AutoML App for Learning/Deploying Machine Learning Models

A PROJECT REPORT

Submitted by

Dharineesh Karthikeyan
CB.SC.I5DAS18006

in partial fulfilment of the requirements for the award of the degree of

INTEGRATED MASTER OF SCIENCE

IN
DATA SCIENCE

AMRITA SCHOOL OF ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE 641112

APRIL 2020

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF ENGINEERING, COIMBATORE, 641112

BONAFIDE CERTIFICATE

This is to certify that the project report entitled **“AutoML App for Learning/Deploying Machine Learning Models”** submitted by **CB.SC.I5DAS18006 Dharineesh Karthikeyan** in partial fulfillment of the requirements for the award of the **Degree of Integrated Master of Science** in **DATA SCIENCE** is a bonafide record of the work carried out under my guidance and supervision at Amrita School of Engineering, Coimbatore.

Class Advisor	Project Coordinator
Designation	Designation

Chairperson
Department of Mathematics
Dr. J. Ravichandran

The project was evaluated by us on:

Internal Examiner	External Examiner

AMRITA SCHOOL OF ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE - 641 112

DEPARTMENT OF MATHEMATICS

DECLARATION

I, **Dharineesh Karthikeyan** (Register Number-**CB.SC.I5DAS18006**), hereby declare that this dissertation entitled **AutoML App for Learning/Deploying Machine Learning models**, is the record of the original work done by me. To the best of knowledge this work has not formed the basis for the award of any degree/diploma/associateship/fellowship/or a similar award to any candidate in any University.

Dharineesh

Place: Coimbatore

Signature of the Student

Date:06-06-2021

COUNTERSIGNED

Dr. Prakash P

Project Advisors

Department of Mathematics

ACKNOWLEDGEMENTS

I would like to express my special thanks of gratitude to Dr. Ravichandran J, Chairperson, Department of Mathematics, Amrita School of Engineering, for giving me the golden opportunity to do this wonderful project on the topic of AutoML, this project helped me to learn and discover about so many new things.

I would also like to thank my class advisor, Dr. Prakash P, Department of Mathematics, Amrita School of Engineering, for supporting me with this project idea and guiding me through out.

I would like to thank the faculty members of Department of Mathematics for allowing me to work for this Project.

My heartfelt thanks to all the staffs for their invaluable teachings over the years and constant inspiration for my project work.

I owe a special debt gratitude to my revered parents and family for their blessings and inspirations.

Coimbatore, Dharineesh Karthikeyan

June 2022

CONTENTS

1. Introduction

1.1 What is AutoML?

AutoML stands for Automated Machine Learning which provides methods and processes to make Machine Learning available for non-Machine Learning experts, to improve efficiency of Machine Learning and to accelerate research on Machine Learning. By automating the time-consuming, iterative tasks of machine learning model development. It allows data scientists, analysts, and developers to build ML models with high scale, efficiency, and productivity all while sustaining model quality.

The success of a machine learning (ML) model crucially relies on human machine learning experts to perform the following tasks:

- Pre-process and clean the data.
- Select and construct appropriate features.
- Select an appropriate model.
- Optimize model hyperparameters.
- Critically analyse the results obtained.

When trying to build an AutoML system we need to take into considerations all of the above tasks and make sure they can be automated and made easy for a non-Machine learning expert to understand and build machine learning models on any data needed.

AutoML is considered to be the future of machine learning/artificial intelligence industry. It is a huge area of research at the moment, as every major IT organization such as Google, Microsoft, etc… are working on building an AutoML app. Thus, this project idea has a lot of scope in the real-world scenario.

1.2 Aim of my AutoML App:

The main idea is to make an application using which the user can upload a dataset, perform all the needed tasks such as feature selection, pre-processing, model selection and hyperparameter tuning with the help of just a few buttons clicks, using a responsive and simple UI (user interface).

The app is built with both learning/developing machine learning models with AutoML in mind, which is achieved by providing the users with ample user interface (UI) options to click and select from to build a complete machine learning model.

But the app is more apt for a teaching/learning focus, where beginners in ML can learn about the different stages and step by step processes needed to build a ML model, and more intermediate people can focus on experimenting with the more minute details such as hyper parameter tuning of the model

The app can also be used for developing and deploying machine learning models for industrial/real-world problems but the issue is that the computational and storage requirements for high-level project cannot be met using this app. But it can still develop machine learning models for simpler use cases with ease, such as projects made by students to practice their machine learning skills.

1.3 Benefits of the App Idea:

As mentioned above, every IT organization is working on a AutoML and the difficulty behind its success is mentioned below:

- The automation part – since everything must be done without any human intervention
- Deployment of an application that is easily accessible and understandable by any non-machine learning expert.

By working on this project, I got a better understanding on how AutoML works and was able to successfully implement the concepts and deploy an app that is working as expected.

The app will be beneficial to students/teachers who want to teach the concepts of machine learning and provide an enjoyable fun platform via which the concepts can be learnt without any actual need of coding a single line of code.

This app could also be deployed for an Industrial/company use case to make building machine learning for more real-world scenarios as well, but the limitations is that the app I made has a limit on the file size of the data and hence won’t be suited for large data projects. But it is possible to make this possible in the future by integrating the application with a cloud support – to get better storage and computation.

The main benefit of this app, is that it is very scalable and anyone can work on it to make it better. For example, adding new features or models into the existing app framework is quite easy and can by having such a scalable app is quite useful.

---

2. Coding / Implementation

2.1 About the coding part

The App is built fully using python language, where the user design for the web application is designed using a Python Package - "streamlit" and the backend machine learning codes are written using packages such as – pandas, numpy, sklearn, etc...

This app is made to work with tabular data (CSV files).

The AutoML task has been split into –

- Classification task:

  Classification algorithms are used to predict/Classify the discrete values such as Male or Female, True or False, Spam or Not Spam, etc.

- Regression task:

  Regression algorithms are used to predict the continuous values such as price, salary, age, etc.

2.2 Major Challenges Faced

We get the data from the user and the data we get can be of any format, contain any data, columns, datatypes and noisy as well. We should be able to handle the different types of data we can receive, and be able to clean and obtain the important information from the data automatically. We should be able to identify all the categorical data and encode them accordingly, also remember them for future reference.

