# [Avocado Harvest Oracle: A Gourmet Journey into Predicting Tomorrow's Green Gold Prices](https://sbaiteam.atlassian.net/browse/D2P-9?atlOrigin=eyJpIjoiMThmNTZkODk3MGM3NDlhMjk2OTYyZmQ5NDY0NjFmNGIiLCJwIjoiaiJ9)

The 'Avocado Harvest Oracle: A Gourmet Journey into Predicting Tomorrow's Green Gold Prices' is a pioneering regression project focused on predicting avocado prices.

By employing advanced regression techniques on this comprehensive dataset, the project aims to predict avocado prices. The model takes into account a multitude of factors, including avocado quantities, bag sizes, temporal aspects, and categorical variables. The precision of the regression analysis contributes to informed decision-making for avocado farmers and traders, optimizing planting, harvesting, and trading strategies based on predicted market trends. In essence, the Avocado Harvest Oracle signifies a convergence of data science and agriculture, offering a data-driven approach to navigate the complexities of the avocado industry.

## Technical Architecture:

In this architecture, a user inputs the relevant engagement features such as small/medium avocado, XLarge Bags, related to avocado properties. The Flask application is responsible for handling the request and returning the regression of the soil moisture status for given month or days.

The Flask application then passes the inputs metrics to the trained algorithm model, which processes the input data and returns the results based on the learned relationship between the input metrics and repayment interval.



# Pre requisites:

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

●       **Anaconda navigator and pycharm:**

o   Refer the link below to download anaconda navigator

o   Link :<https://youtu.be/1ra4zH2G4o0>

●       **Python packages:**

o   Open anaconda prompt as administrator

o   Type “pip install numpy” and click enter.

o   Type “pip install pandas” and click enter.

o   Type “pip install scikit-learn” and click enter.

o   Type ”pip install matplotlib” and click enter.

o   Type ”pip install scipy” and click enter.

o   Type ”pip install pickle-mixin” and click enter.

o   Type ”pip install seaborn” and click enter.

o   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 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:
  + Define problem / Problem understanding
* Specify the problem statement
* Business Requirements
* Social or Business Impact
  + Data Collection and Preparation
* Collect the dataset
* Data Preparation
  + Exploratory Data Analysis
* Descriptive statistical
* Visual Analysis
  + Model Building
* Train-test split
* Training and testing the Models using multiple algorithms
  + Performance Testing
* Comparing model accuracy
  + Model Deployment
* Save the best model
* Test the model
* Integrate with Web Framework
* GUI
  + Project Demonstration & Documentation

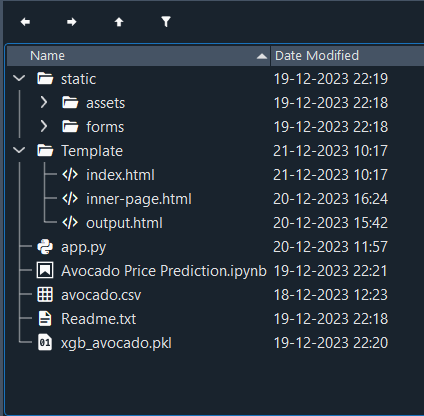
## Prior Knowledge:

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

* + ML Concepts:
* Supervised learning: [https://www.javatpoint.com/supervised-machine-learning](http://www.javatpoint.com/supervised-machine-learning)
* Linear Regression: <https://www.javatpoint.com/linear-regression-in-machine-learning>
* KNN: <https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning>
* SVR: <https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm>
* Decisiontree: <https://www.geeksforgeeks.org/python-decision-tree-regression-using-sklearn/>
* Random forest: <https://www.geeksforgeeks.org/random-forest-regression-in-python/>
* XGBoost: <https://www.javatpoint.com/xgboost-ml-model-in-python>
* 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](http://www.youtube.com/watch?v=lj4I_CvBnt0)

## Project Structure:

Create project folder which contains files as shown below:



* The data obtained is in csv files splitted for training and testing.
* We are building a Flask application which will require the html files to be stored in the templates folder.
* app.py file is used for routing purposes using scripting.
* rfc.pkl is the saved model. This will further be used in the Flask integration.

# Milestone 1: Define Problem/ Problem Understanding

## Activity 1: Specify the Problem Statement.

The purpose of the 'Avocado Harvest Oracle: A Gourmet Journey into Predicting Tomorrow's Green Gold Prices’ is to access the prices of avocados.

## Activity 2: Business Requirements.

The project focuses on robust risk management and financial planning for avocado pricing. It leverages historical data and predictive models to analyze trends in avocado prices, sizes, and external factors such as climate and market demand. Financial planning involves allocating funds for data tasks, model development, and accommodating unforeseen price variations. To enhance risk mitigation, a contingency fund is designated, and financial strategies are explored. The implementation of an early warning system includes continuous monitoring tools and machine learning for anomaly detection, ensuring timely alerts and interventions in response to significant deviations in avocado price predictions. This approach seeks to optimize decision-making and navigate market uncertainties efficiently.

## Activity 3: Social or Business Impact.

Avocado price prediction has significant social and business impacts. Socially, it enhances economic welfare by optimizing pricing strategies and supporting sustainable practices. This stability extends to communities dependent on avocado farming, indirectly benefiting livelihoods. On the business front, predicting avocado prices enables optimized market strategies, increased profitability, and operational efficiency through resource savings. Proactive prediction also aids risk mitigation, minimizing losses and ensuring long-term stability in the avocado market.

# Milestone 2: Data Collection and Preparation:

Machine Learning depends heavily on data. It is the most crucial part aspect that makes algorithm training possible. So, this section guides on how to download 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/neuromusic/avocado-prices>

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

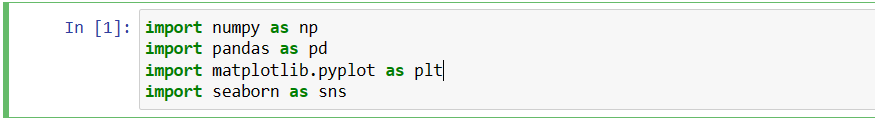
Note: There are a number of techniques for understanding the data. But here we have used some of it. In an additional way, you can use multiple techniques.

