

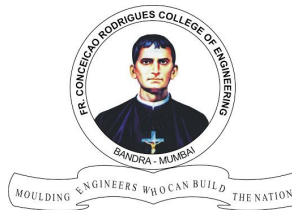
ML Web Application for Regression

A MINI-PROJECT REPORT

By

Devin Barboza (8315)
Sanfer Noronha (8354)
Benita Rego (8362)
Mayank Srivastava (8370)

Under the guidance of
Prof. Sujata Deshmukh



DEPARTMENT OF COMPUTER ENGINEERING
FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING
FR. AGNEL ASHRAM, BANDRA (W),
MUMBAI - 400 050.

UNIVERSITY OF MUMBAI
(2019 – 2020)

FR. CONCEICAO RODRIGUES COLLEGE OF
ENGINEERING
FR. AGNEL ASHRAM, BANDRA (W),
MUMBAI - 400050.

1. Problem Definition

To design a top-notch web based application to train different types of regression models namely:

- a. Linear Regression
- b. Lasso Regression
- c. Ridge Regression
- d. Bayesian Ridge Regression

and use the same models in order to make predictions for any dataset given by the user.

The Web application can also be used to visualize the datasets and derive key insights from it.

2. Requirements

- a. **Node.js:** Node js is used in order to create the server for this web app. It is recommended to have the latest version of the same installed.

The following node packages are used in the development of web application:

- I. Vue.js with Vuetify: Vue.js is a framework of JavaScript which is used for developing the front end of web application.
- II. Axios: Axios is a node package used for Cross Origin Resource Sharing, used for connecting Python with the designed front end using REST APIs.

- b. **Python:** The system will require python installed, along with the following libraries:

- i. Sci-kit learn
- ii. Os
- iii. Matplotlib
- iv. Seaborn
- v. Pandas
- vi. Numpy
- vii. Flask: Flask is used in order to run the python code for the models in the backend and to return the results to the Node js Server.

3. Working

Steps to run the project:

1) In the folder named "web-app", run "npm install" (this is to install any library from npm needed)

Then run the command "npm run serve" (to run the web app server)

2) In the folder named "api", run "pip install -r requirements.txt" (this is to install any python libraries for flask server)

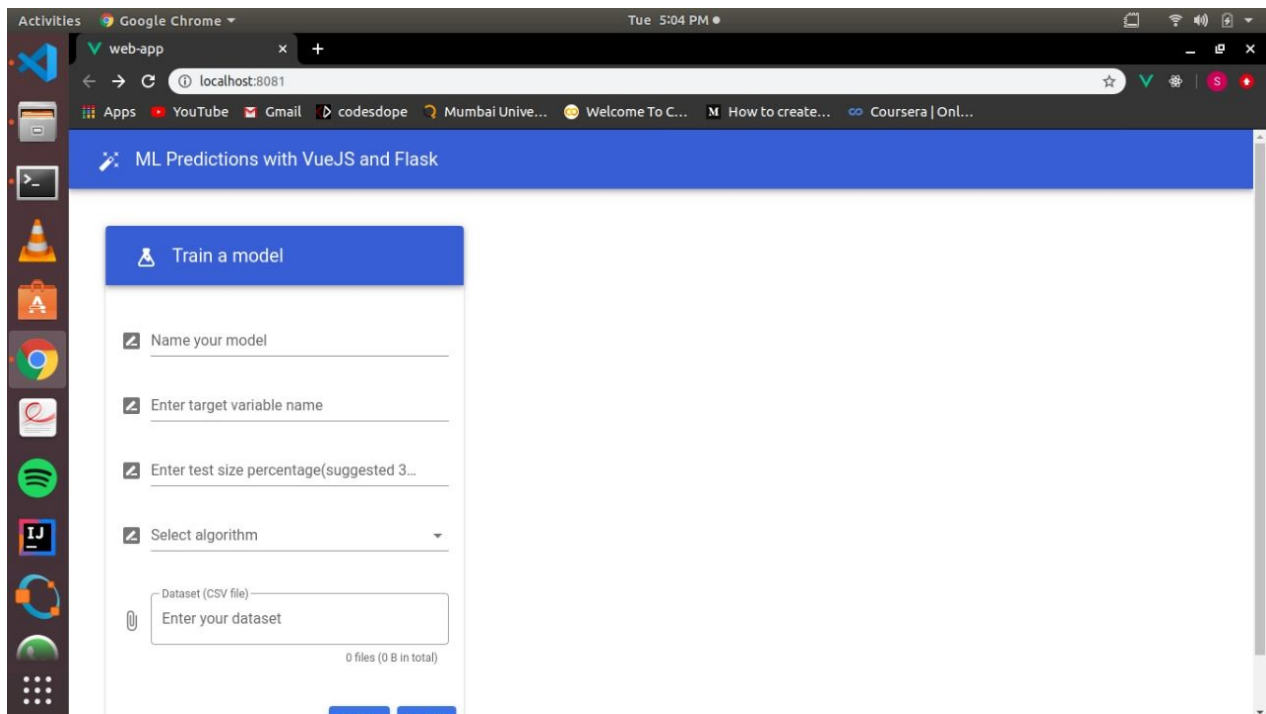
Then run the command "python app.py" (to run the flask server)

