May 9, 2021

Final Phase

Steven Guo
cs 4300
University of Missouri-St. Louis
Click Here For Google Colab Source

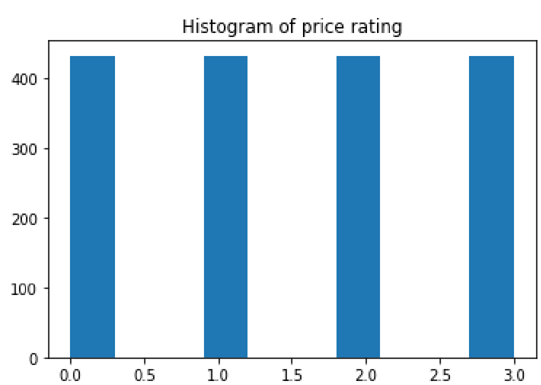
Abstract

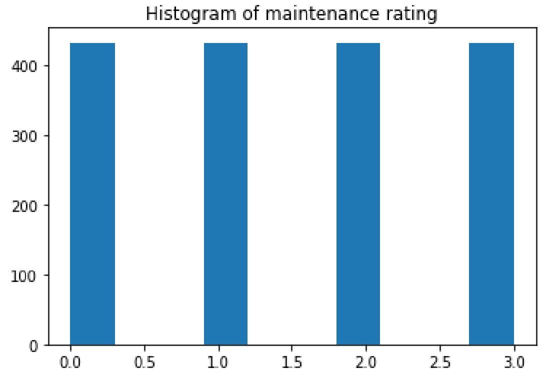
I chose to work on this project because it would be helpful in deciding on what car to buy in the future.

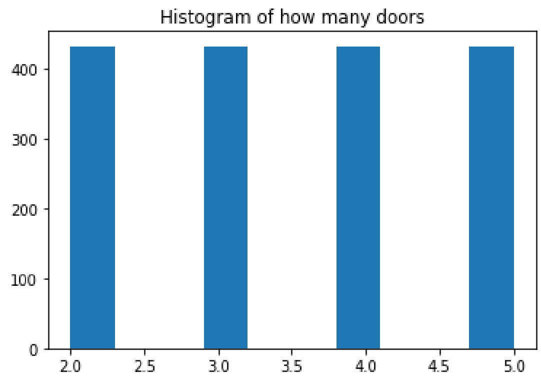
Car Evaluation Database was derived from a simple hierarchical decision model originally developed for the demonstration of DEX, M. Bohanec, V. Rajkovic: Expert system for decision making. Sistemica 1(1), pp. 145-157, 1990.). The model evaluates cars according to the following concept structure: CAR car acceptability

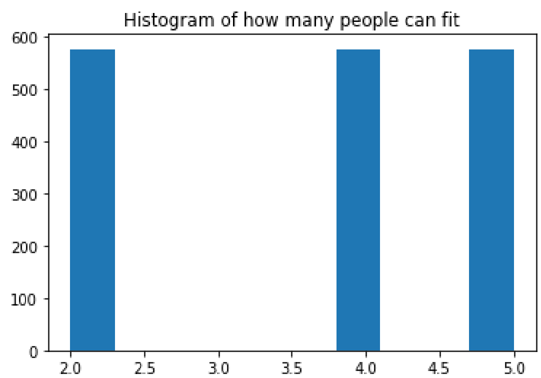
. PRICE overall price
. . buying buying price
. . main price of the maintenance
. TECH technical characteristics
. . COMFORT comfort
. . . doors number of doors
. . . persons capacity in terms of persons to carry
. . . lug boot the size of luggage boot
. . safety estimated safety of the car