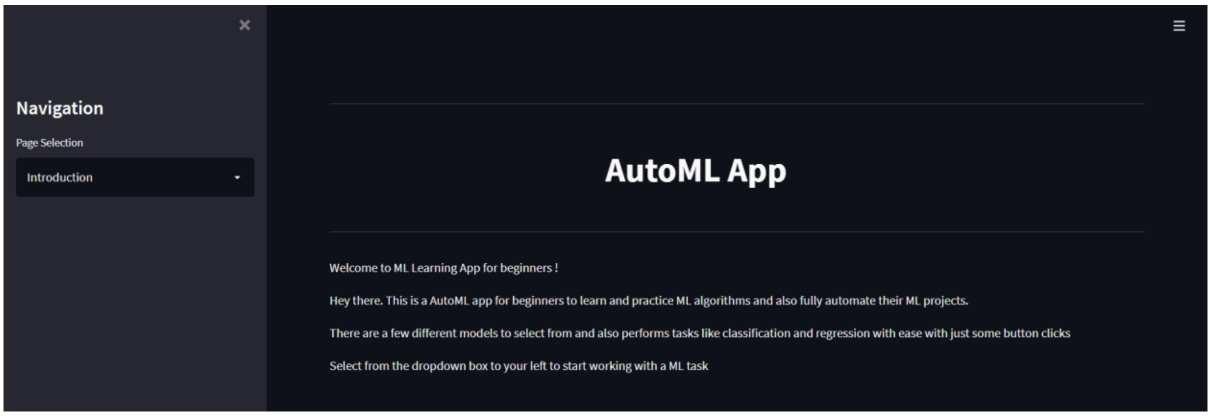
Every task in the machine learning pipelining has to be done automatically and the options must be enriched with variety for the learner to learn and experience.

The web page is the user interface which is the most important part that links the user's selection of choices with the backend machine learning codes that follow them. The front end is built using Streamlit, which it is a stateless app. Hence, if the app reloads there is a chance for it to forget all the values such as the feature selection, hyper parameter value selected. Hence, the app has to be tested multiple times to make sure no important information is lost during entering any value.

3. Logical Approach of the app

3.1 Introduction Page

The focus of the project is to build a user-friendly app that the user can easily access and work with. For that purpose, the app is built with the user ease of understanding and using the app in mind. Whenever the user loads the app, he is first greeted with a simple welcome page and directs the user to the next upcoming pages depending on the user's needs.



AutoML App

Welcome to ML Learning App for beginners !

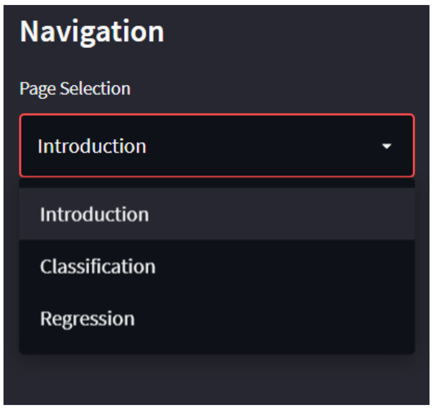
Hey there. This is a AutoML app for beginners to learn and practice ML algorithms and also fully automate their ML projects.

There are a few different models to select from and also performs tasks like classification and regression with ease with just some button clicks

Select from the dropdown box to your left to start working with a ML task

The user depending on his needs/choice can choose if he wants to perform a classification or regression task.

On the left side of the introduction page, the user can select the intended task – classification or regression with the help of the dropdown box provided.



3.2 AutoML Tasks Split Up

The AutoML task has been identified and split into 7 parts. They are the following:

1. Dataset Selection

   The user has to upload the data to work on in this step.

2. Target Variable and Feature selection

   The user has to mention the target column for the AutoML to work on it.

   Also, feature selection can be performed here where the user is provided with an option to select just the needed/best features instead of selecting all of them.

3. Pre-processing data and Train-test split

   The user is provided with a list of options to select from to perform pre-processing on the data such as various normalization techniques such as Min-max scalar, Standard Scalar.

   Next, the user can split the train-test data here to test the performance of the model at the later stages.

4. Classification/Regression Model selection

   Depending on the task, we have a few models from which the user can select any one to work on the data and check its performance.

5. Hyperparameter tunings

Each model has a few hyperparameters which affect the performance by a lot. The user is provided with the option to modify the hyperparameters can check how the performance is affected by it. This is a very important and difficult step for even trained machine learning experts.

6. Plotting (Only for classification task)

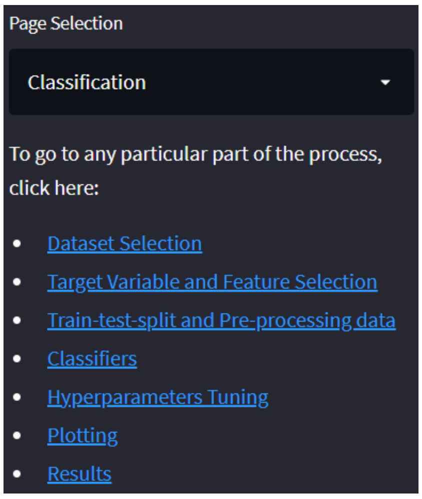
   In the case of classification task, we can actually plot certain graphs related to the performance such as a confusion matrix (in a form of a neat diagram), ROC and Precison-Recall curve to better understand how the classification model is working.

7. Performance Results

   In any machine learning application, the most important thing is the results of how the model is performing. This helps us create benchmarks for the performance, and also compare how the model is performing between changes/updates to the AutoML pipeline.

In order to make sure that the user is comfortable with the app. The app has an inbuilt indexing for all the steps in the process for classification/regression task. Using this feature, the user can freely move in between sections and update anything at any given moment.

Classification

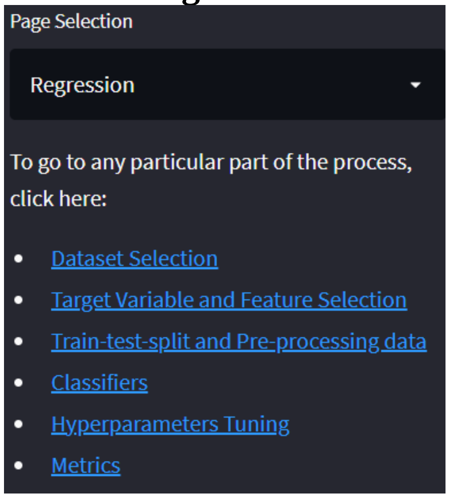

Page Selection

Classification

To go to any particular part of the process, click here:

- Dataset Selection
- Target Variable and Feature Selection
- Train-test-split and Pre-processing data
- Classifiers
- Hyperparameters Tuning
- Plotting
- Results

