CSCI-4930/5930: Machine Learning

Final Project Report

1. Title **of the project**: Car Acceptancy Prediction Tool
2. Team:

I. Hitesh Ram Kotha – [hiteshram.kotha@ucdenver.edu](mailto:hiteshram.kotha@ucdenver.edu)

II. Raghu Veera Reddy Malreddy – [raghuveerareddy.malreddy@ucdenver.edu](mailto:raghuveerareddy.malreddy@ucdenver.edu)

III. Vyshnavi Sankenani - [vyshnavi.sankenani@ucdenver.edu](mailto:vyshnavi.sankenani@ucdenver.edu)

1. Introduction
   1. Definition of the task(s)/problem:
2. Create a Portal to take inputs from the user.
3. Vehicle Price Depreciation calculation.
4. Implement Classification algorithms and store the metrics.
5. Implement a price negotiation feature.
6. Implement License plate recognition.
7. Implement Facial recognition.
   1. Purpose of the task(s):
8. Customers willing to sell their cars visits the portal and inputs their Car’s Make, Model, Year Purchased, Expected Price, Maintenance Cost, Number of doors, Number of Persons can be seated, Luggage Boot Space and Safety Level.
9. In the dataset the price column is categorical where the values are ‘Very High’, ‘High’, ‘Medium’ and ‘Low’ which are relative terms and the user cannot judge that the price he quoted is High, Medium or Low. Hence, we calculate the actual deprecated price of the car based on make and model and compare it with user’s quoted price and determine if it’s Very High, High, Medium or Low. In Detail explanation is given in the file below
10. We will implement the classic classification algorithms to predict if we accept the car or not and visualize the results.
11. We wanted to extend our proposed project by implementing a price negotiating feature where in spite the customer’s quote for the car is acceptable for us, we will reduce the price in such a way that it’s acceptable to customer and in a way, we can maximize the profit.
12. We wanted to implement a license plate recognition mechanism to check if the car has been listed in any of the missing car list.
13. We will implement a facial recognition feature so as to check if the customer needs any back-ground verification by taking an image of customer and verifying with police records.
14. Information on the Dataset:

We have 6 Attributes namely

* Buying price – The amount bid by the user to sell his car. It is a categorical column which has only the values Very High, High, Medium and Low.
* Maintenance – The amount required for maintenance; it takes the values similar to buying price.
* Doors – Number of doors available 2,3,4,5-more.
* Persons – Number of person’s can be seated in the car 2, 4, more
* Luggage Boot Space – Available space for boot. Small, Medium and Large.
* Safety – Level of safety available in the car low, medium and high,

1. Proposed method**:**

We will implement a web application to take the user inputs in python. The depreciated value of a car is already completed using the description in the document. We are looking forward to implement the following classification algorithms and use the one which has good evaluation metrics.

* K – Nearest Neighbor
* Naïve Bayes Algorithm
* Support Vector Machines
* Decision Trees and
* Random Forest

In order to extend our proposal, we thought it would be ideal to implement the feature of negotiation using reinforcement learning concept where even though user quotes an acceptable price, algorithm prompts a reduced price and gets rewarded if the user accepts it and gets penalized if the user does not agree. It made things complicated as we had to make many iterations from system to customer. Hence, we formulated another strategy i.e. after inputting the details of the car and if the classifier predicts that it is only accepted or good and the reason is due to high selling amount proposed by the customer. We lowered the price step by step and ran classifier till we get the deal prediction is very good. We have programmed conditions in such a way that the proposed price is always greater than depreciation value so that customers can benefit (if they have already spent money on maintenance and improving safety conditions) instead of getting hard depreciated value, and the car dealer can earn some profit instead of agreeing to the original price.

We took this opportunity to implement number plate recognition mechanism so as to check if the car has been listed in missing cars list and if found it can be reported to authorities. We have implemented using Tesseract-OCR

We have implemented facial recognition to check if there any police records against the customer, so as to do a background verification.

