TEAM-4

LU BO ZHEN

YEO MIN XIU SHERLIN

KHANT ZAW HEIN

NGUYEN BINH TRAN

NANG SHRI KHAM MERNG

WONG CHEE KIAT

JIANG BINGYU

Model Testing

This project contains two models. They are deployed using the Flask app framework and with the help of Waitress web server gateway interface. The available models can be tested with both the Postman and .NET Web Client.

Model 1 is the Logistic Regression Model based on IRIS dataset which gives a prediction on the classification when given 4 input parameters (Septal length, Septal width, Petal length, Petal width).

Model 2 is the Time Series Moving Average Model based on Amazon stock price dataset which gives a prediction of the average stock price with past 50 days data and past 200 days data when given 1 input parameter (Date).

With the use of .NET Web Client built using .NET Core MVC framework, the models can be tested conveniently. When the web client is launched, the home/index page displayed allows for selection of the model for prediction - Logistic Regression or Time Series Moving Average.

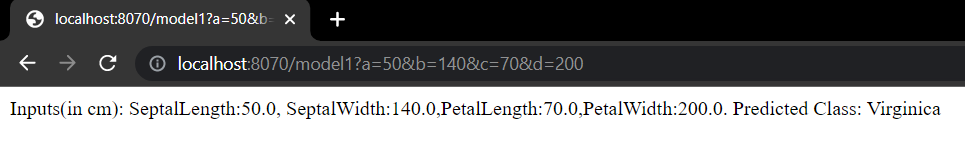
After selecting a model type and click submit, the subsequent model page will be displayed for testing. Then input the required parameters into the input fields and click submit. The predicted value will then be displayed under the output section. To produce another result, change the input parameters accordingly and submit again. It is also possible to click back to return to the model selection home page and change to a different model type for testing.

Valid Inputs for Model Testing

For Model 1 (Logistic Regression Model based on IRIS dataset), the inputs must be positive numbers (can be integer or float). Examples of invalid inputs are string and negative numbers. The input field was set to number field to prevent server response error because of invalid string inputs. If the input value is a negative number, it will not trigger a server response error but will cause the testing and prediction to be invalid.

For Model 2 (Time Series Moving Average based on Amazon stock price dataset), the inputs must follow a date format (e.g., **YYYY-MM-DD**) and need to be a date that is at least 50 days or 200 days later than the start date of the dataset. This is done to ensure that the predicted average displayed is a number. If not, Nan (Not a number) will be displayed as the average accordingly. In addition, the input date must be in the dataset for the model to do the prediction of average. If the input value given is not in the right format or out of the dataset range, predicted average cannot be retrieved and a message indicating date not in dataset will be displayed.

**A. Default testing (Using Flask App and Waitress Deployment)**



1. Using a browser (address bar), enter:
   1. “localhost” with designated server port number
   2. Model name (e.g. model1)
   3. Query string (containing the X variable names and values)
2. Predicted Y value can be directly viewed from the browser

**B. Testing with POSTMAN**

**1. Model-1: Classifying the outcome using Logistic Regression**

* Server port is 8070 in server.py:

Text

Description automatically generated

* Input URL for testing:

Graphical user interface, text, application, email

Description automatically generated

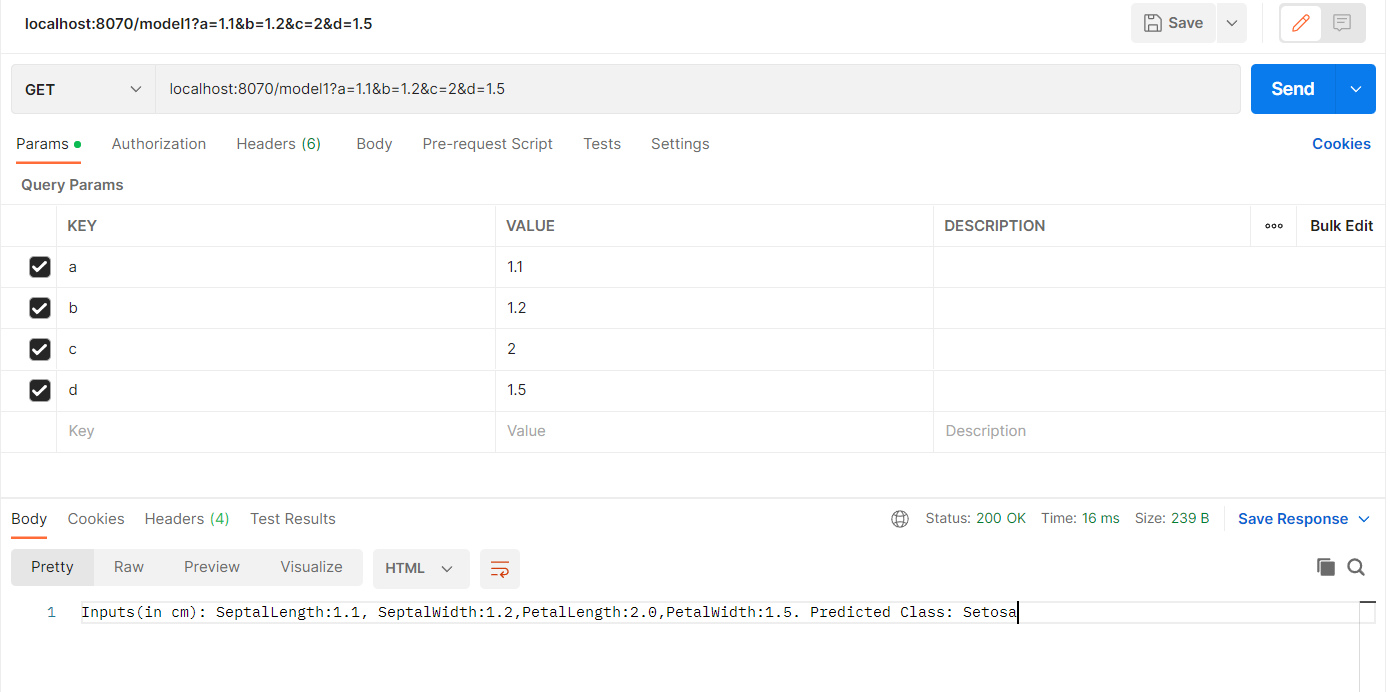
* Providing values to predict the outcome
* Fill-in ‘Keys’ and ‘Values’ in Postman:

A picture containing application

Description automatically generated

* The following result will show up in the HTML body:





**2. Model-2: Predicting average stock price with the Time Series Moving Average Model**

* Input the correct date format to predict:

**Graphical user interface, text, application, email

Description automatically generated**

**Result:**

Graphical user interface, text, application

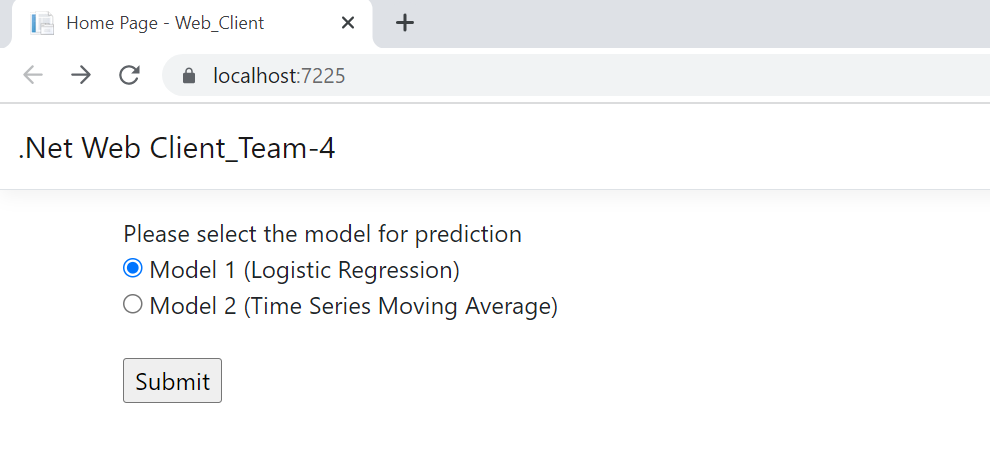
Description automatically generated

* The following outcome will return if the input format is correct, but the date is invalid