Regression


Page Selection

Regression

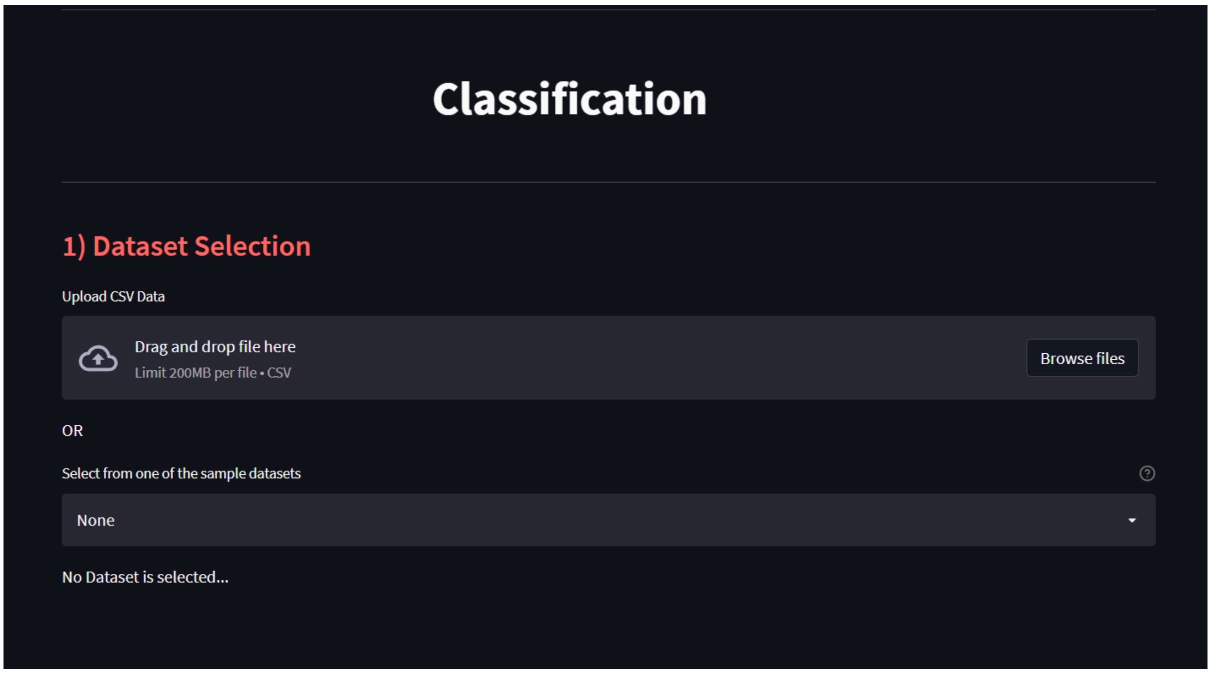
To go to any particular part of the process, click here:

- Dataset Selection
- Target Variable and Feature Selection
- Train-test-split and Pre-processing data
- Classifiers
- Hyperparameters Tuning
- Metrics

4. Working of App – Classification Task

If the user selects the classification task, they are requested to upload a dataset. Since, the app has an educational focus – the users are also provided with a few sample datasets to select from.

The next steps of the AutoML pipeline follows only if the uploaded dataset is of the correct format and bug free.


Classification

1) Dataset Selection

Upload CSV Data

Drag and drop file here

Limit 200MB per file • CSV

Browse files

OR

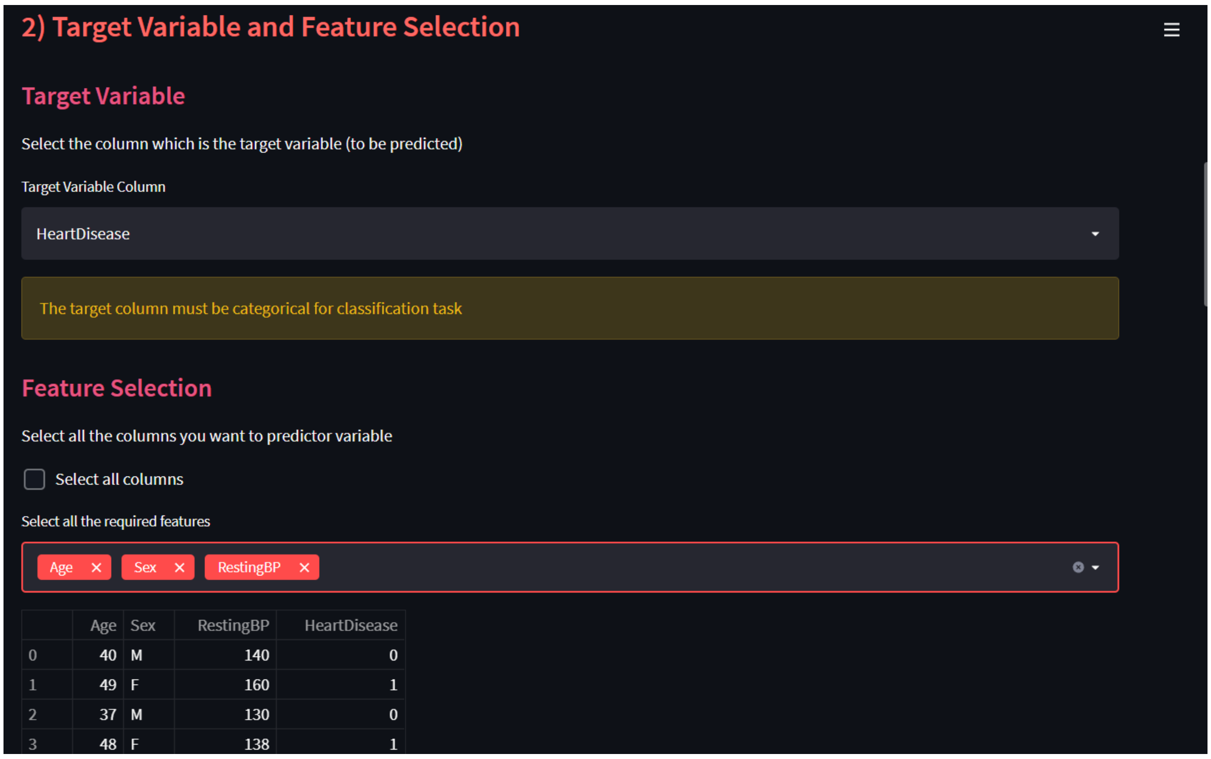
Select from one of the sample datasets

None

No Dataset is selected...

After a dataset has been selected, an option to view the dataset is given to the user and the next 6 steps of the AutoML is displayed to the user.

The next step of selecting the target variables and feature selection. The user is presented with a warning message when selecting the target column in order to make sure that the target variable is a categorical variable.