*\*Original transactions may not happen in this way but we just made up use-cases to incorporate the features.*

1. Summary of contributions: List all members of the team, and who contributed in what part.

|  |  |  |
| --- | --- | --- |
| **Member name** | **Task** | **Comment** |
| Raghu Veera Reddy | Data Translation, Classification | Conversion of price into respective categorical value and implement classification and identify best classifier |
| Vyshnavi Sankenani | Web application implementation, License plate recognition | To create a web application which accepts user inputs and implementation of license plate recognition to check if it exists in lost cars list. |
| Hitesh Ram Kotha | Negotiation Feature Implementation, Facial Recognition | Negotiation feature is implemented using the classifier |

1. Implementation:

**Data Translation**

**Introduction:** The original data set that we acquired from Kaggle consists of categorical values. Let’s consider a column buying which consist of the values “vhigh”, “vlow”, “med”, “high”, “low” in it. So, while considering these categorical values and do task on the data. But we would like to the user to enter details of the car through the dropdown list provided in the frontend. So, the basic ideas are to have meaningful relation with the user entered input values and the data set. As of now there is relationship between user and proposed system that we are going to derive with the dataset.

**New Data Set:** First thing we need when establishing the relationship with the categorical values of original data set and with user is that we need to have a new dataset with required information not in categorical.

The car.csv dataset consists of 479 different Models and 35 different makes with each models MSRP.

* Make is the name of the company that manufactured.
* Model is the manufactured company car name that is unique.
* MSRP is Manufacturer’s Suggested Retail Price.

And the dataset has some other columns which is also important and may be used later. With required data gathered the next step is to establish the relationship.

**How to calculate rate of depreciation and what is depreciation?**

Depreciation is the reduced value of an asset with the passage of time in particular to wear and tear. The rate of depreciation tells about the car or an asset value after some years and how much its values is reduced when the asset was actually brought. For example, let us consider a car. So, the car was brought in 2018. After the purchase itself the car price is reduced 20% of its original value. According to standard depreciation rate system that generally used in market.

For calculating the value of a car after n years, we are using a formula that will calculate the value after some research over present market trend and some general scenarios.

A = P \* (1 – R / 100) n

A = value of car after n years

P = Original Value of car

R= Rate of depreciation per annum,

n= No of year the car is used.

So, with the above formula the value of the car after using n years can be calculated.

**What does the code do?**

For now, the primary result is to obtain the categorical value of the entered value. For now, we made it as command line input values. Make, Model, MSRP and year of car originally brought.

With the input values the target MSRP is found from the dataset and used for calculation of amount of car after n years(A).

The rate of depreciation is considered as 20% for the first year and 10% for ever year. Once the value ‘A’ is obtained. It is used to compare with the user’s value.

Here the value is compared with the user entered amount he wants to sell. The output will be the categorical value of the percentage increase or decrease of the original value(A). And the output will be [veryhigh, high, med, low, verylow], which is categorical.

**Output:**

User is asked to enter the input details (Made on frontend now). Entered as user through terminal. Input will have:

A screenshot of a social media post

Description automatically generated

Make, Model, purchase year and car value customer want to sell is given by customer And ***High*** is the output after calculating the depreciation amount.

Another example can be:

A screenshot of a cell phone

Description automatically generated

From the above images we can see that the output is categorical, and we do not know what the original amount of the car is. The below images will show that:

A picture containing drawing

Description automatically generated

Here from the above we have the input data entered by user and the output depreciation can be seen as:

A picture containing ball, hitting, black, player

Description automatically generated

From the above the actual price after calculating depreciation amount is 29556.0 but the user wishes to sell this at 30000. So, the output from the above will be “***med***” can be seen from the below image.



Conclusion: Finally, there is some meaning full relation that is established between with the users input and dataset. It is obtained with the above method.