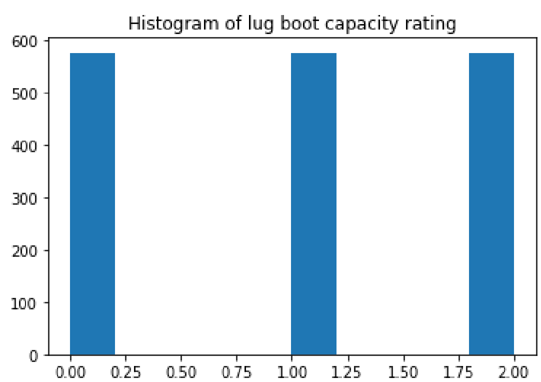
1 Histogram

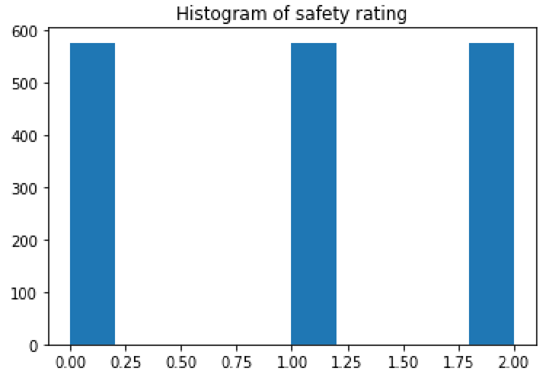


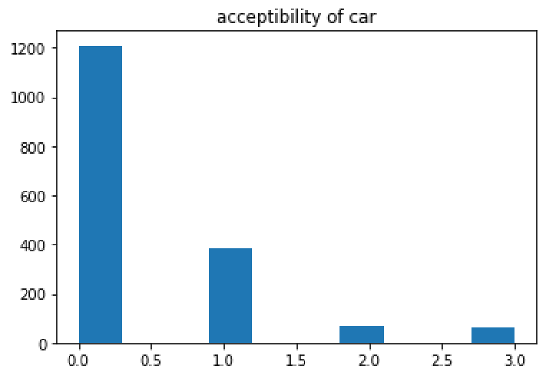

Histogram of maintenance rating


Histogram of how many doors


Histogram of how many people can fit


Histogram of lug boot capacity rating

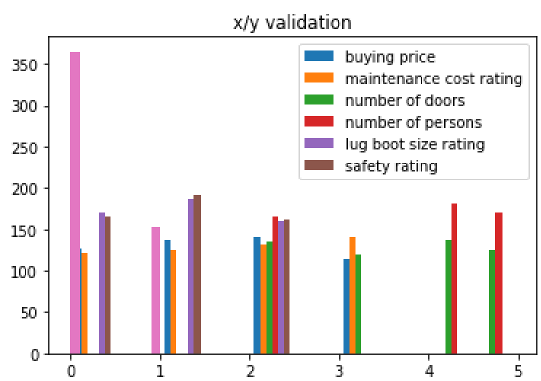




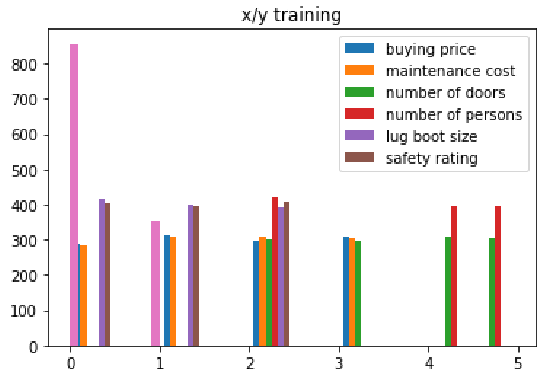
2 Dataset manipulation

I normalized the last column which determines if the car is unacc, acc, good, or vgood. 0,1,2,3 respectively. I made all values not zero equal 1 for logistic regression. Which 0 determines the car to be unacceptable and 1 to be acceptable. Then I find 30 percent of the data and split it into training and validation, 30 percent validation set and 70 percent training.