2) Target Variable and Feature Selection

Target Variable

Select the column which is the target variable (to be predicted)

Target Variable Column

HeartDisease

The target column must be categorical for classification task

Feature Selection

Select all the columns you want to predictor variable

Select all columns

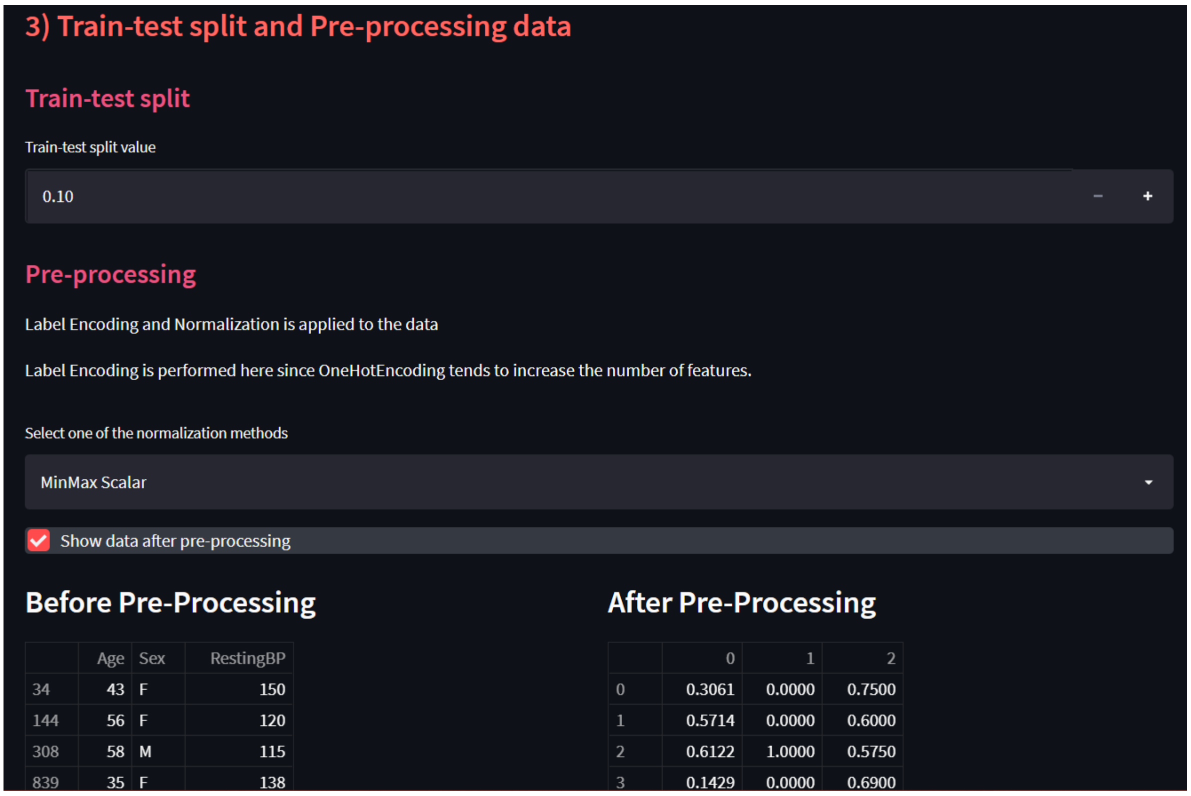
Select all the required features

Age ✕ Sex ✕ RestingBP ✕

	Age	Sex	RestingBP	HeartDisease
0	40	M	140	0
1	49	F	160	1
2	37	M	130	0
3	48	F	138	1

Also, an example of feature selection is shown in the fig. above where instead of selecting all the available features only 3 of the features – age, sex and resting BP are being selected. The user is also able to see the data after feature selection.

The third step of the pipeline, train-test split and pre-processing is where the user mentions the train-test split value and pre-processing is performed depending on the user choice selection. The user is also able to see the data before and after the pre-processing is applied on the data.



3) Train-test split and Pre-processing data

Train-test split

Train-test split value

0.10

Pre-processing

Label Encoding and Normalization is applied to the data

Label Encoding is performed here since OneHotEncoding tends to increase the number of features.

Select one of the normalization methods

MinMax Scalar

✓ Show data after pre-processing

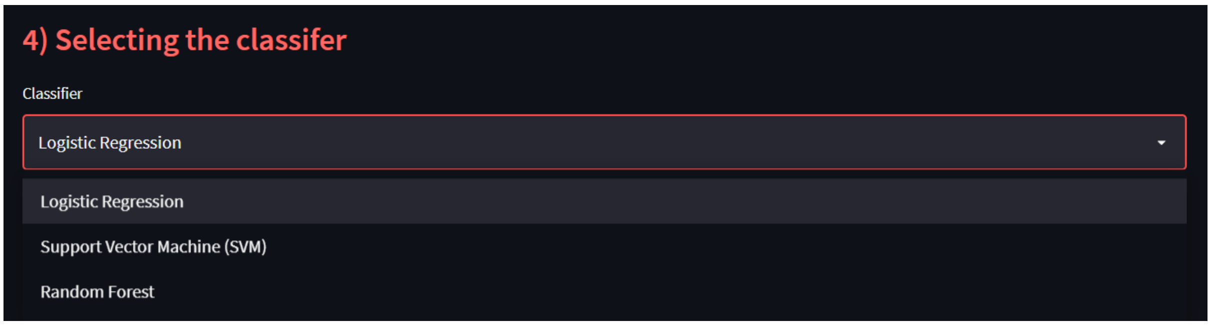
Before Pre-Processing

	Age	Sex	RestingBP
34	43	F	150
144	56	F	120
308	58	M	115
839	35	F	138

After Pre-Processing

	0	1	2
0	0.3061	0.0000	0.7500
1	0.5714	0.0000	0.6000
2	0.6122	1.0000	0.5750
3	0.1429	0.0000	0.6900

After the data is ready to be fitted with a machine learning model, the user is provided with a list of available models. At the moment, the app supports 3 very famous classification machine learning models – Logistic Regression, Support vector machine, Random Forest.



4) Selecting the classifer

Classifier

Logistic Regression

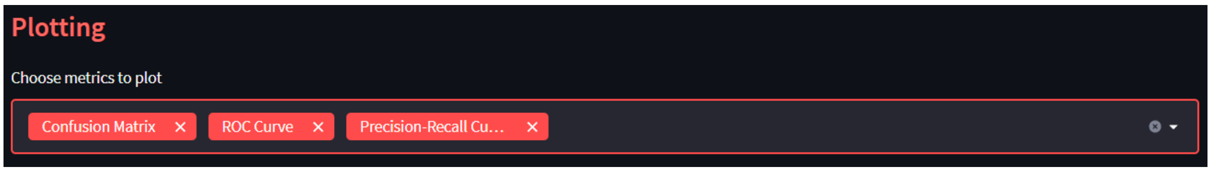
- Logistic Regression
- Support Vector Machine (SVM)
- Random Forest

Based on the selection of the model, the user is provided with the different hyperparameter options available for each model. The user can tune these parameters to their liking, and see how it affects the performance of the model.

Let's have a look at some of the various hyperparameter options provided for each of the model.

Logistic Regression	**SVM**	**Random Forest**
C (Regularization parameter)	C (Regularization parameter)	No. of trees in forest
Max No. of Iterations	Kernel	Criterion
Solver	Gamma	Max depth of tree
Penalty		Max features
		Bootstrap samples
		Out-of bag samples

Next step in classification is the plotting, where the user can select one or more of the available plotting options – Confusion matrix, ROC and Precision-Recall curve.