**About this Data**

**Attribute Information:**

|  |  |
| --- | --- |
| **Feature** | **Description** |
| Small/medium Avocados | Total number of avocados with PLU 4046 sold |
| large Avocados | Total number of avocados with PLU 4225 sold |
| extra\_large Avocados | Total number of avocados with PLU 4770 sold |
| Small Bags | Quantity of avocados sold in small-sized bags. |
| Large Bags | Quantity of avocados sold in large-sized bags. |
| XLarge Bags | Quantity of avocados sold in extra-large-sized bags. |
| year | the year |
| Month | The month |
| Day | The day |
| type\_encoded | conventional or organic |
| Region\_encoded | the city or region of the observation |

**Activity 1.1: Importing the Libraries**

****

**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.

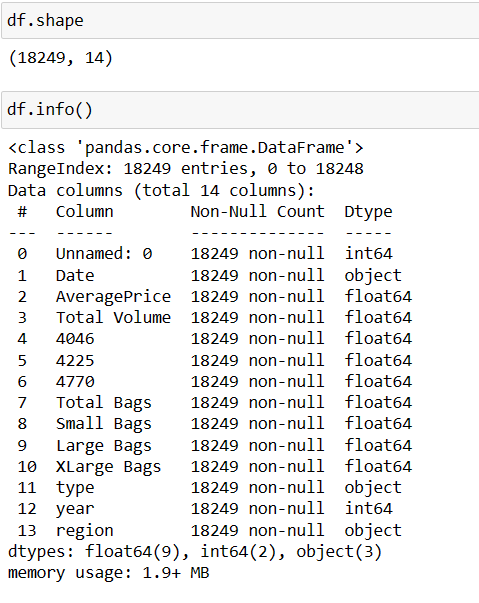
The download data set not suitable for training the machine learning model as it might have so much randomness so we need to clean the dataset properly in order to fetch good results. This activity includes the following steps.

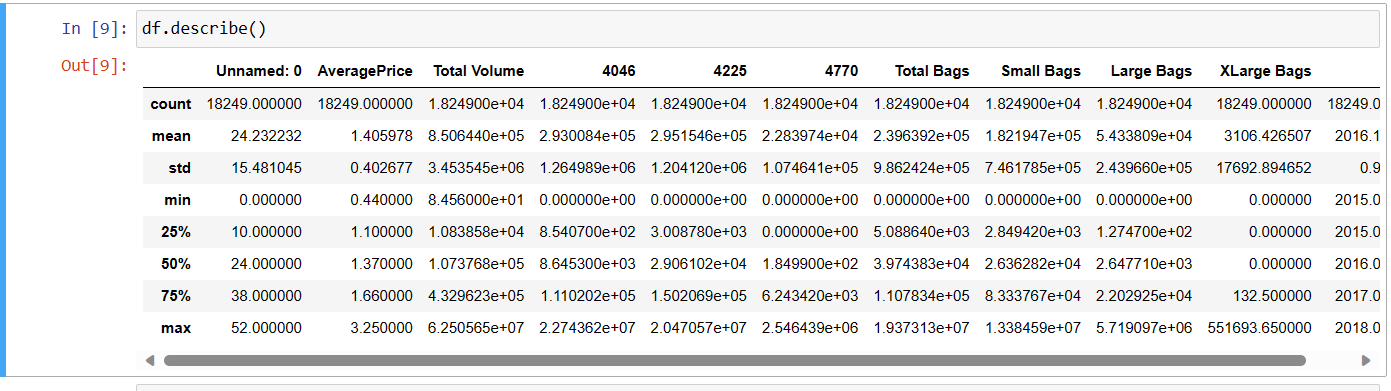
* Getting the Preliminary Information about the Dataset
* Handling missing values
* Dropping Unwanted column

**Activity 2.1: Getting the Preliminary Information about the Dataset**

i.e. Non-Null, Count, Dtype

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() and df.describe() function is used.

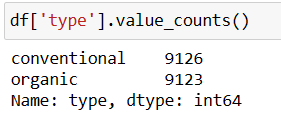




# Milestone 3: Exploratory Data Analysis

## Activity 1: Descriptive Statistical

Descriptive analysis is to study of the basic features of data with the statistical process. Here pandas have 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: Visualization

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 Class Analysis:

## 

## 

## 

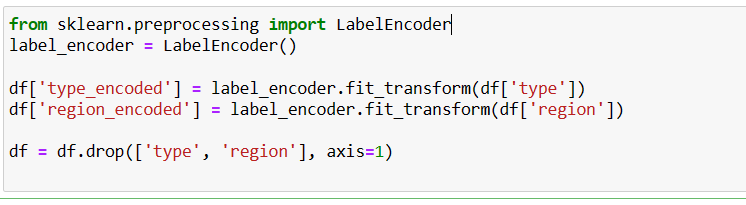
## 

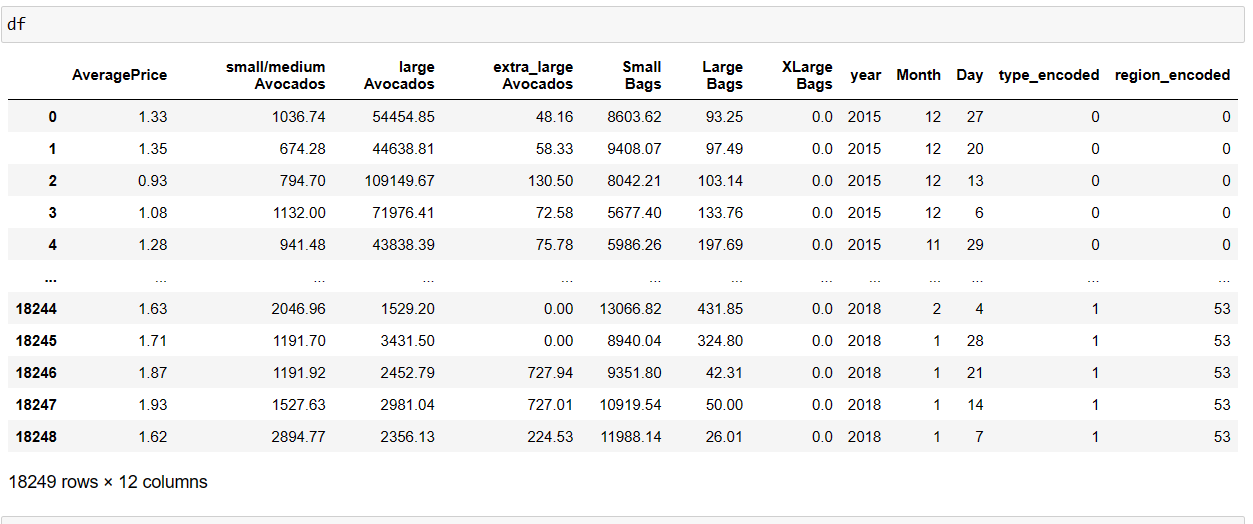
## 

## Activity 3: Encoding Data

## 

All the columns we are having in our data are categorical so we need to encode them.

