorithm. You can test with all algorithm. With the help of predict () function.



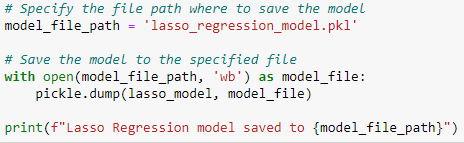
# Milestone 5: Performance Testing & Hyperparameter Tuning

## Activity 1: Comparing model accuracy before & after applying hyper parameter tuning (Hyper parameter tuning is optional). For this project it is not required

# 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:

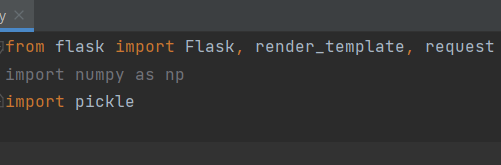
For this project create two HTML files namely

* index.html
* inner\_page.html

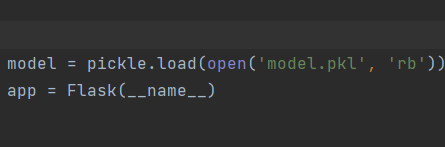
and save them in the templates folder. Refer this [link](https://drive.google.com/drive/folders/1K-C2uvRstV8x6bwsOY_BWBvXdzjbbfPF?usp=share_link) for templates.

## Activity 2.2: Build Python code:

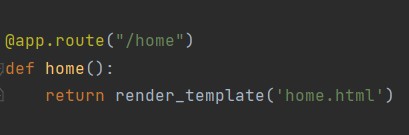
Import the libraries



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.



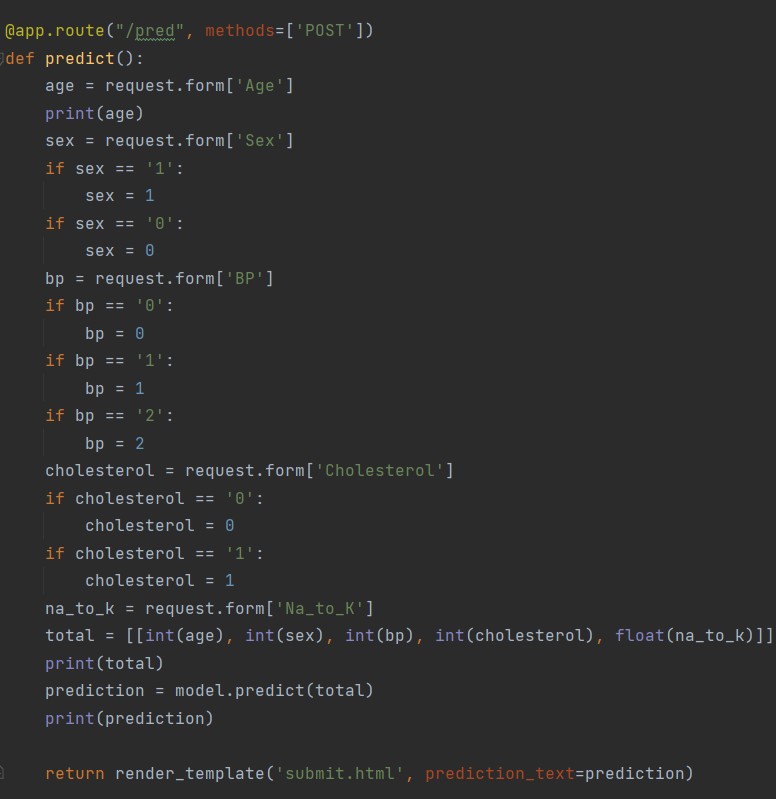
Render HTML page:



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

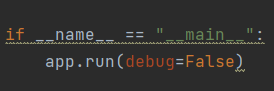
In the above example, ‘/’ URL is bound with the home.html function. Hence, when the home page of the web server is opened in the browser, the html page will be rendered. Whenever you enter the values from the html page the values can be retrieved using POST Method.