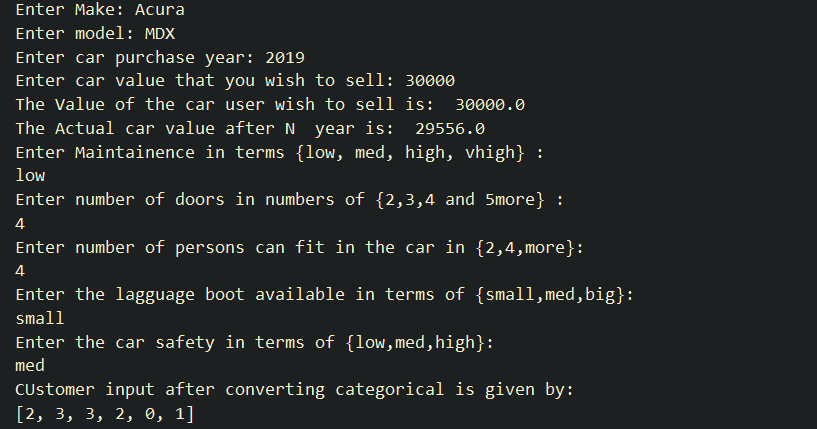
**Converting categorical data to Integer data**

From the above process the we are able to find the depreciation amount and a categorical value which tell whether proposed value by customer is high or vhigh, med, low or accepted. Now that we have buying value from the above. We ask customer to enter the furthermore specifications of car. Like number of door, number of people can fit in the car and safety and maintenance of car. And luggage boot space in the car. So, with the data entered by customer, we will use this information and predict whether the car is acceptable or not. One of the example of how customer entered looks can be seen from the below:

A picture containing black, display, apple, red

Description automatically generated

So from that we have the value from customer input and after converting them it will look like this:



In the bottom of the image above we can see the integer data. Here it is done manually. First, we used the ***pandas factorize*** on original data set and we have come to know the unique values of each columns and their corresponding integer data, and we used that information and converted customer input manually because the data here entered is of one row so it won’t be feasible for one row if we use any encoding schemes. So first we encoded on the whole dataset and found the value of each value assigned those values to the respective inputs given by customer. From the above we have converted the data into integer. The next step will be performing the classification on original Data set and finding which classification algorithm performs better or predicts more accurately.

**Classifier Identification**

We have considered 5 classifier for our prediction. Trained our models and then found accuracy of each classifier. The 5 classifiers are KNN, Naïve Bayern’s, SVM, Decision Tree and Random Forest. Here we have trained each model on our data set and then found accuracy of each classifier with respect to our dataset and plotted the accuracy as below:

A close up of a logo

Description automatically generated

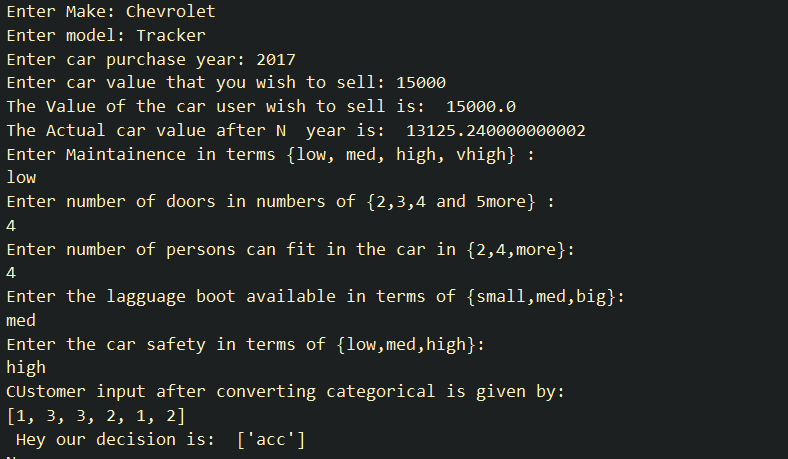
A screenshot of a cell phone

Description automatically generated

From the above we can see that decision tree had performed better and random forest too. After evaluating series of times, we found decision tree is consistent and predicts better with this data. So once after identifying the classifier, we again trained with our dataset and predicted the target.