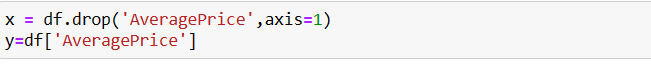


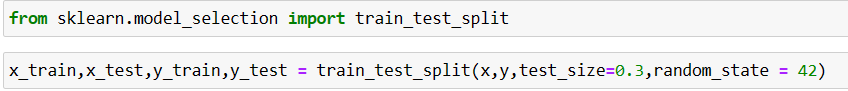


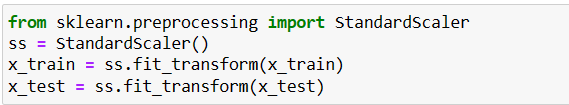
# Milestone 4 : Model Building

## Activity 1 : Train-test split

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, 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.

In the current project we have below columns as x variable and y variables (Target variable):

🡪small/medium Avocados

🡪large Avocados

🡪extra\_large Avocados

🡪Small Bags

🡪Large Bags

🡪XLarge Bags

🡪year

🡪Month

🡪Day

🡪type\_encoded

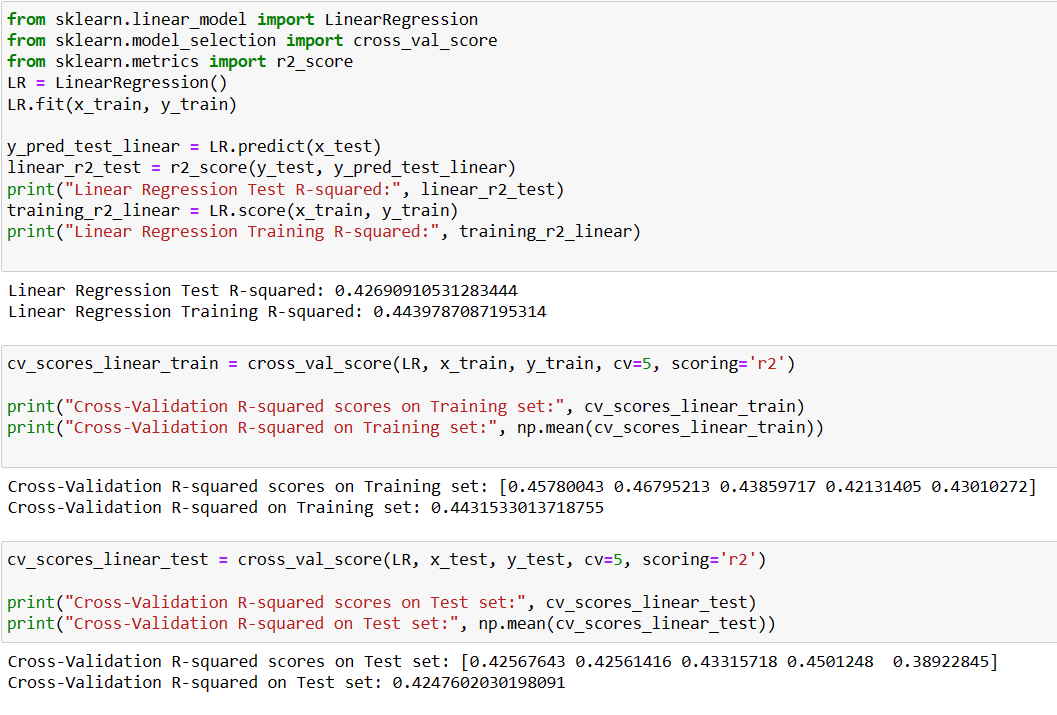
🡪region\_encoded

## Activity 2: Training and testing the models using 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 classification algorithms. The best model is saved based on its performance.

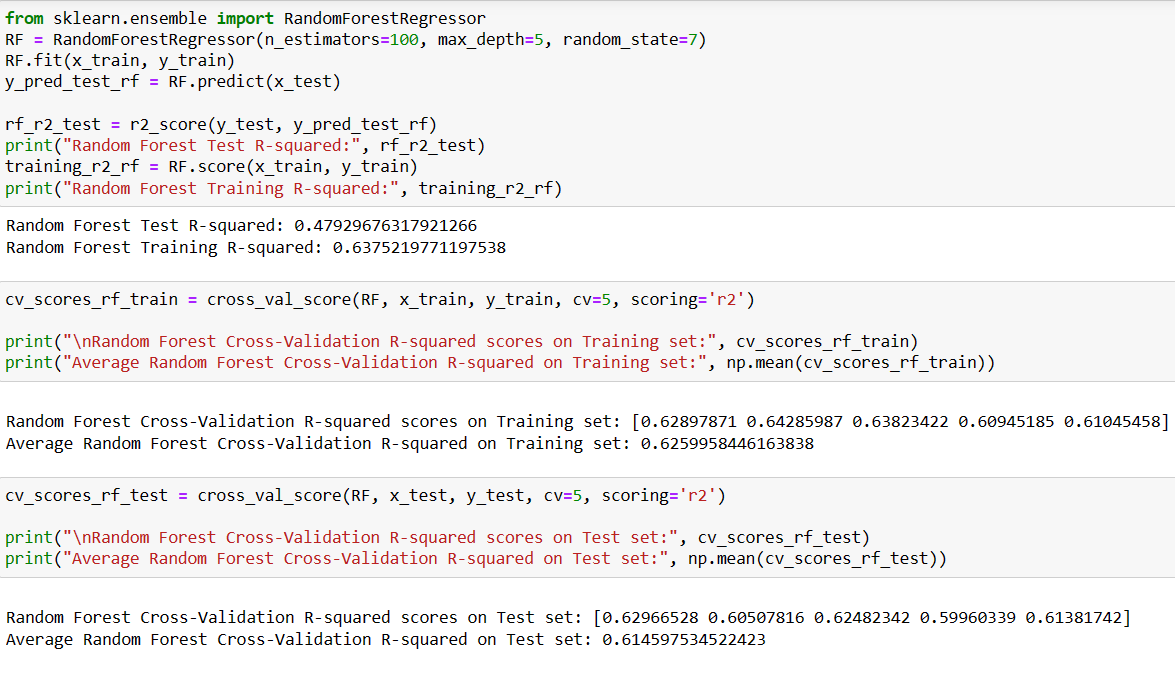
## Activity 2.1 Linear Regression:

A function named “LR” is created and train and test data are passed as the parameters. Inside the function, “linear 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 a new variable. For evaluating the model accuracy is calculated.



