EECS605 Project

Ford Focus Price Prediction

Qi Yang

Apr 8th, 2022

1 Indroduction

In this project, we train a linear regression model and deploy it onto AWS cloud. The machine learning algorithm is used to predict price for a Ford Focus car based on provided information about the car. We also create a heroku website and establish a pipeline to connect the website to AWS cloud. Users can upload their own informaiton file of the car in the website and get the predicted price. They can also choose to demo with input files we provided in the dropdown menu to check whether our application can really work.

2 ML Algorithm Training

2.1 dataset

The dataset used in this project is 100000 UK Used Car Data Sets from Kaggle. The specific URL is https://www.kaggle.com/datasets/adityadesai13/used-car-dataset-ford-and-mercedes
This project only uses Ford Focus dataset, which has 6 features and 5454 samples. Features includes: Year(Registration year), Price(price in British Pounds), Mileage(Distace used in miles), Transmission(type of gearbox:Manual or Automatic), Fuel type(Engine fuel type:petrol or diesel), Engine size(total volume of the cylinders in the engine in litres). The project is aimed to use 5 features(year, , mileage, transmission,engine size) to predict a new price.

2.2 Model Training

2.2.1 prepocessing and checking missing data

Firstly, we need to prepocess the dataset. We convert the string features(transmission and fuel type) into numeric features. For the dataset we use, we only have limited number of different values for each string features. For example, for the feature Transmission, there are only two different types. So we can simply convert them into number 1 or number 2 manually. But for other dataset with a large number of different string feature values, we can use LabelEncoder from scikit-learn to convert the string features into numeric feartures.

Secondly, we need to check the missing data in the dataset. In the dataset we use, we have 0.02 percent of missing data. For simplicity, we replace these missing data with average values. Lastly, we randomly divide the dataset into a training set and a test set in a ratio of 7:3.

2.2.2 model training

This project uses the linear regression model to train the dataset. And we use a lot of related functions of linear regression in the scikit-learn library to train our model more conveniently and accurately. Reference:https://scikit-learn.org/stable/modules/generated/sklearn/linear-model/LinearRegression/html?highlight=linear20regressionsklearn/linear-model
Firstly, we want to figure out the potential relationship between the price and other variables, so we check the correlation value of each variables to identify whether the price is denpendent on these variables.

	year	price	transmission	mileage	fuelType	engineSize
year	1.000000	0.761746	0.061151	-0.808609	-0.096480	-0.184722
price	0.761746	1.000000	0.094377	-0.741632	-0.096347	0.255457
transmission	0.061151	0.094377	1.000000	-0.107125	0.031997	0.043861
mileage	-0.808609	-0.741632	-0.107125	1.000000	0.232573	0.164202
fuelType	-0.096480	-0.096347	0.031997	0.232573	1.000000	0.490878
engineSize	-0.184722	0.255457	0.043861	0.164202	0.490878	1.000000

Figure 1: correlation value table between features

Then we compute the five coefficients for the linear regression objective function.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 \tag{1}$$

We get the Intercepts β_0=-2611418.185, and the rest coefficients β_1=1298.31, β_2=133.9250, β_3=-0.064630, β_4=-2277.508, β_5=6157.62
And compute the R^2 value for both train and test data. We get R^2_{train}=0.8388023897662493 and R^2_{test}=0.8098576827133581, which are acceptable.

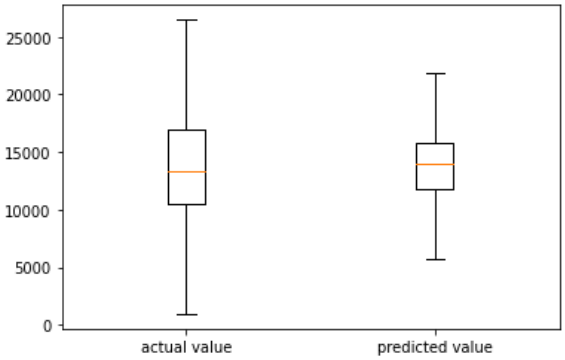

Figure 2: boxplot

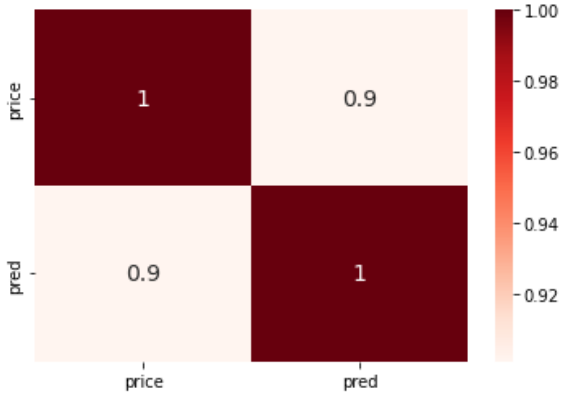

Figure 3: heatmap

The heatmap, histogram and boxplot show the visualizations of predicted values and actual values. The heatmap indicates that the difference between actual values and predicted values is close. The correlation value shows that the similarity of the prediced values and actual values is around 0.9. Meanwhile, the histrogram and boxplot display the range of predicted values and actual values. The histogram reveals that the actual values are more dispersed, while the predicated value are more concentrated. The similar situation can also be found in the boxplot. Simply speaking, the predicted values are mostly covered by the actual values. The limitation of predicted values avoids the existences of the outliers and increases the accuracy of the model. And the mean squared error is 140655.29137024521, which is also acceptable based on the magnitude of the price.

Lastly, package the trained model into a .onnx file.