4. Link

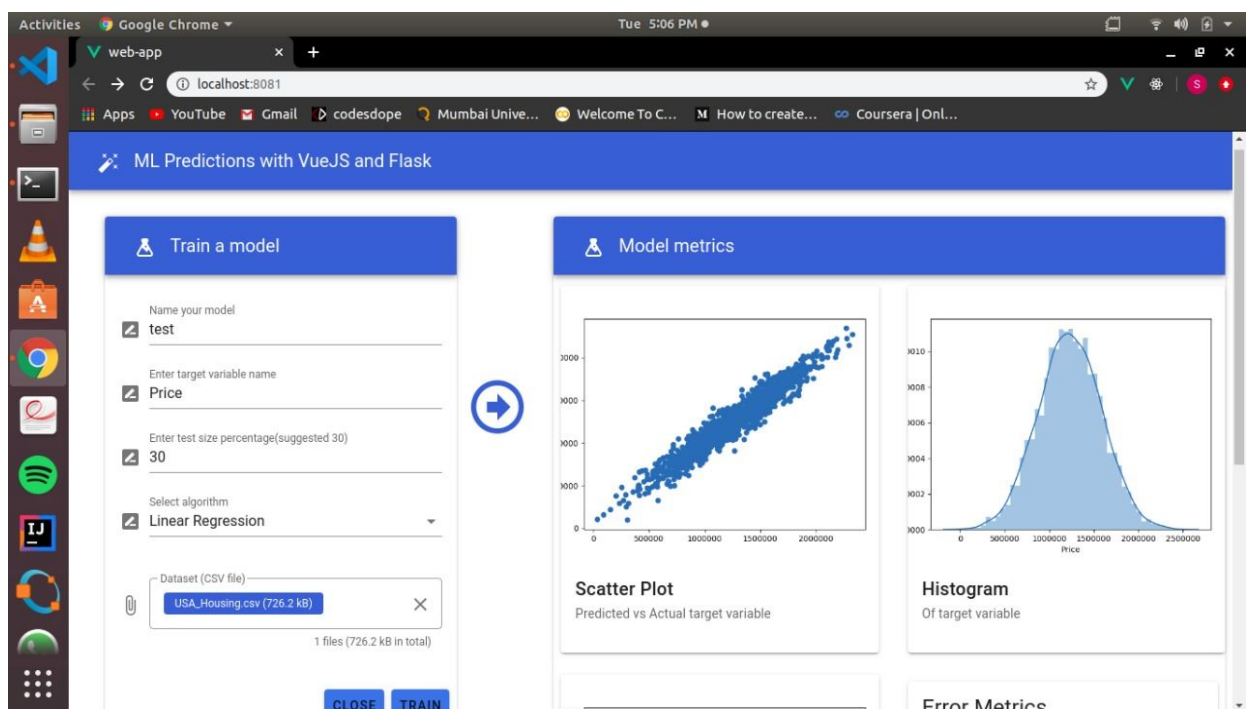
<https://github.com/sanfernoronha/ml-examples-vuejs-flask>