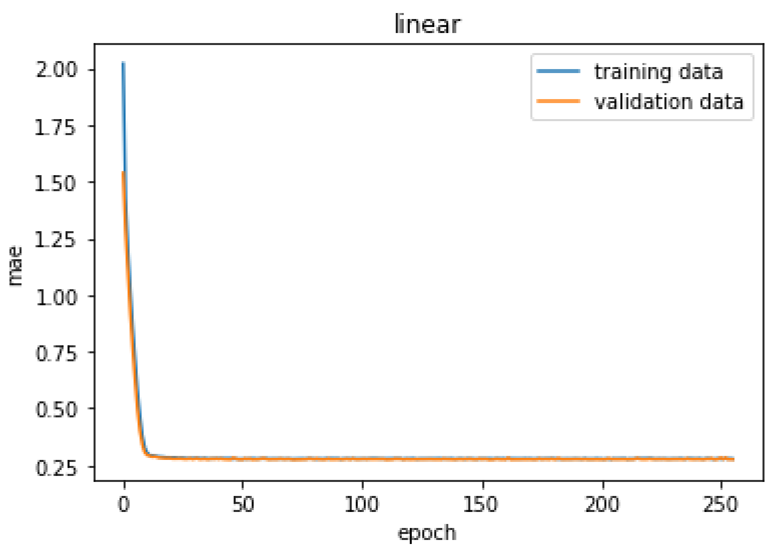
3 Validation Dataset



4 Training Dataset



5 accuracy on the validation set




```
Training
Accuracy: 10.50%
Precision: 56.83%
Recall: 10.50%
F1-score: 0.18
Validation
Accuracy: 9.07%
Precision: 50.33%
Recall: 9.07%
F1-score: 0.15
```

To get this I made a neural network with 6 neurons with linear activation in the first layer and 1 neuron with linear activation in the last layer. The accuracy on the validation set is 10.50 percent

6 Neural network experimentation

The first layer will have 6 neurons to match the colunms I have. When I add more layers I double the neurons every additional layer. I also use relu activation with the last activation being sigmoid. I tried it with 4 initial neurons and doubling it every additional layer, but it did not like it. I was getting zeros for precision, recall, and f1-scores

model	accuracy	percision	recall	fl-score
1 layer	26.78	18.64	41.85	0.26
2 layer	60	43.08	98.1	0.6
3 layer	58.68	25.19	18.21	0.21
4 layer	77.44	77.46	36.41	0.5
5 layer	78.51	76.21	42.66	0.55
6 layer	74.21	54.73	88.04	0.67
7 layer	75.7	73.72	31.25	0.44
8 layer	70	77.78	1.9	0.04

7 modifying activation functions

Here I ran some experiments involving activations between linear and sigmoid. This is the result. All experiments at this point is run with values from the base case. The only thing that will be changed is the activation

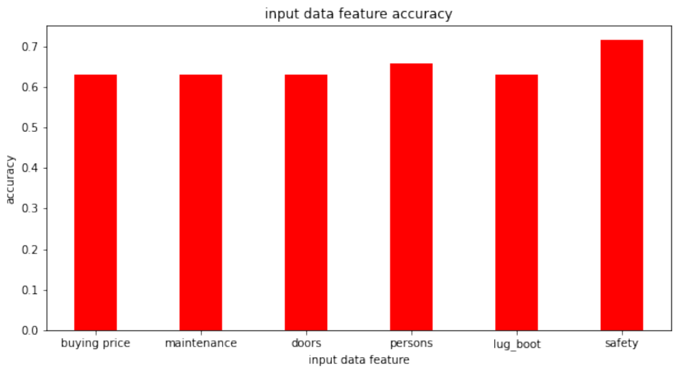
model	accuracy	precision	recall	f1-score
base	26.78	18.64	41.85	0.26
linear(last)	70.41	49.58	70.41	0.58
linear(all)	65.45	83.83	65.45	0.66
sigmoid(last)	41.57	33.62	100	0.5
sigmoid(all)	69.59	0	0	0