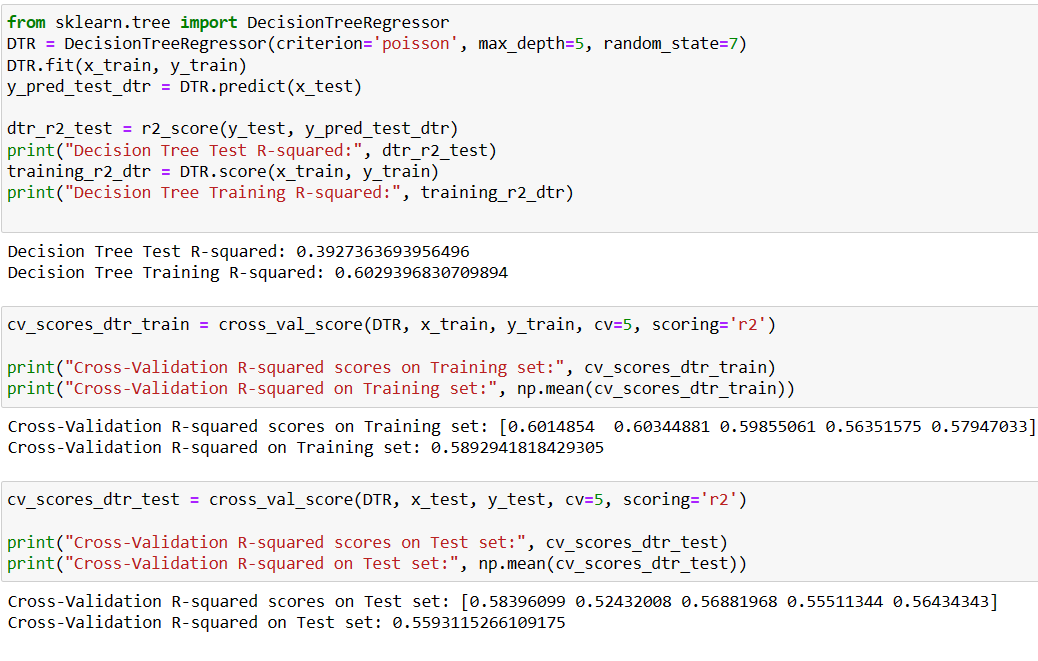
## Activity 2.2 Random Forest Regressor:

A function named “RF” is created and train and test data are passed as the parameters. Inside the function, Random Forest Regressor algorithm is initialized 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 accuracy is calculated.



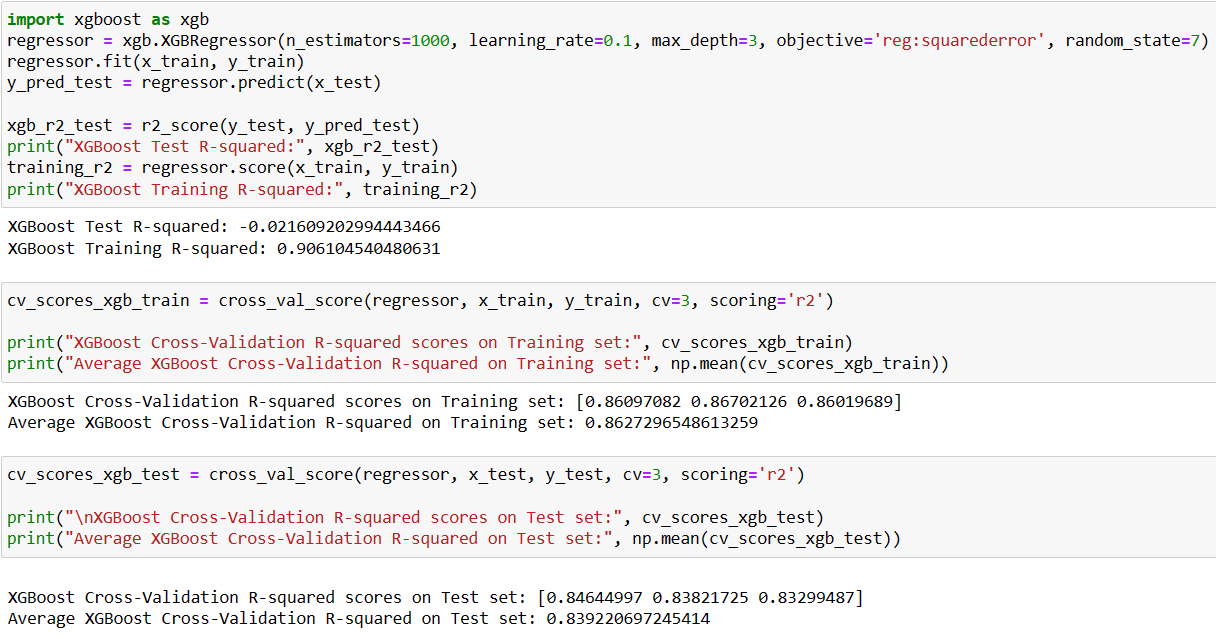
## Activity 2.3: Decision Tree Regressor:

A function named “DTR” is created and train and test data are passed as the parameters. Inside the function, Decision Tree Regressor 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, overfitting and accuracy is calculated.