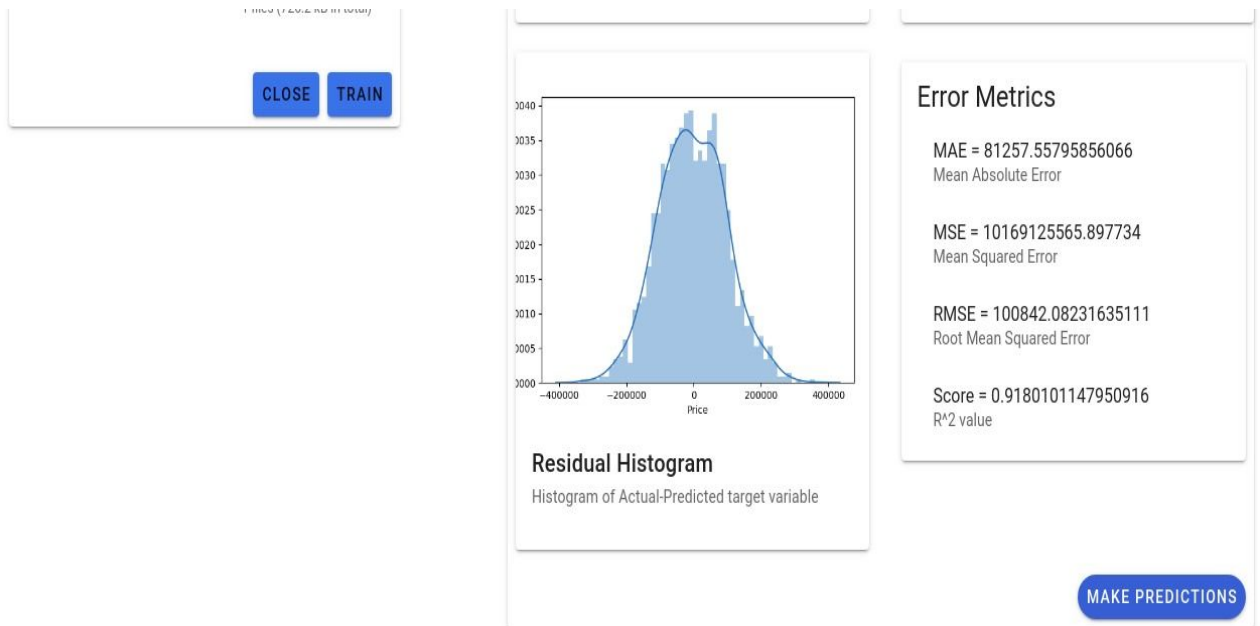
5. Screenshots

Beginning with selection of type of Model you need, specify test size and select the dataset .csv file:



On clicking on train the model gets trained and Model Metrics along with the results of Exploratory Data Analysis is done:





On Clicking “Make Predictions”, user needs to input data for the required features:

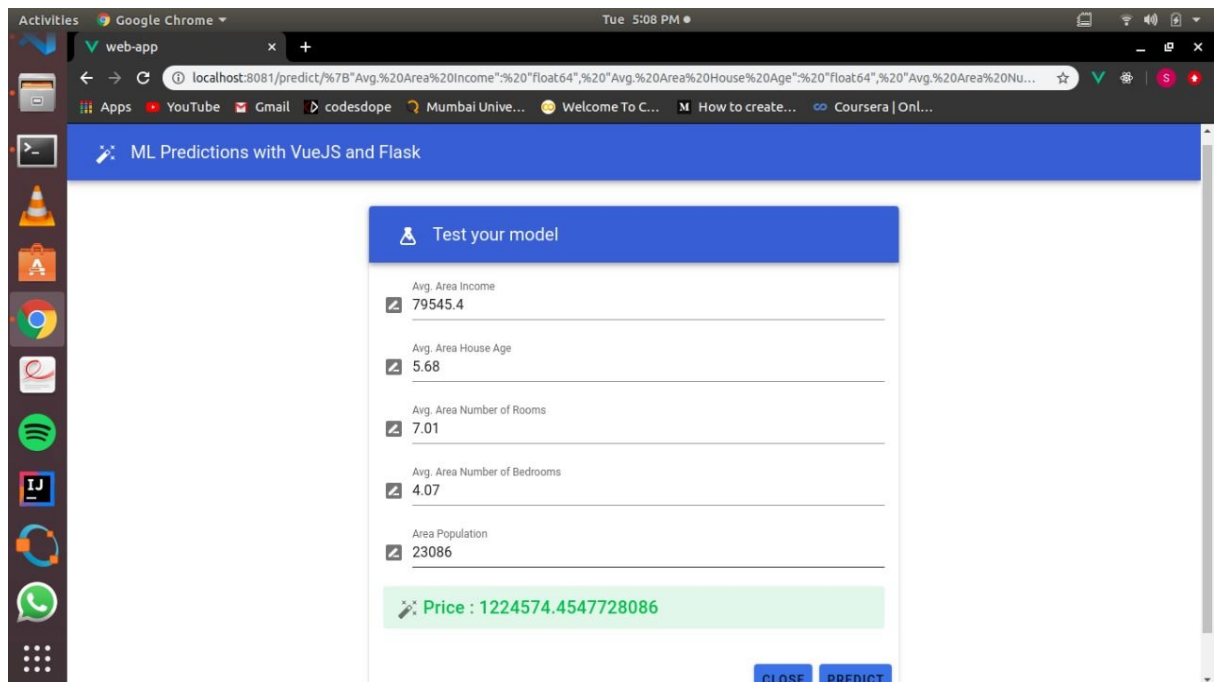
The image shows a web application interface in a Google Chrome browser window. The browser's address bar displays the URL: `localhost:8081/predict/%7B%20Avg.%20Session%20Length%20float64%20Time%20on%20App%20float64%20Time%20on%20Website%20float64%20Length%20of%20Membership%20float64%7D`. The page title is "ML Predictions with VueJS and Flask".