8 evaluation

model	accuracy	precision	recall	f1-score
base	26.78	18.64	41.85	0.26
linear(last)	70.41	49.58	70.41	0.58
linear(all)	65.45	83.83	65.45	0.66
sigmoid(last)	41.57	33.62	100	0.5
sigmoid(all)	69.59	0	0	0
overfitting	90.74	79.57	92.46	0.86

Essentially everything performed horribly except for overfitting. Not sure what is going on with all layers having sigmoid activation. I assume there were some errors involving that. The model that had all linear layers did okay. The model that had sigmoid as its last layer had a 100 in recall so the positives were identified correctly.

9 Feature Importance



This is accuracy of all features of the dataset.
buying price rating-63.14
maintenance rating-63.14
number of doors-63.14
number of people-65.70
lug boot rating-63.14
safety rating-71.57

From what I can tell, buying price, maintenance, number of doors, and lug boot rating are all the same accuracy. They also have the lowest accuracy and I will be removing these features top to bottom. Then I will remove features with higher accuracy to try to study feature importance. The accuracy on these input features are very close together so there may not be any features to remove. The process of removing input features is this:

buying price, maintenance-number of doors, lug boot rating, number of people, safety rating

10 Removing Input Features

Base Case

```
Accuracy: 87.52%
Precision: 83.45%
Recall: 82.51%
F1-score: 0.83
```

First Input removed-buying price rating

```
Accuracy: 86.69%
Precision: 82.91%
Recall: 80.49%
F1-score: 0.82
```

Second Input removed-maintenance rating

```
Accuracy: 86.45%
Precision: 82.34%
Recall: 80.49%
F1-score: 0.81
```

Third Input removed-number of doors

```
Accuracy: 88.10%
Precision: 84.63%
Recall: 82.74%
F1-score: 0.84
```

Fourth Input removed-number of people

```
Accuracy: 72.89%
Precision: 63.47%
Recall: 62.33%
F1-score: 0.63
```

Fifth Input removed-lug boot rating

```
Accuracy: 87.52%
Precision: 83.00%
Recall: 83.18%
F1-score: 0.83
```

Sixth Input removed-safety rating

```
Accuracy: 70.33%
Precision: 60.43%
Recall: 56.50%
F1-score: 0.58
```

Results

I tested the accuracy of removing an input feature from the base dataset by removing the input column and then finding dataset's accuracy. I did this with each input feature just to test the accuracy and to see if it matches the input data feature accuracy. Unfortunately this was not the case. The lowest accuracy was the buying price according to the table, but when testing the input feature removal on the dataset the lowest accuracy was the safety rating.

11 Iteratively remove one feature at a time

buying price

```
Accuracy: 86.03%
Precision: 82.13%
Recall: 79.37%
F1-score: 0.81
```

buying price,maintenance

```
Accuracy: 87.69%
Precision: 84.94%
Recall: 80.94%
F1-score: 0.83
```

buying price,maintenance,number of doors

```
Accuracy: 85.95%
Precision: 81.65%
Recall: 79.82%
F1-score: 0.81
```

buying price,maintenance,number of doors,lug boot rating

```
Accuracy: 88.18%
Precision: 85.82%
Recall: 81.39%
F1-score: 0.84
```

buying price,maintenance,number of doors,lug boot rating,number of people

```
Accuracy: 86.86%
Precision: 83.76%
Recall: 79.82%
F1-score: 0.82
```

buying price,maintenance,number of doors,lug boot rating,number of people,safety rating

```
Accuracy: 87.77%
Precision: 85.99%
Recall: 79.82%
F1-score: 0.83
```

Results

The highest accuracy achieved was 88.18 percent and that was when buying price, maintenance, number of doors, and lug boot rating input features were removed. The lowest accuracy achieved was 85.95 and that was when buying price and maintenance was removed. The base case has an accuracy of 87.52 so removing buying price, maintenance, number of doors, and lug boot rating input features will make it perform better