Plotting

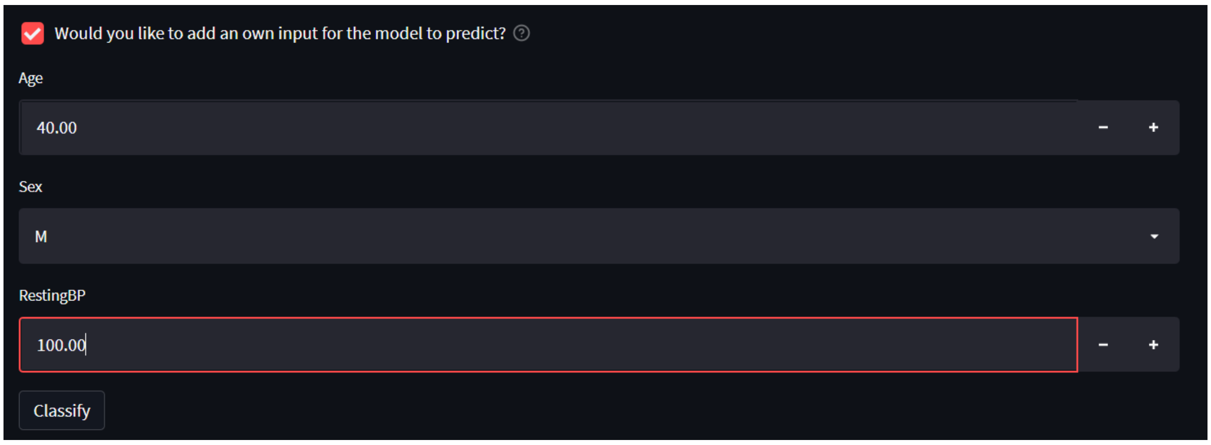
Choose metrics to plot

Confusion Matrix ✕ ROC Curve ✕ Precision-Recall Cu... ✕

The plot results are displayed to the user later along with the performance/metrics of the model.

If the user has any specific test case for which the model prediction is required, the user has the option to select a check box for an added provision of passing the inputs for which the model predicted value will be returned.

Just in case the user doesn't have any such inputs for the model, he can just leave the check box unmarked and classify the model without any test inputs.


✓ Would you like to add an own input for the model to predict?

Age

40.00

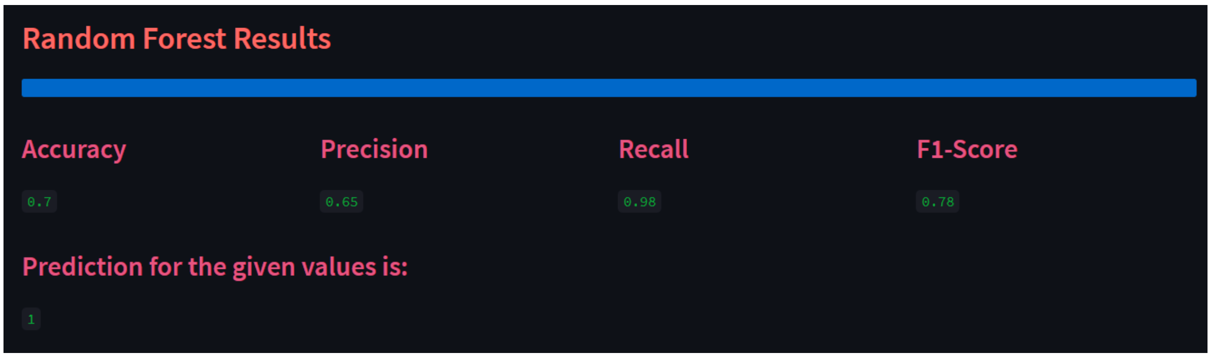
Sex

M

RestingBP

100.00

When the user clicks on the "Classify" button, the model built as per the given user specifications and the performance metrics can be checked by the user.