**Predictions:**

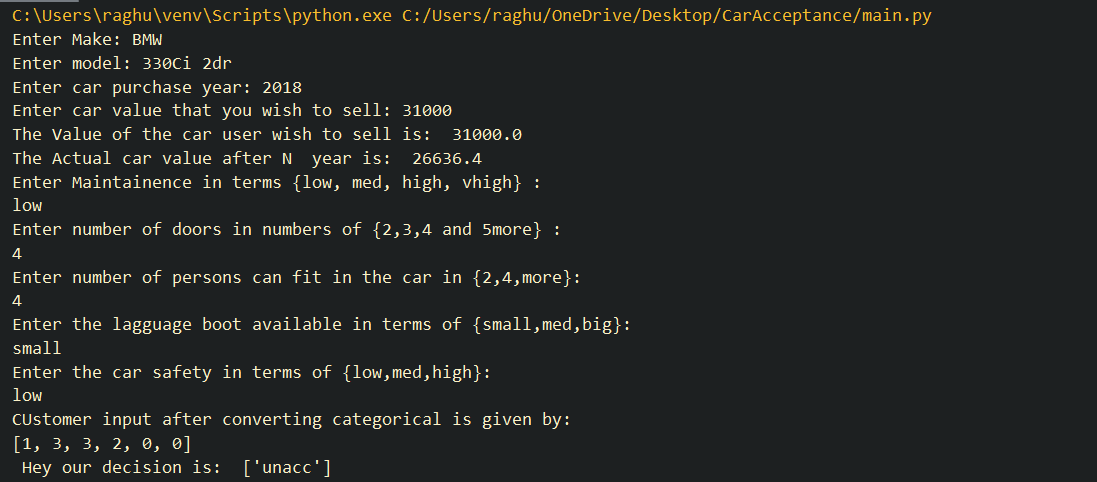
So once after finding the classifier, we trained the model and tested multiple times whether it is predicting right for the test data and then we gave worked on lot of inputs as customer and then evaluated the results based on the specifications provided by user. Decision tree is trained on data and tested. Once the target has been predicted it will be of Accepted, Unaccepted, vgood, good. Apart from accepted the other target values goes into further analysis for negotiation. Inputs sent to prediction are:



Here the values [1,3,3,2,1,2] are sent to:



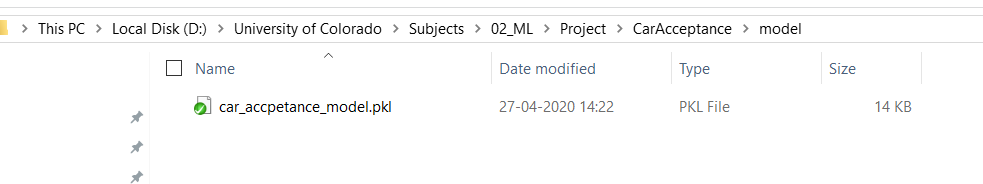
And above accepted is target predicted. We will see unhappy cases now:



The above is the unaccepted case. This will be going further negotiation process.