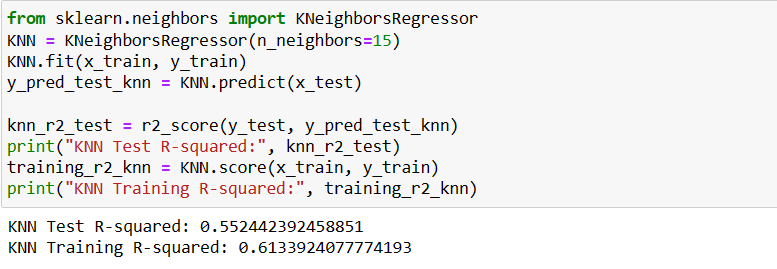
## Activity 2.4: XGBoost:

A function named “regressor” is created and train and test data are passed as the parameters. Inside the function, XGBoost 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, overfitting and accuracy is calculated.



## Activity 2.5 K-Nearest neighbour:

A function named “KNN” is created and train and test data are passed as the parameters. Inside the function, “kneighbours classifier” 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 accuracy is calculated.

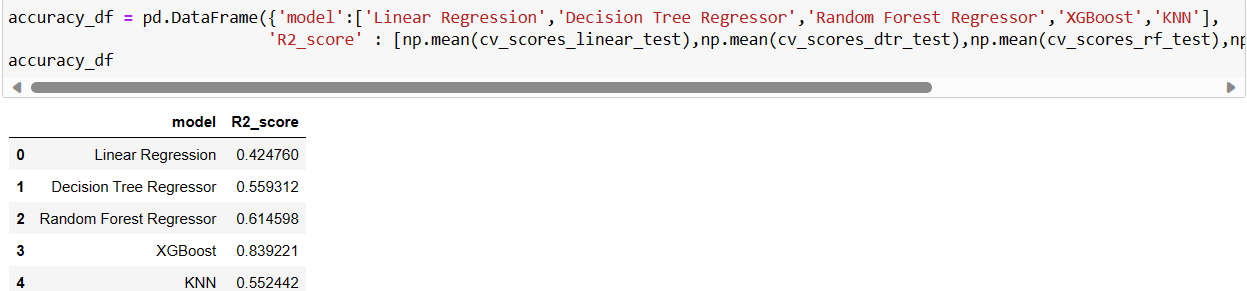


# Milestone 5 : Performance Testing

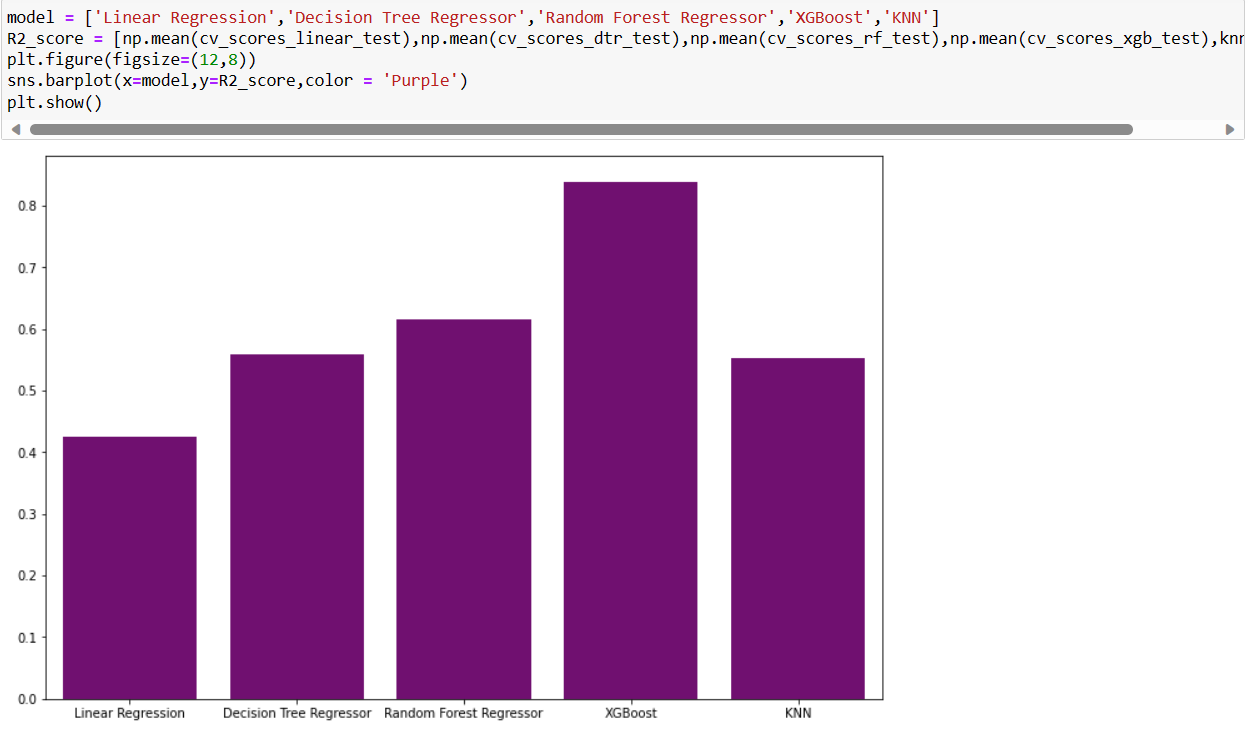
## Activity 1: Comparing all the Models.

For comparing the above five models, the accuracy\_df function is used.

Below is the accuracy comparison of all the models and we can clearly see that accuracy for Linear Regression, Decision Tree, Random Forest, XGBoost and K-Nearest Neighbors is 94 percent so we can take any of this model for our classification purpose.



## Activity 2: Graphical representation of the model comparison.

****

# Milestone 6: Model Deployment

## Activity 1: Save and load the best model

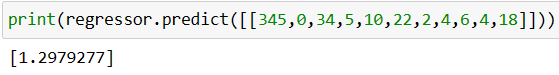
Saving the best model after comparing its performance using different evaluation metrics means selecting the model with the highest performance. This can be useful in avoiding the need to retrain the model every time it is needed and also to be able to use it in the future. 

We save the model using the pickle library into a file named rfc.pkl

## Activity 2: Test the model:

Let’s test the model first in python notebook itself.

As we have 6 features in this model, let’s check the output by giving all the inputs.



We can see above that out model has predicted “1.2979277”, that means model has classified this as Critical condition.

Hence, we can conclude that, out model is giving the accurate results.

## Activity 3: 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 3.1: Building HTML pages:

For this project create two HTML files namely and save them in the templates folder.