3 Example Work

3.1 successful cases

We set the basic input information as: Year 2015,Transmission Manual,mileage 50000miles,fuel type petrol,engine size 2litres, and get our origin output result as 13768.25pounds. Then we change one of parameters each time to test the result.

year 2010	7276.5	year 2019	18961.5	mileage 20000	15707.25
year 2011	8575	year 2020	20259.75	mileage30000	15060.75
year 2012	9873.25	automatic	15749.5	mileage 40000	14414.5
year 2013	11171.5	diesel	13338	mileage 50000	13768.25
year 2014	12469.75	enginesize1	7610.5	mileage 60000	13122
year 2015	13768.25	engine size1.5	10689.5	mileage 70000	12475.5
year 2016	15066.5	engine size1.6	11305.25	mileage 80000	11829.25
year 2017	16364.75	engine size2.3	15615.5	mileage 90000	11183
year 2018	17663	mileage 10000	16353.5	mileage 100000	10536.75

From the table above, it can be seen that the results are consistent with real life experience, which proves that our model can work properly.

3.1.1 bad cases

We now know two types of bad cases. The first is that the input numeric features are numbers that have no real meaning. The output can be very outrageous at this point. This is because we do not set a input range for those numeric features. The second is to enter inconsistent values in the string feature. At this point the model will report an error.
For example, if we enter the year200, we will get a negative result as -2342671.8, which is absolutely impossible in real life. If we enter the fuel type as a number, we can see "invalid literal for int() with base 10: 'Wrong fuel type' ".

4 Cloud Architecture

We deploy all our back-end programs on AWS, and our front-end applications are deployed on the heroku. Then we use the AWS APIs to create a pipeline to connect them. We create

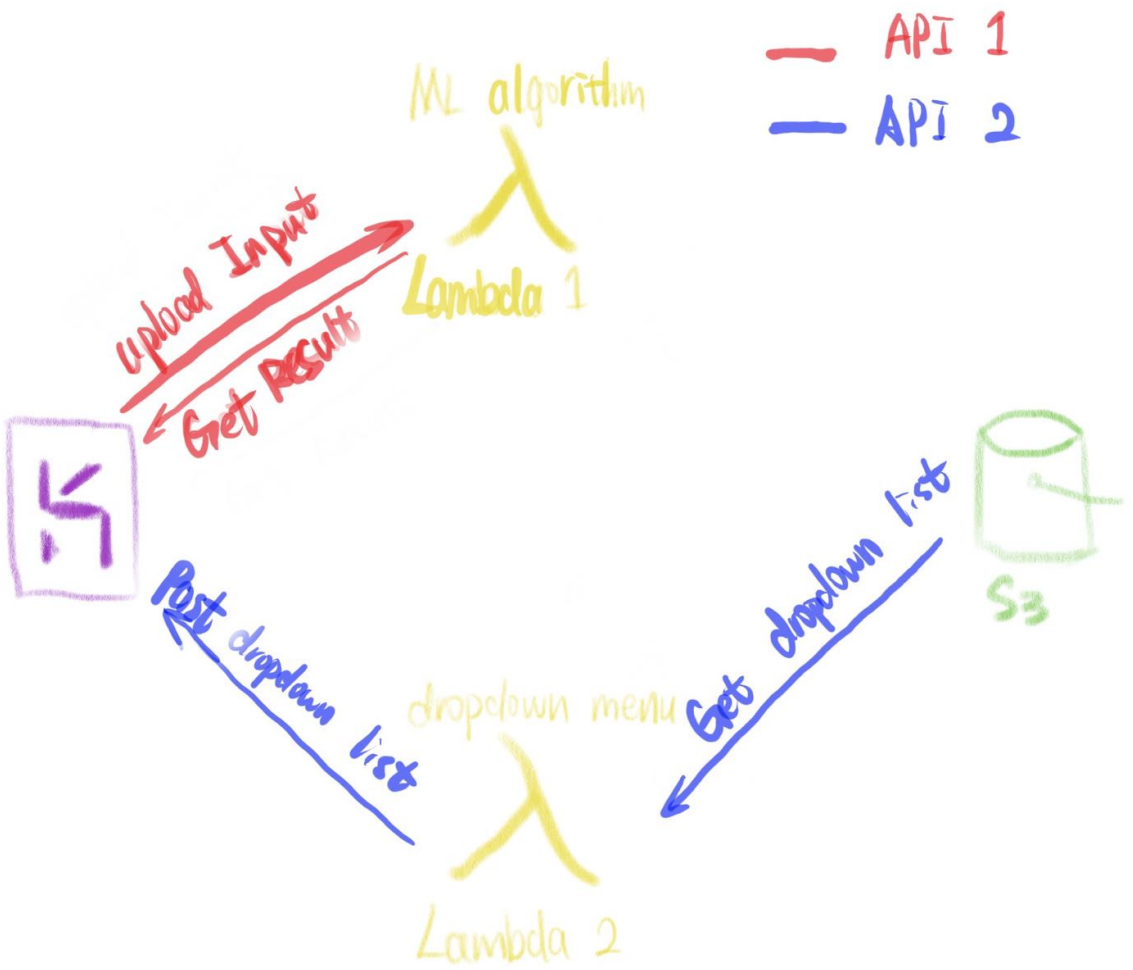

Figure 4: Cloud Architecture

two Lambda functions in AWS. One is uesd to deploy our trained ML learning algorithm, the other one is used to create the drop down menu for users. And we have two S3 buckets.One stores the trained model file. The other one stores all the demo examples so that users do not need to input anything but still can demo our work. Besides that we have two APIs. One has only POST method used to connect heroku website and our ML algorithm Lambda function, so users can upload files to get result. The other one has POST and GET method so that our ML algorithm Lambda can get the demo input files from S3 bucket and put the results to the heroku website.