**Price Negotiation:**

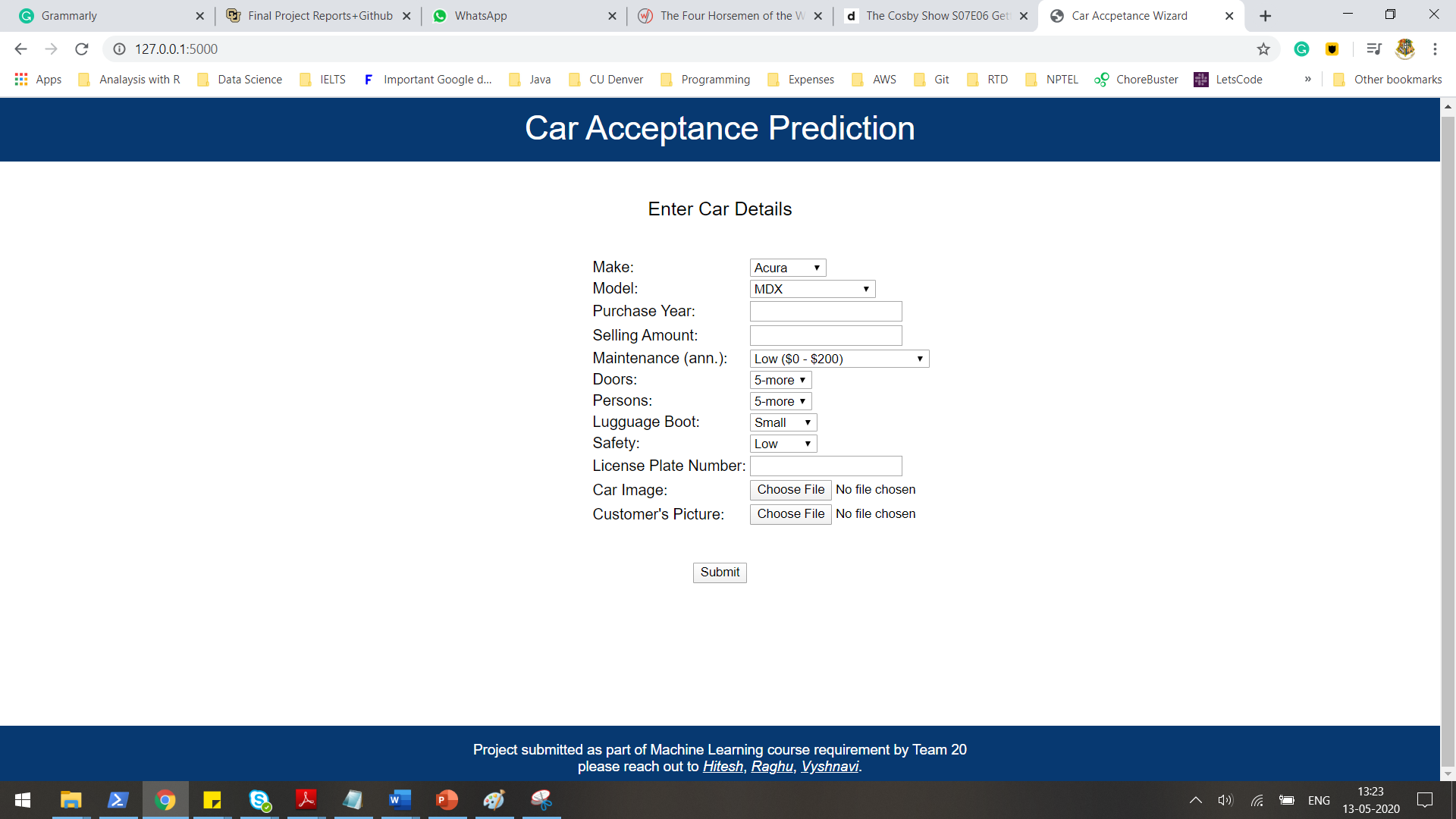
Storing the Classifier - After selecting appropriate classifier, we have stored it using pickle library as below