Retrieves the value from UI:



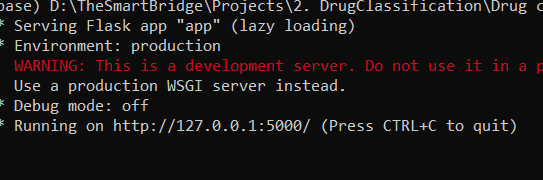
Here we are routing our app to predict() function. This function retrieves all the values from the HTML page using Post request. That is stored in an array. This array is passed to the model.predict() function. This function returns the prediction. And this prediction value will be rendered to the text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

Main Function:

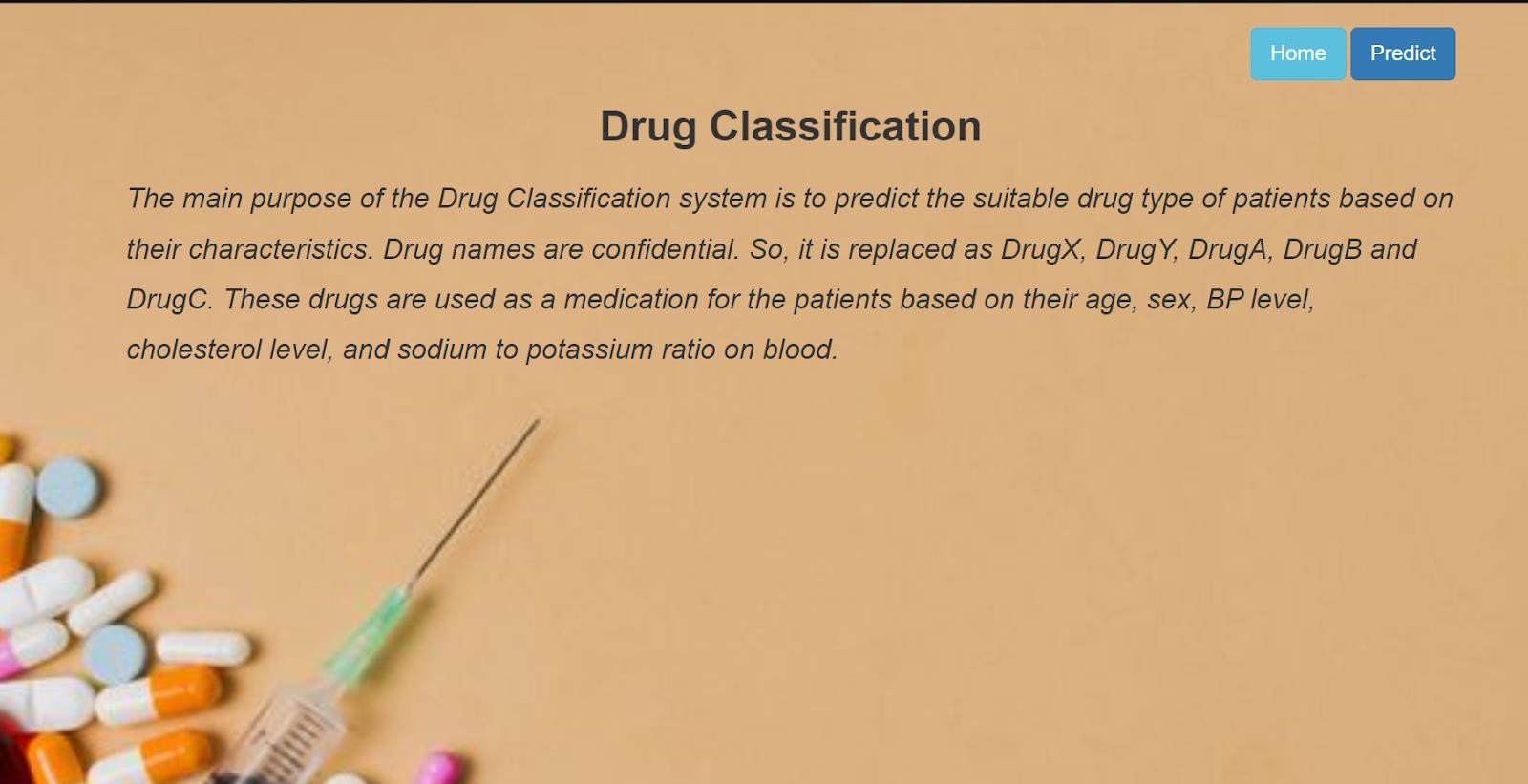


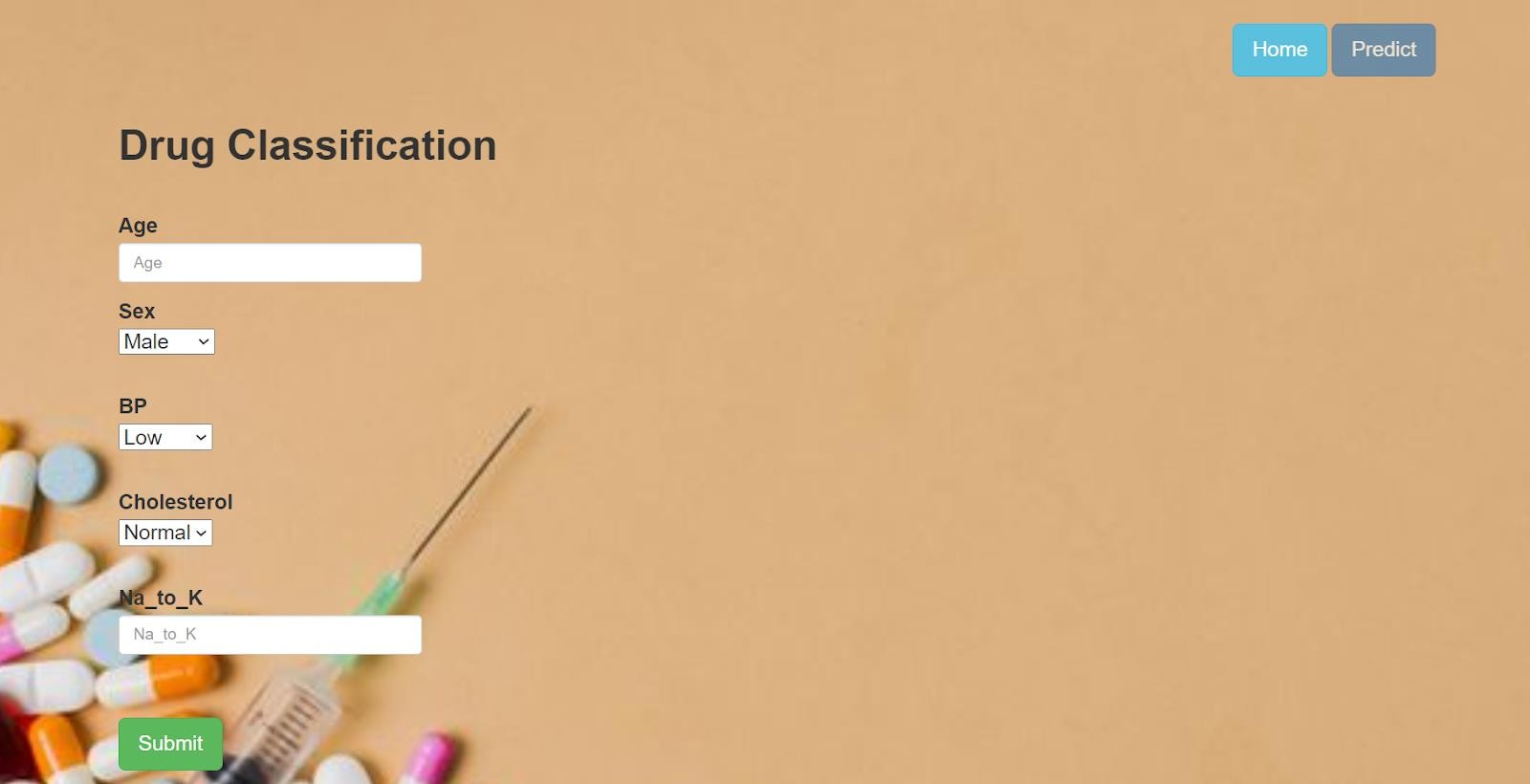
## Activity 2.3: Run the web application