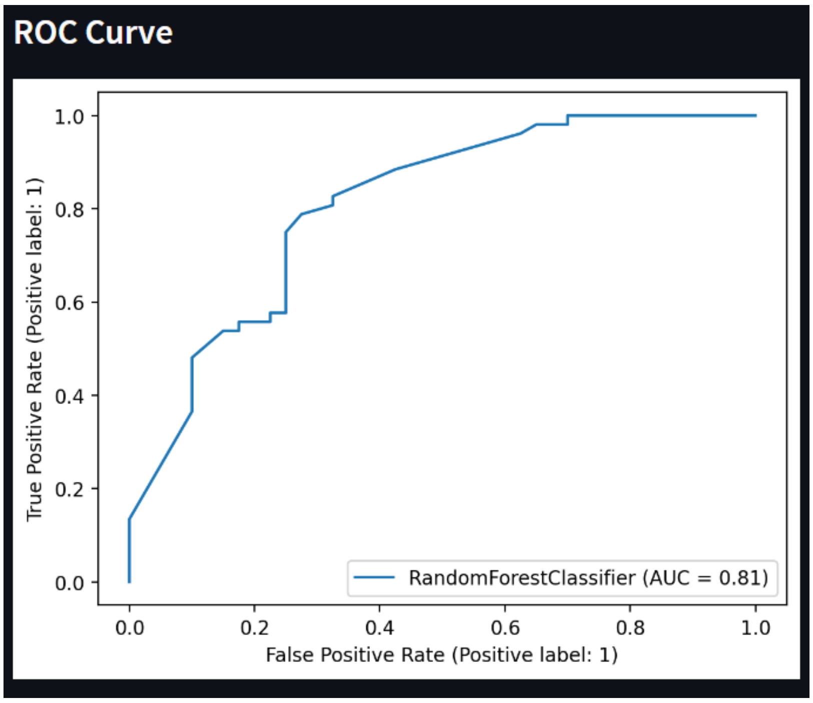
Random Forest Results

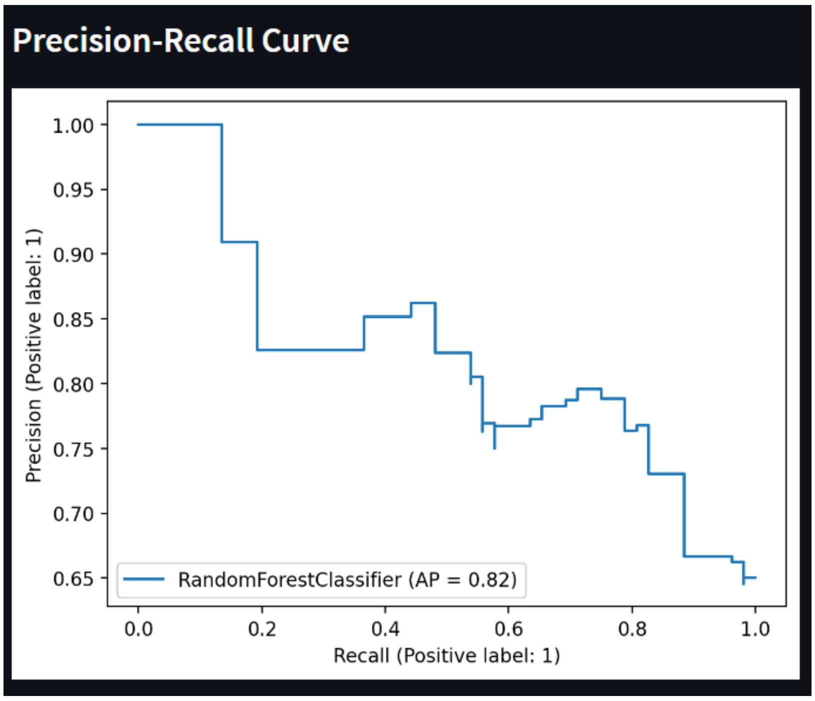
Accuracy	Precision	Recall	F1-Score
0.7	0.65	0.98	0.78

Prediction for the given values is:

```
1
```

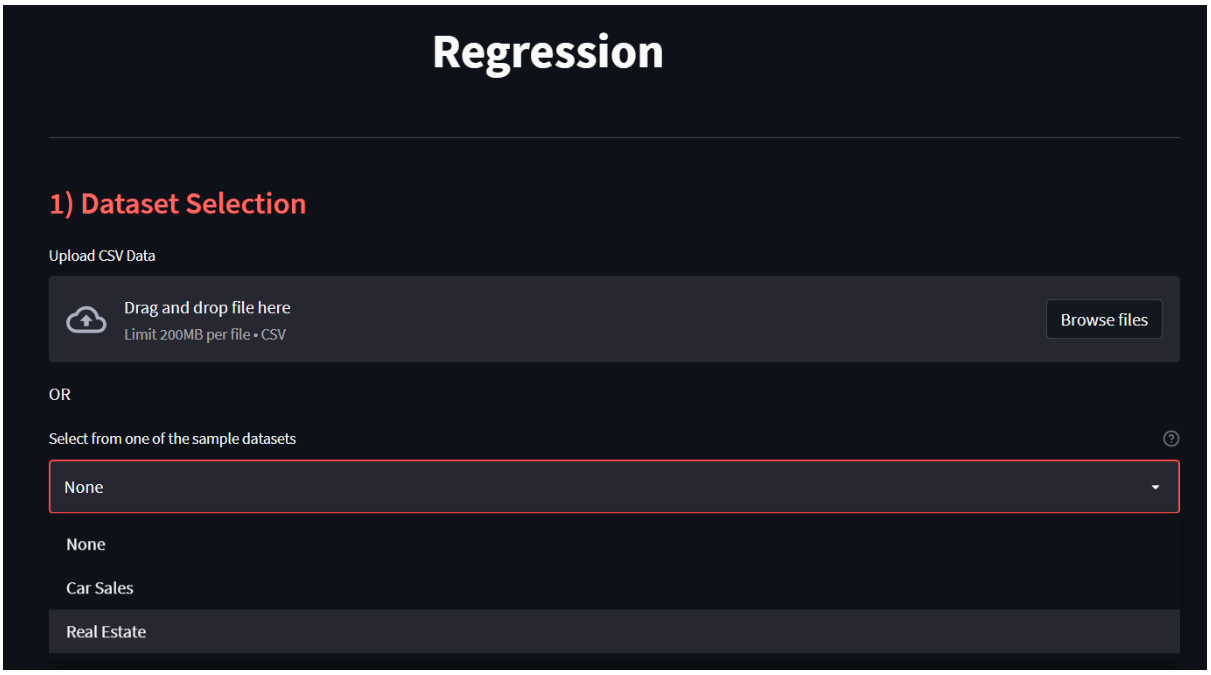
From the above fig, we can see the model performance metrics and make an assessment about how the model is performing. In addition to this, if we had passed any plots during the plotting step of the AutoML pipeline. The plots are also displayed below the results as such





5. Working of App – Regression Task

The working of the regression task part of the app is quite similar to the above-mentioned classification task. It starts with the user selecting the regression task from the drop-down box. After which, we go to the regression page. The first step is uploading the dataset or selecting one from the available sample datasets.



Regression

1) Dataset Selection

Upload CSV Data

Drag and drop file here
Limit 200MB per file • CSV

Browse files

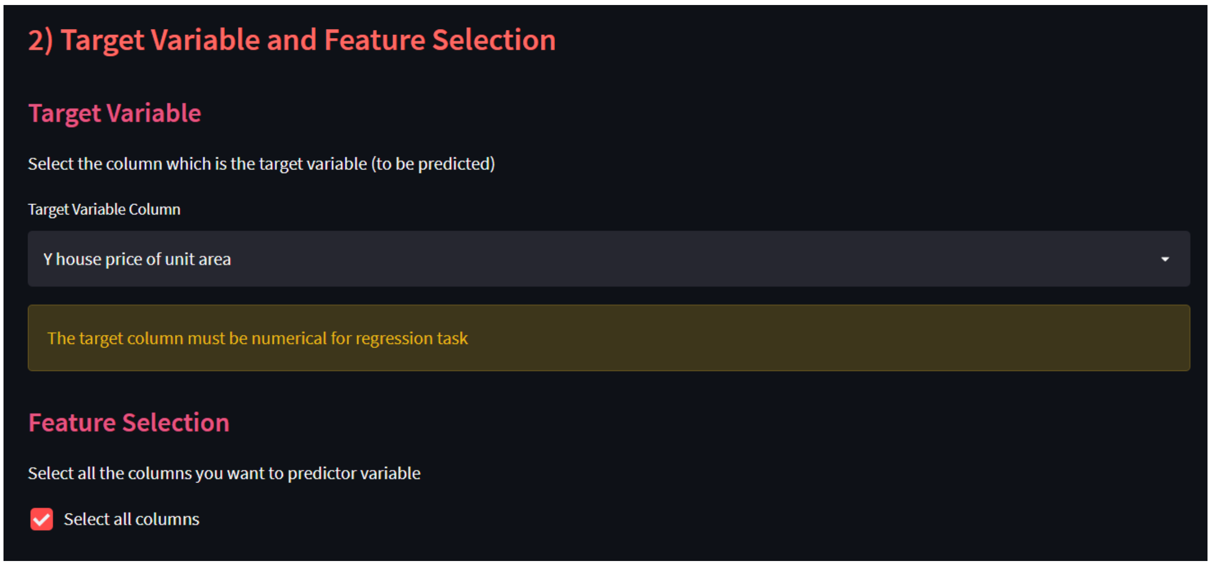
OR

Select from one of the sample datasets

None

None
Car Sales
Real Estate

In the next step of target variable and feature selection, the only difference is that the warning message displayed to the user requests the user to select a target column which is numerical since the task to be performed is regression.