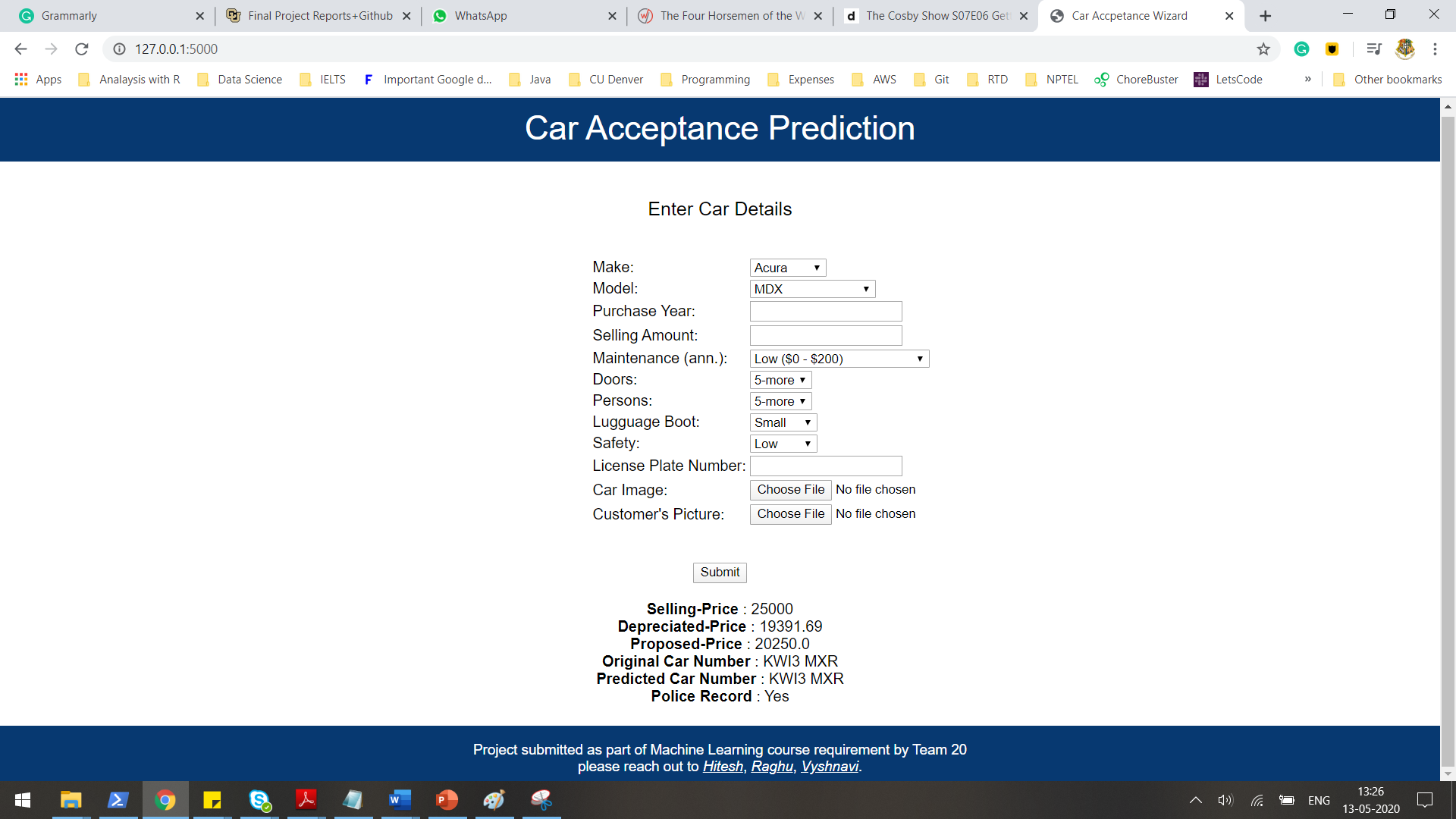


Using this classifier, we will try to update our buying price. We will initiate negotiate only when we get the initial prediction of the deal as ‘Accepted’ or ‘Good’ and the price quoted by the customer is either ‘High’ or ‘Very High’. We will reduce the customer’s quote by ten percent and run the classifier with the same rest of the parameters and check if the deal gets to ‘Very Good’ and simultaneously we will check if its greater than depreciated value. If we got beneath depreciated value and the deal is still being predicted as accepted or good, that’s because other parameters like high maintenance or low safety has factored in and we will just quote the price above depreciation price for buy out.