* index.html
* inner-page .html
* output.html

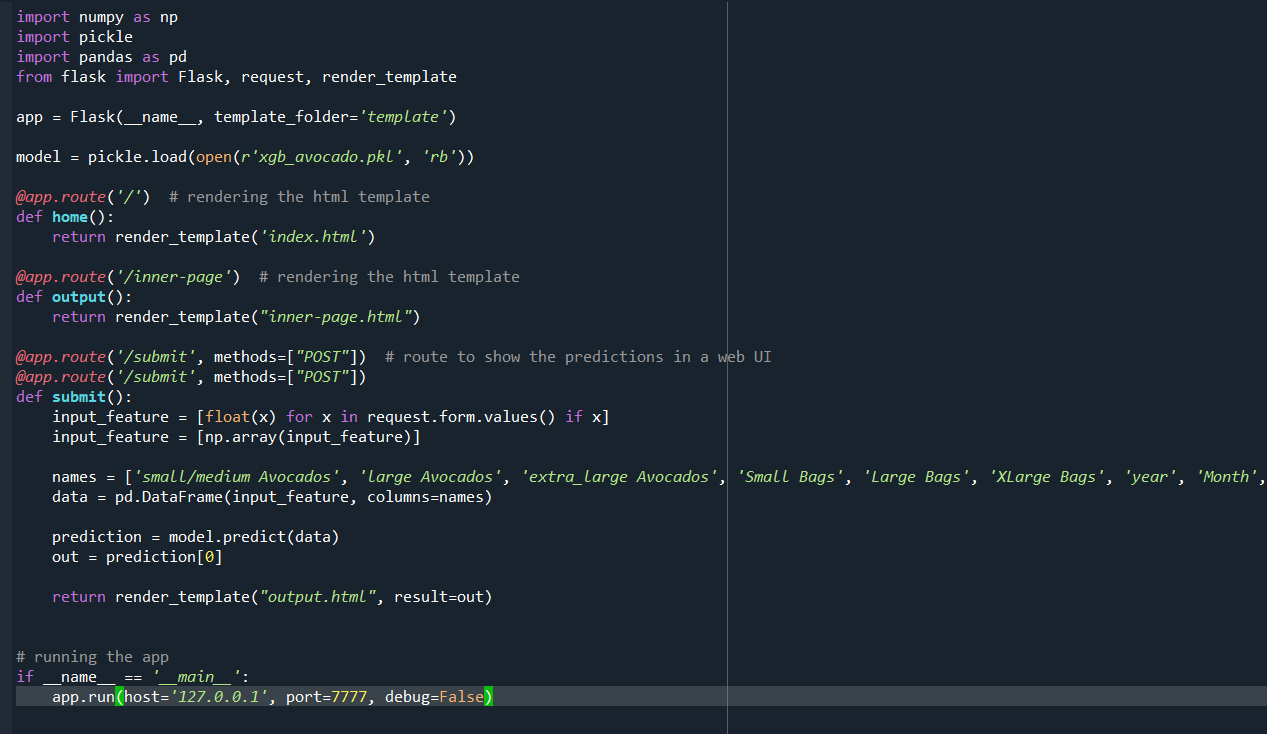


## Activity 3.2: Build Python code

Create a new app.py file which will be store in the Flask folder.

* Import the necessary Libraries.

### Render HTML page:

****

Setting up the application This code defines a Flask web application that uses a pre-trained Xgboost model to predict the status ASD detected or not. The joblib library is used to load the pre-trained Xgboost model from a .pkl file. Then, a route / is defined that renders the index.html template, which contains a form that allows the user to input the features. When the user submits the form, the predict() function is called, which retrieves the input values from the form and computes the Detect ASD. Then, the Random Forest model is used to predict based on the input values, and the predicted status is returned as the output.

## Activity 4: GUI:

The GUI (Graphical User Interface) created in this Flask application is designed to predict Avocado price prediction bases on below features:

🡪small/medium Avocados

🡪large Avocados

🡪extra\_large Avocados

🡪Small Bags

🡪Large Bags

🡪XLarge Bags

🡪year

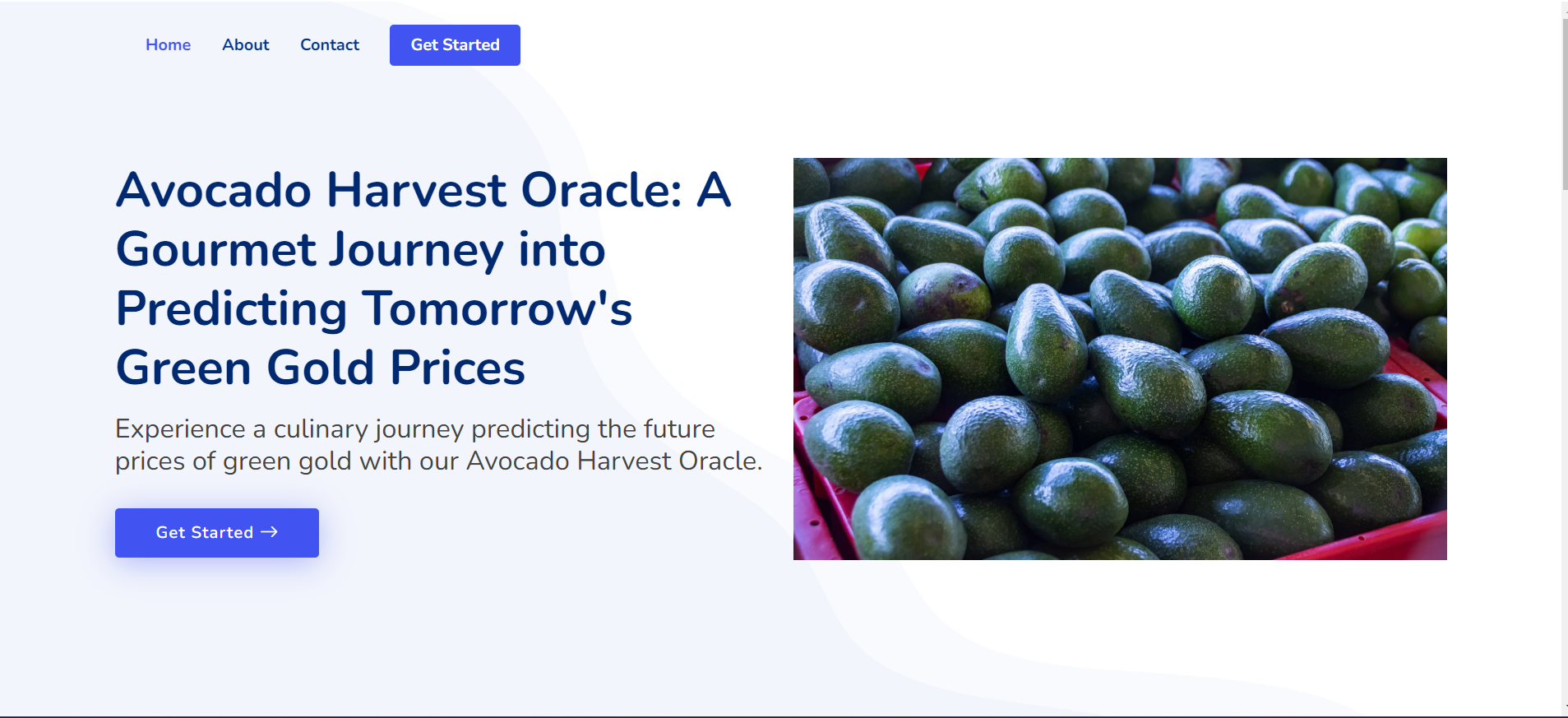
🡪Month

🡪Day

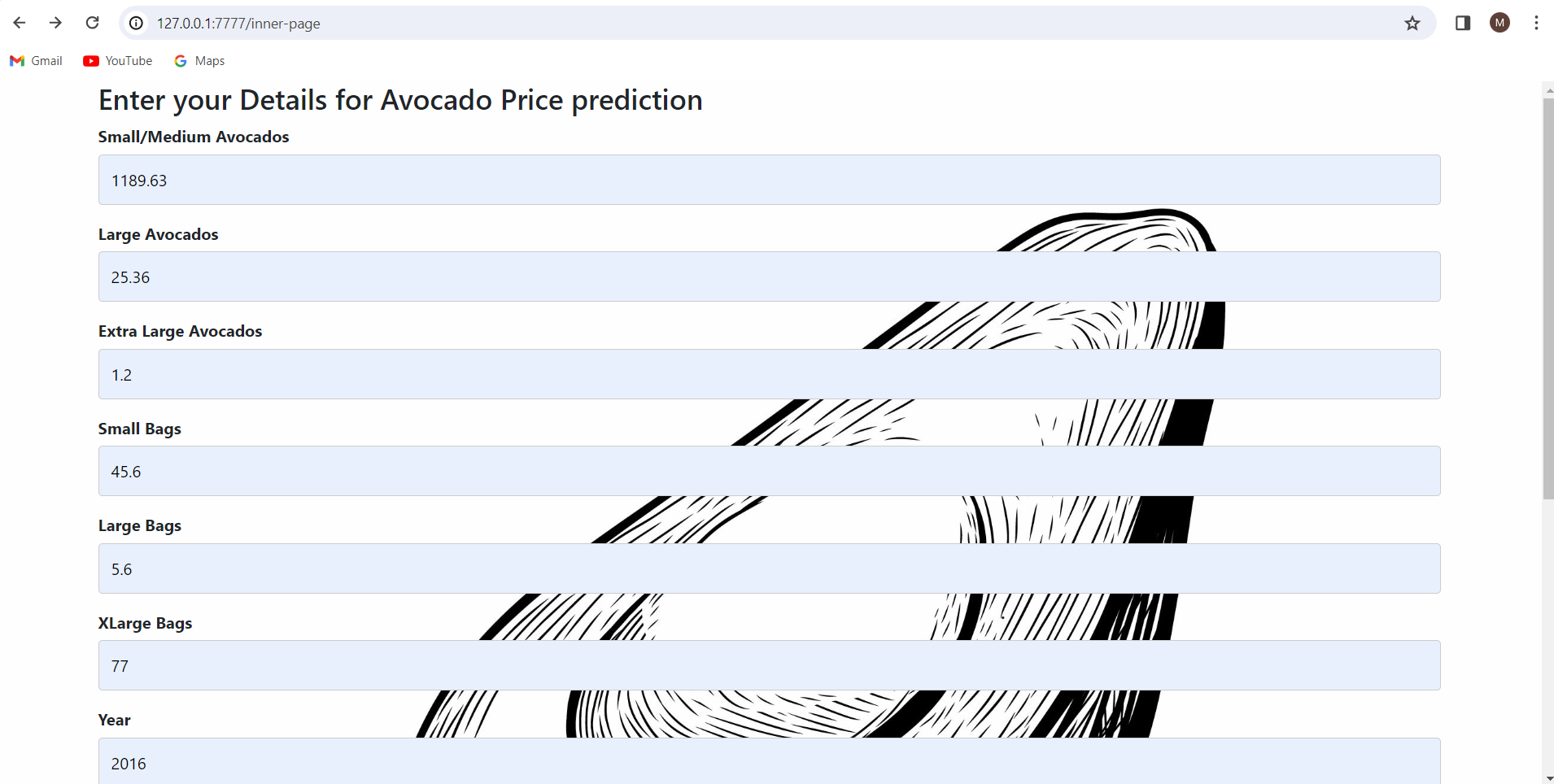
🡪type\_encoded

🡪region\_encoded

The user can input this feature in a form provided in the home page of the web application. After clicking on the "Get Started" button, the application will predict the level of freedom of that country based on the rules and the random forest model defined in the Python script.