2) Target Variable and Feature Selection

Target Variable

Select the column which is the target variable (to be predicted)

Target Variable Column: Y house price of unit area

The target column must be numerical for regression task

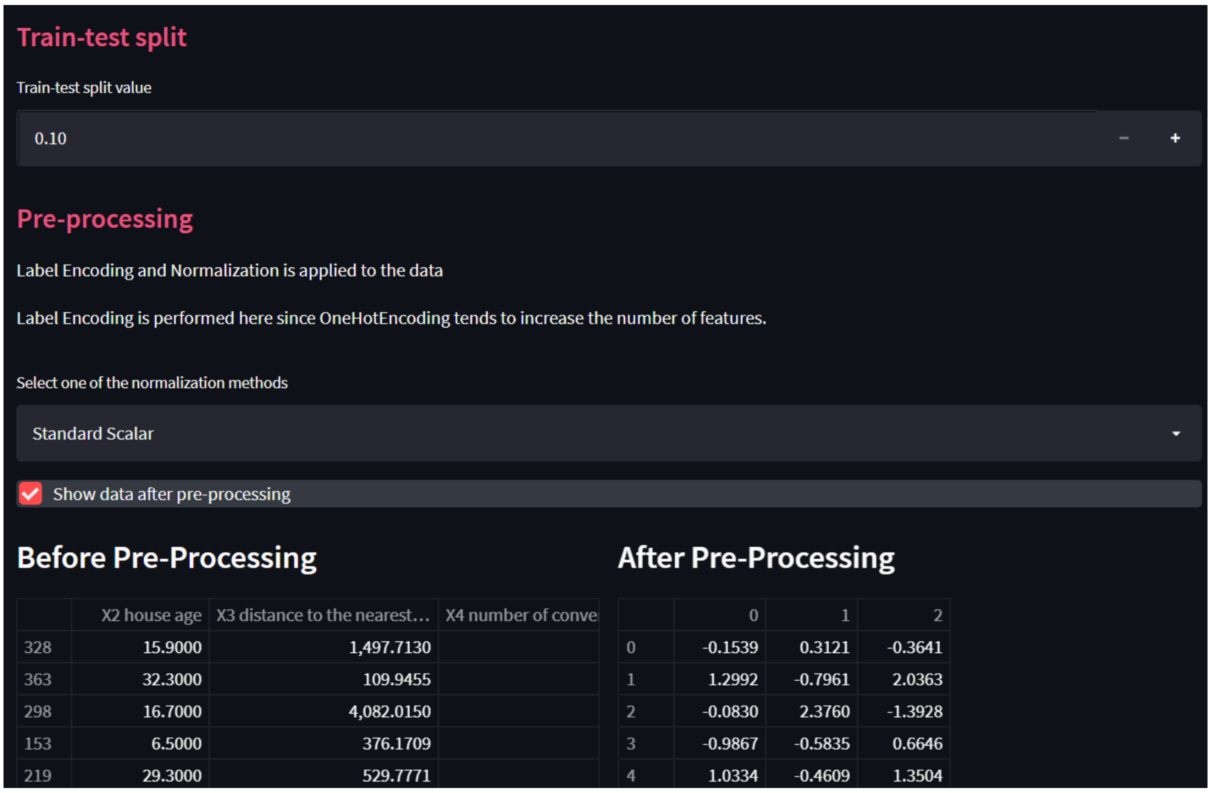
Feature Selection

Select all the columns you want to predictor variable

✓ Select all columns

Also, in this case instead of performing feature selection to select the best features, we have ticked the check box asking the model to use all the available features to build the model.

The next step of train-test split and pre-processing is exactly the same as the classification task.



Train-test split

Train-test split value: 0.10

Pre-processing

Label Encoding and Normalization is applied to the data

Label Encoding is performed here since OneHotEncoding tends to increase the number of features.

Select one of the normalization methods: Standard Scalar

✓ Show data after pre-processing

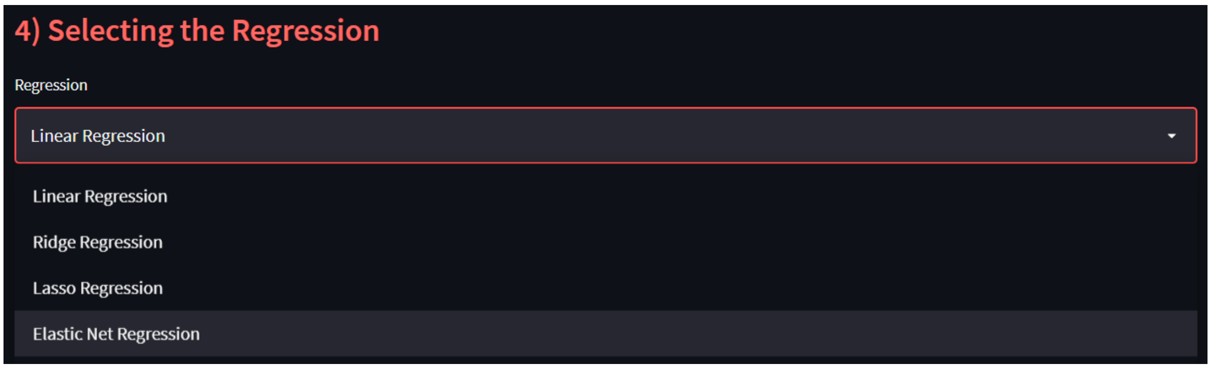
Before Pre-Processing

	X2 house age	X3 distance to the nearest…	X4 number of conve
328	15.9000	1,497.7130	
363	32.3000	109.9455	
298	16.7000	4,082.0150	
153	6.5000	376.1709	
219	29.3000	529.7771	

After Pre-Processing

	0	1	2
0	-0.1539	0.3121	-0.3641
1	1.2992	-0.7961	2.0363
2	-0.0830	2.3760	-1.3928
3	-0.9867	-0.5835	0.6646
4	1.0334	-0.4609	1.3504

The regression model selection options are provided to the user, with 4 regression model choices – Linear regression, Ridge regression, Lasso regression, Elastic Net regression.