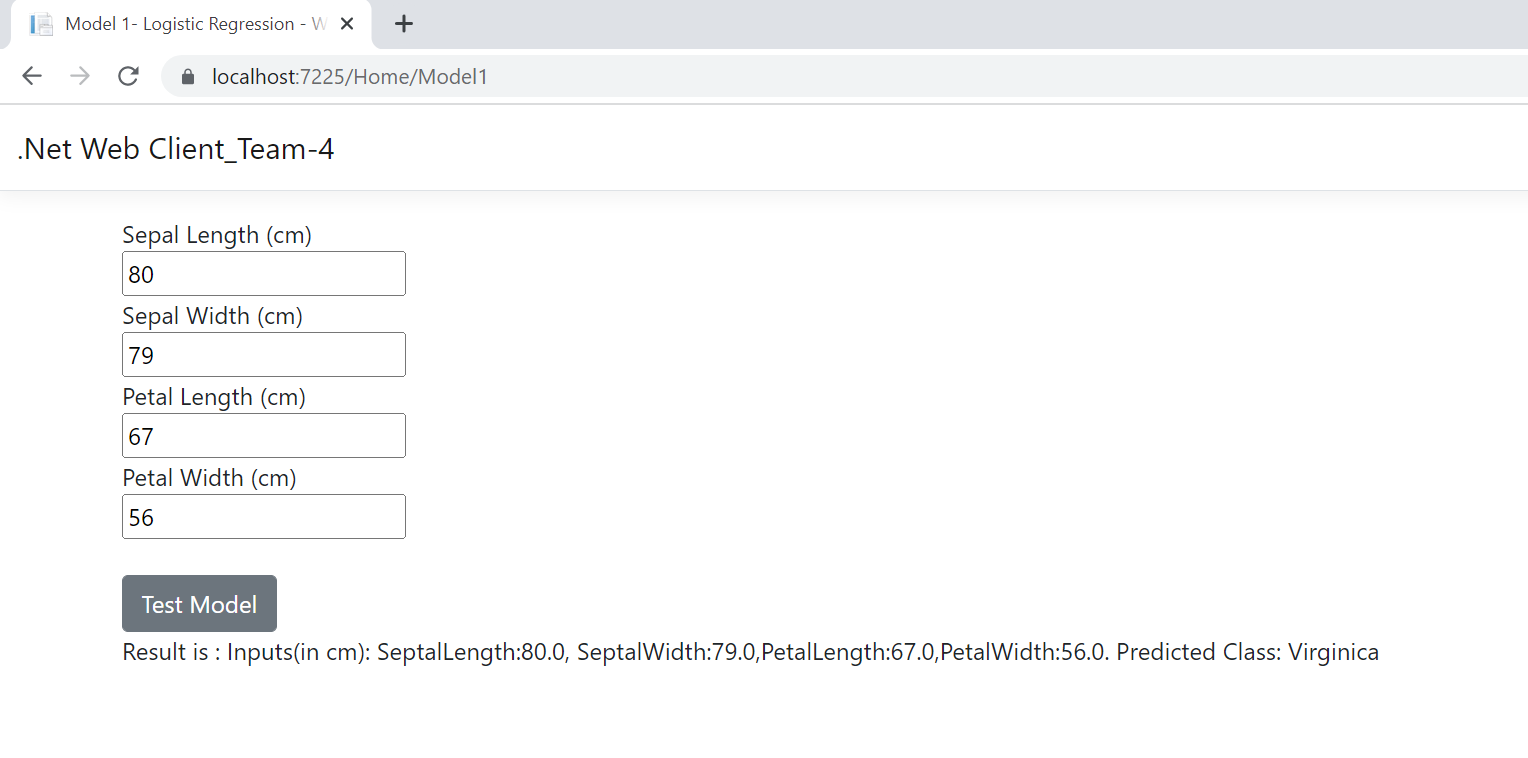
Graphical user interface, text, application, chat or text message

Description automatically generated

**C. Testing with ASP.NET Web Client**



1. Select a model from the main (Home/Index) page



1. On the selected model’s page, enter the X values into the provided input boxes
2. Upon clicking “Test Model”, the HTML form values will be submitted and used as inputs to send HTTP Get Request to the Flask App
3. A page with the predicted Y result will be returned, based on Http Response from the Flask App

\*Testing method remains the same for Model 2 (Time Series):

Graphical user interface, text, application

Description automatically generated

Model selection

Graphical user interface, text, application, email

Description automatically generated

Input values & returning of results

Graphical user interface, text, application, email

Description automatically generated

Alternative format for date input

Graphical user interface, text, application

Description automatically generated

Example of an unaccepted input