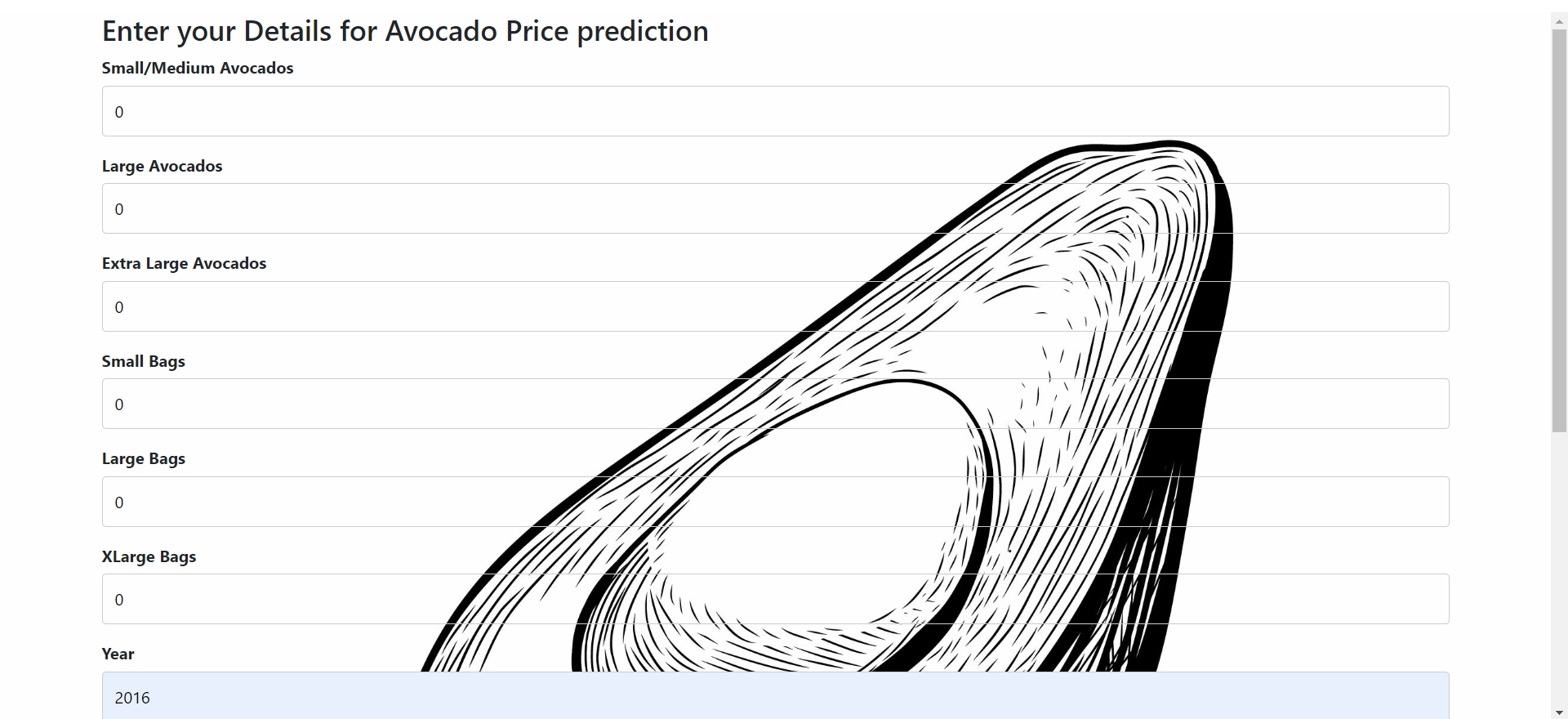
Input 1:



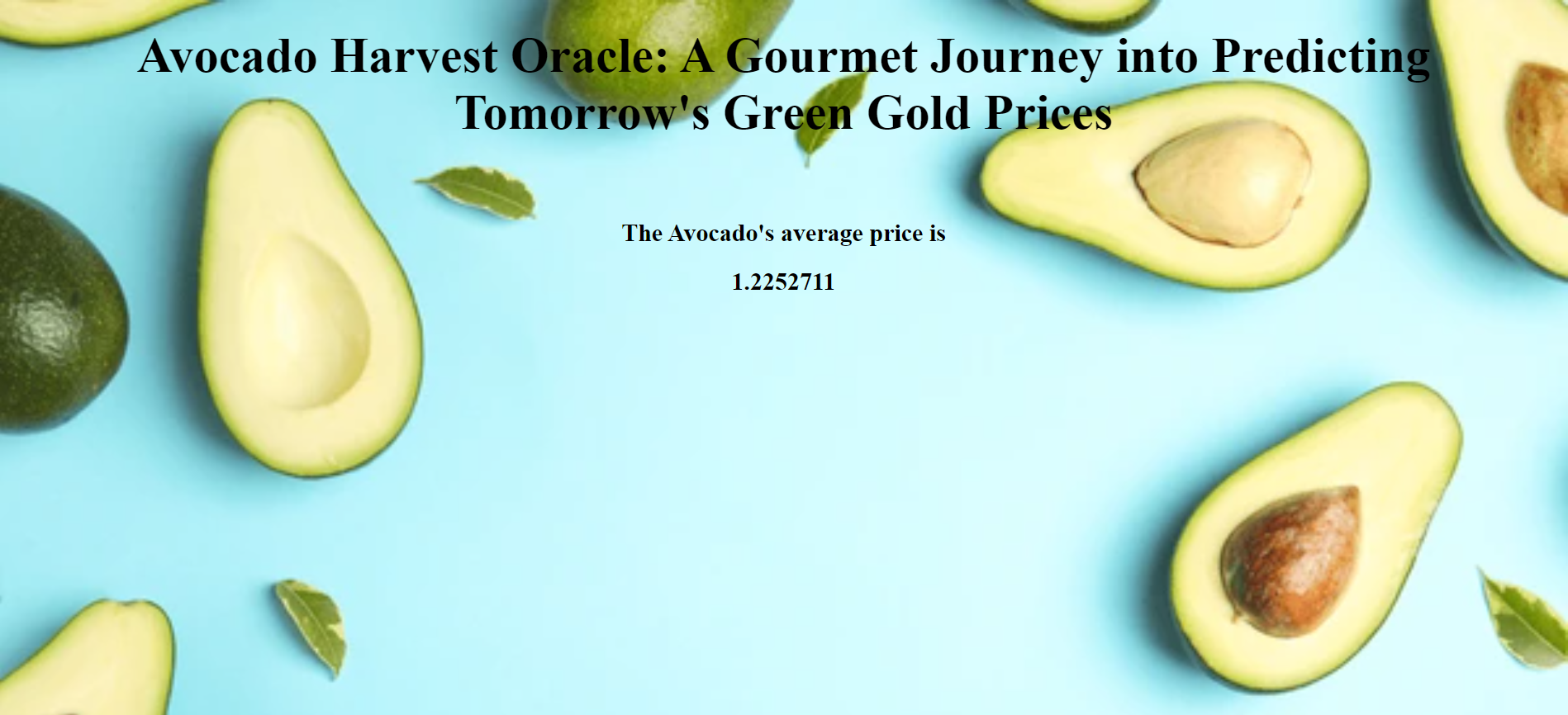
Output 1:



Input 2:



Output 2:



# 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