4) Selecting the Regression

Regression

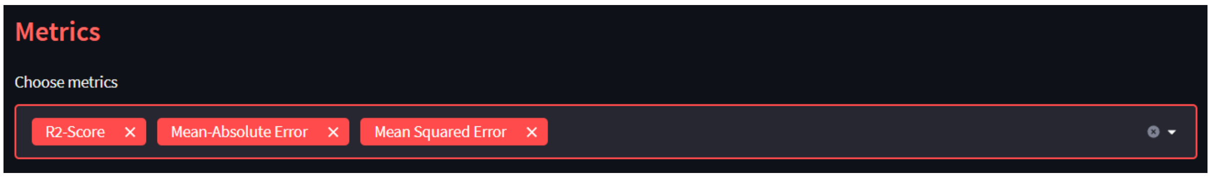
Linear Regression

Once the user selects one of the regression models to proceed with, the user is presented with all their respective model's hyperparameters for the user to tune with to see how the model performance changes.

Compared to the classification models, the regression model only has a very few numbers of important hyperparameters. The simplest model – the linear regression model doesn't have any hyperparameters. Also, the other models have only a 2-4 hyperparameters and they are mentioned below.

Ridge Regression	**Lasso Regression**	**Elastic Net Regression**
Alpha	Max Iterations	Alpha
Max Iterations	Selection	L1 Ratio
Solver		Max Iterations
		Selection

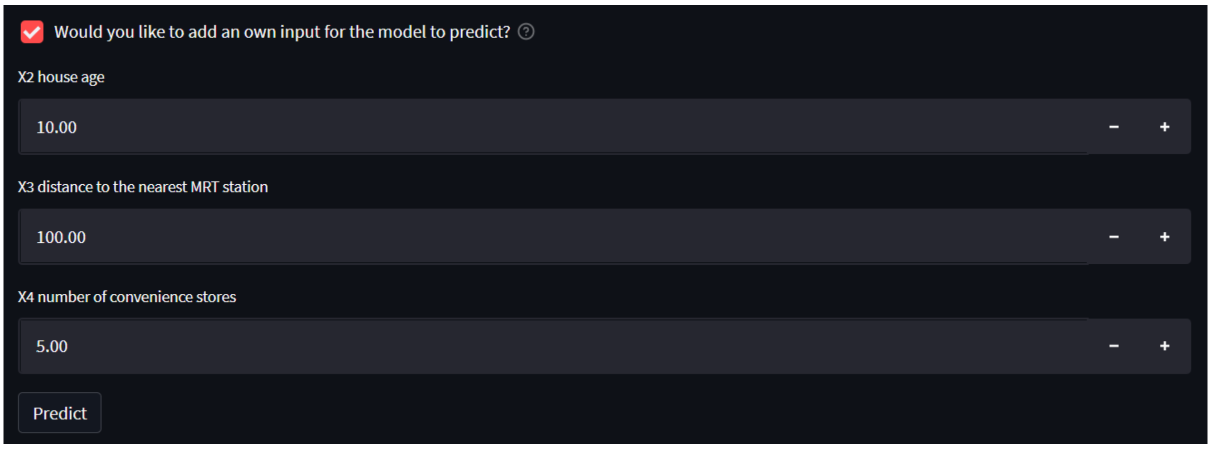
Next the user gets to choose the metrics to evaluate the model with, where the user has the choice to select one or more of the given metrics which include – R2 Score, Mean-Absolute Error, Mean-Squared Error. These metrics are selected as they are some of the most popularly used regression metrics.



Metrics

Choose metrics: R2-Score, Mean-Absolute Error, Mean Squared Error

Similar to the classification task, the user if has any test case scenarios in mind, he can send the inputs before the training itself just by clicking the check box to add inputs for test cases. The user, next has to enter the values for each feature used to build the model.


Would you like to add an own input for the model to predict?

X2 house age

10.00

X3 distance to the nearest MRT station

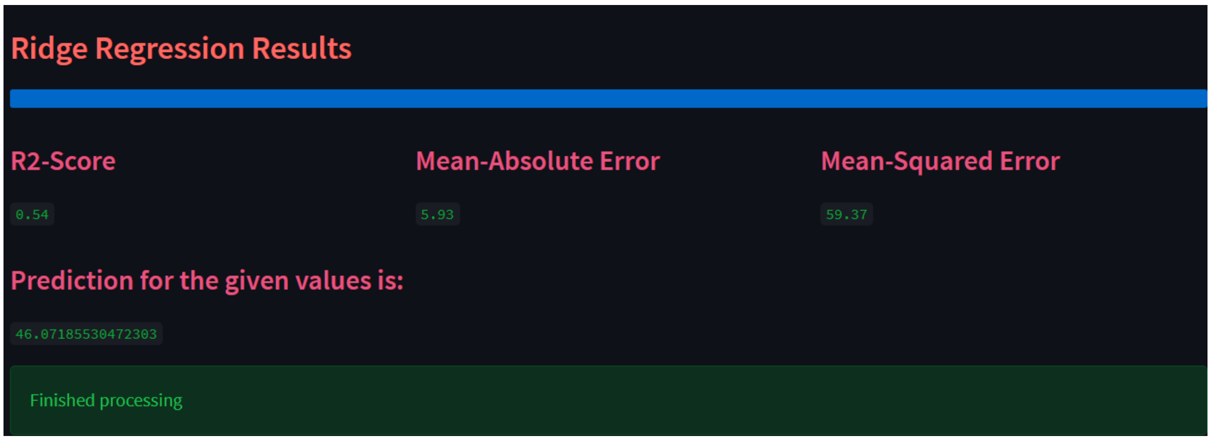
100.00

X4 number of convenience stores

5.00

If the user doesn't have any test case on mind, the above check box can be avoided. Next, the user can click on the "Predict" button and the model is built as per the given user specifications.

The metrics selected by the user is next tested and the results are returned back to the user, using which the performance of the model can be measured and compared.


Ridge Regression Results

R2-Score 0.54

Mean-Absolute Error 5.93

Mean-Squared Error 59.37

Prediction for the given values is:

46.07185530472303

Finished processing

6. Conclusion

I have built a AutoML app that is fully automated and any user can build a classification/regression model in under 2-3 mins without using even a single line of code.

The AutoML app potentially includes every stage from beginning with a raw dataset to building a machine learning model ready for deployment. It also has the ability to help with learning of machine learning model for beginners and also able to deploy small real world and industrial projects.

The added benefit of being a web-based application, means that it can be easily deployed and accessed by anyone.

6.1 Future Works that can be added

As mentioned before the application was built with scalability in mind for future updates with ease. The classification/regression models available at the moment can be increased in the future by adding support to more models and also more pre-processing steps and hyperparameters.

Also, if the project could be integrated with a cloud service subscription. It would help increase the app's ability to build more higher scale projects which require more storage facilities and computational power.

7. References

1. Inspiration for selecting AutoML: https://docs.microsoft.com/en-us/azure/machine-learning/concept-automated-ml

2. Inspiration for UI designs: https://github.com/jrieke/best-of-streamlit

3. Streamlit Documentation: https://docs.streamlit.io/

4. Sklearn Documentation: https://scikit-learn.org/stable/