The main content area features a "Test your model" form with the following fields:

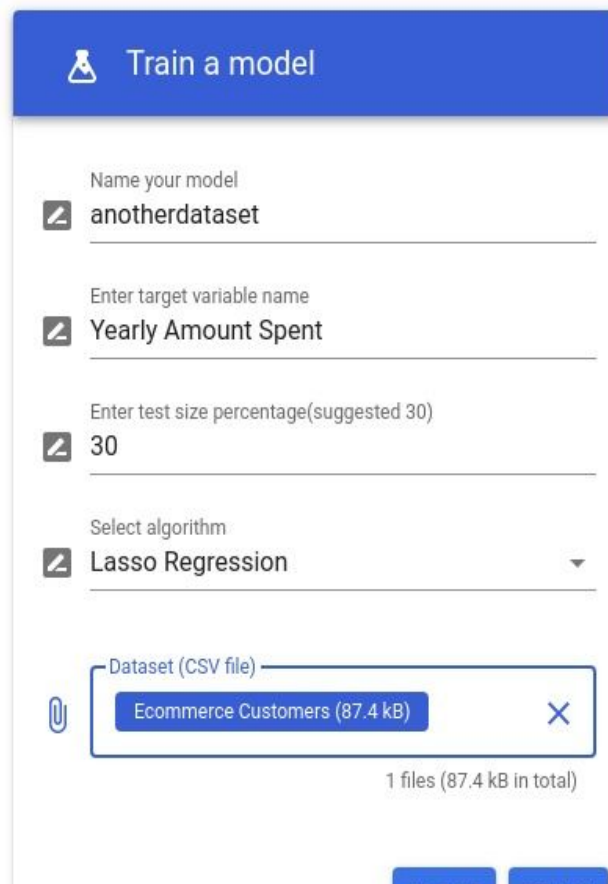
- ☒ Avg. Session Length
- ☒ Time on App
- ☒ Time on Website
- ☒ Length of Membership

At the bottom of the form, there are two buttons: "CLOSE" and "PREDICT".