**Web Application**

We have created a web application to take inputs from the user and we are running the python code using flask in background. Webapp takes the information and sends to a python file and renders the output to the user.





**License Plate Recognition**

The following steps were taken in implementation of license plate recognition. And the following libraries were considered for implementation along with OpenCV for image manipulation.

* OpenALPR
* Tesseract-OCR

Step 1 – Read the original image from the webapp and resize the image to 500x500 pixels



Step 2 – Convert the image to grayscale



Step 3 – A Bilateral filter is applied which is a non-linear, edge-preserving, and noise-reducing smoothing filter for images. It replaces the intensity of each pixel with a weighted average of intensity values from nearby pixels.



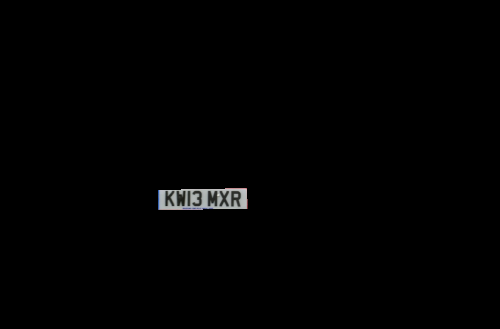
Step 4 – We Identify the canny edges by using multi-stage algorithm to detect a wide range of edges in images.

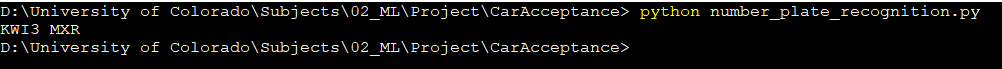


Step 5 - Find the contours in from the edges detected and sort the top 30 contours.



Step 6 - Get the perimeter of each contour and select those with 4 corners. Mask all other parts of the image and show the final image. Read the text using Tesseract OCR.

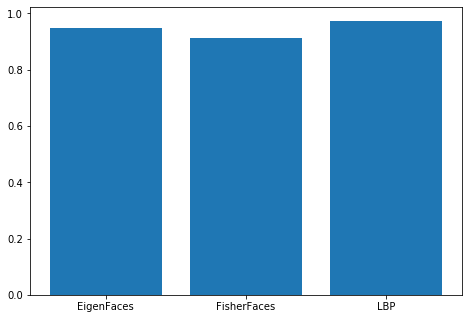




We store the prediction and compare with the input user has given and we will calculate the similarity and store the results to test the classifier.

Facial Recognition

We have looked into three kinds of algorithms Eigen Faces, Fisher Faces and Local Binary Patterns. We have taken subjects from yale face database and trained a recognizer. We take the input image and run against predict with classifier and check for the distance. If the distance is greater than the threshold, we do not need any background verification else we will subject the customer to background verification. For each algorithm we calculated accuracy and finalized that LBP has highest accuracy. Usually, Fisher face algorithm will have higher accuracy compared to Eigen faces as we do not have various illuminations or different expressions in the dataset. The classifier needs to be retrained for updated police records. We have plotted the accuracies as below.



1. Future Work

* Check possibilities of Reinforcement Learning in price negotiation as part of learning.
* To do Fine grain analysis on the number plate to identify Genuineness of the plate.
* To implement age factor in facial recognition.

1. List of references:

[1] <https://medium.com/@agustindev/face-recognition-with-opencv-280ec1213ffd>

[2] <https://www.superdatascience.com/blogs/opencv-face-recognition>

[3] <https://docs.opencv.org/3.4/da/d60/tutorial_face_main.html>

[4] <https://www.hindawi.com/journals/acisc/2020/8535861/>