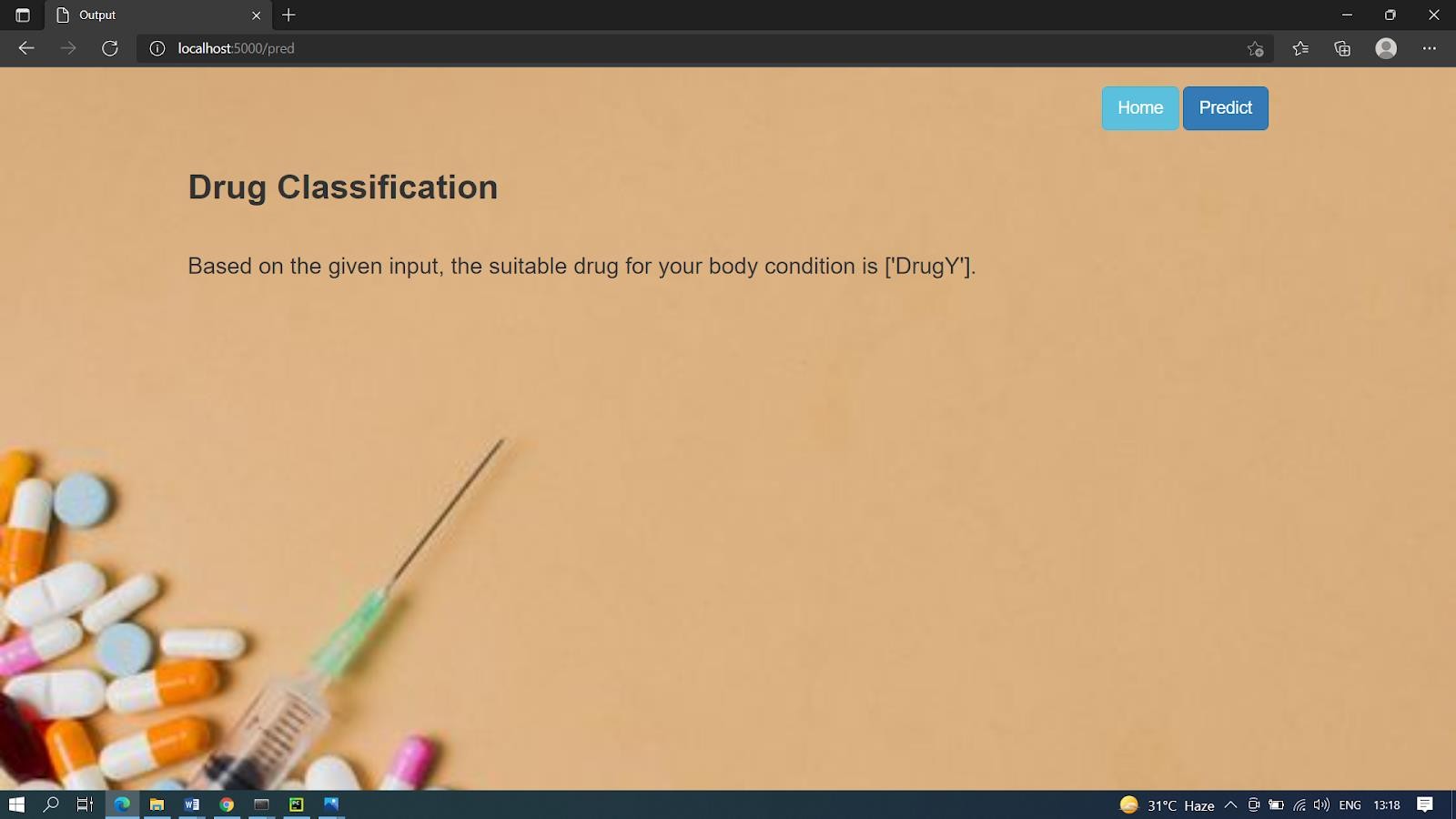
* + Open anaconda prompt from the start menu
  + Navigate to the folder where your python script is.
  + Now type “python app.py” command
  + Navigate to the localhost where you can view your web page.
  + Click on the predict button from the top left corner, enter the inputs, click on the submit button,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



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







**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