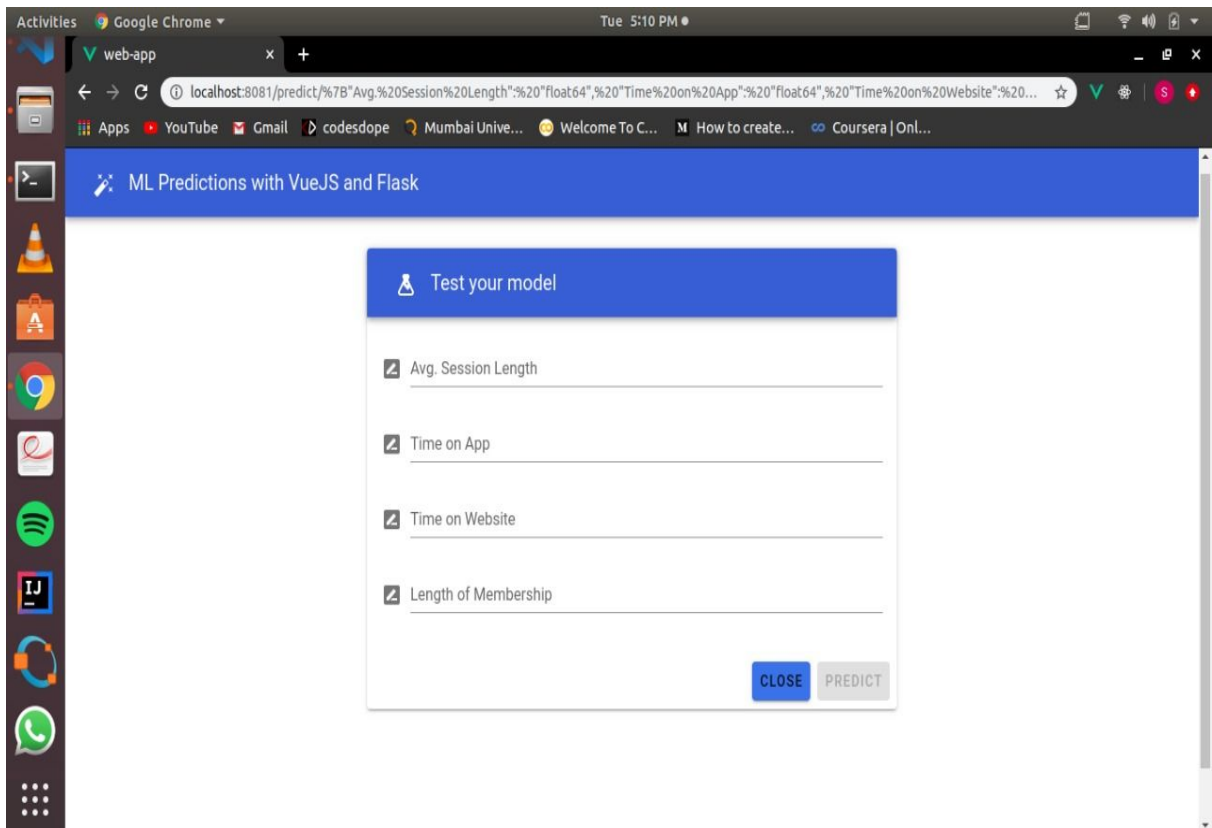
Once user has filled the data into the given fields on clicking on Predict, Predictions for Target variable is done:



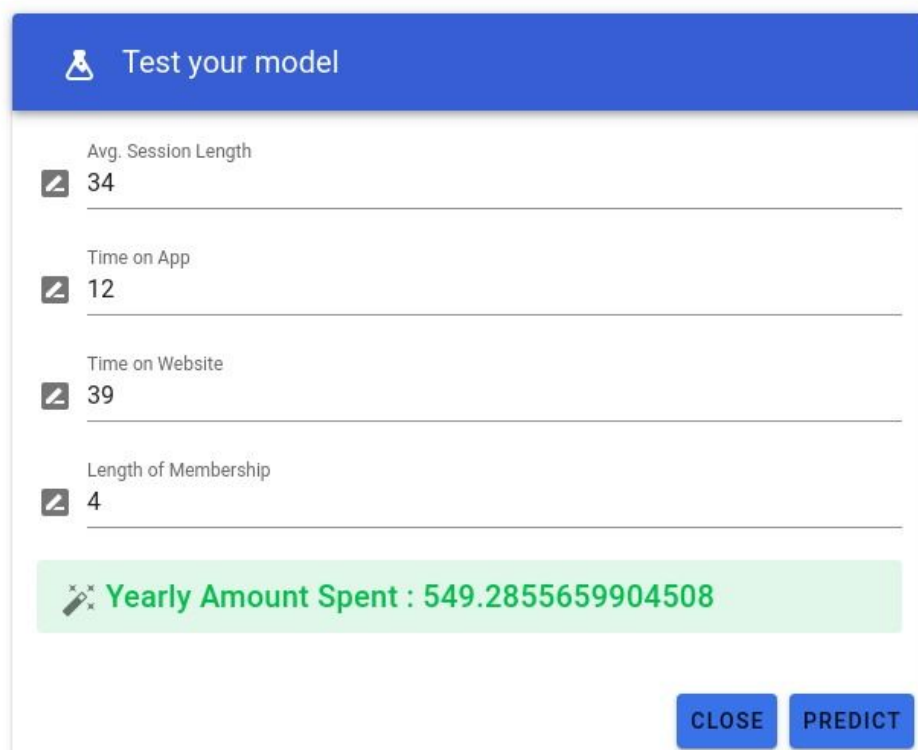
On selection of different dataset and different Regression Algorithm:



After training is completed and we click on “Make Predictions” the fields as we see are being generated dynamically depending on the number of features in the provided dataset :



Finally the predictions are made the target variable is also generated dynamically



6. Flow Diagram

