Prediction over the combustible consumption (mpg) and predicting whether a car is able or not

Juan Luis Ruiz Vanegas
ENES Unidad Morelia
juanluisruiz971@comunidad.unam.mx

Figura 1: EDA and Machine Learning

[?]

RESUMEN

A car's efficiency can be measured based on the distance it is able to travel using a certain amount of fuel. Knowing the MPG (Miles per gallon) of a car gives us a picture of how efficient our car is.

Comparing the MPG of different makes and models is a good practice for the consumer who is interested in saving costs and car tax rates.

KEYWORDS

MPG, Car efficiency, Fuel Economy, Machine Learning, Exploratory Data Analisys, Binary classification

Car binary classification, 28 de abril de 2021, Morelia, Mich

ACM ISBN 978-1-4503-1234-5/17/07.
https://doi.org/10.1145/8888888.7777777

ACM Reference Format:
Juan Luis Ruiz Vanegas. 2021. Prediction over the combustible consumption (mpg) and predicting whether a car is able or not. In *Proceedings of Car binary classification.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/8888888.7777777

1. INTRODUCTION

Saving fuel is the main challenge facing many companies in the transportation industry. Don't forget that a large part of your business budget is spent on fuel, so creating a strategy that reduces dependence on fuel is the best way to significantly increase profits. [4gf [n.d.]].

As is evident, the first advantage of a reduction in fuel consumption is in the direct costs of fuel purchases. [4gf [n.d.]].

Why is it important to save fuel? You save money on gasoline purchases each year by choosing the most efficient vehicle for your needs.

Reduces climate change. Carbon dioxide (CO2) from the combustion of gasoline and diesel fuel contributes to global climate change.

Reduces the cost of dependence on oil. Our dependence on oil makes us vulnerable to oil market manipulation and price shocks.

Increases energy sustainability. Oil being a non-renewable resource, we cannot maintain the current rate of use indefinitely. Use it wisely now; it will give us time to find more sustainable alternative technologies and fuels.[fue [n.d.]],

Recent work on prediction has shown that excellent results are obtained using Machine Learning techniques [Antonio et al. 2010] or Data Mining techniques [Antonio et al. 2010], with Regression or Classification techniques such as Support Vector Machines (SVM) and Logistic Regression. In this work I also include Decision Trees and *Neural Networks.*

With these Classification and Regression techniques I will obtain a good binary estimator that based on the attributes of our *Dataset* manages to estimate the probability that a car has a good MPG performance (is_able). It also performs an Exploratory Data Analysis (*EDA*) as an Attribute Engineering technique to better understand the dataset.

2. METHODOLOGY

The methodology followed to carry out this work is based on [Gerón 2017]

The methodology followed is listed below.

1. Problem detection (section 2.1)
2. Data acquisition (section 2.2)
3. Exploración de los datos (sección 2.3)
4. Data exploration (section 2.3)
5. Tuning of model parameters (section 2.5)
6. Presentation of Results and Conclusions (section 3 and 4)

2.1. Problem detection

The objective of this work is to apply Classification and Regression techniques to obtain a good binary estimator that based on the attributes of our Dataset can estimate the probability that a car has a good efficiency based on MPG, choose which class is a car and based on properties of a car, determine the MPG. To carry it out, a Logistic Regression model, Decision Trees, Support Vector Machines were preliminarily chosen, since they are binary classifiers (which can also be used for multi-classification techniques) and belong to the most important supervised learning algorithms. In addition, I implemented a Neural Network to perform the MPG Regression. This kind of classifiers are easy to implement and are very promising for Classification and Regression tasks.

2.1.1. Logistic Regression. [Pant 2019] Logistic regression is a supervised learning algorithm for binary classification, which can be adapted to multi-class problems.

How does logistic regression work? Well, like a linear regression model, a logistic regression model calculates the weighted sum of the input vector plus a bias term, but instead of generating the result directly, it draws probabilities as shown in formula 1.

$$\hat{p} = h_\theta(x) = \sigma(\theta^T \cdot X) \tag{1}$$

Estimated probability of a logistic regression model

$$\sigma(t) = \frac{1}{1 + e(-t)} \tag{2}$$

Logistic regression ensures you converge to the solution, because the equation of the cost function is convex and therefore always reaches a global minimum regardless of the optimizer you use to minimize the cost function.

The important considerations when applying this model is that being a binary classifier is prone to be seen within an underfitting problem, since it could be a too simple model for our data and another problem to consider is that the classes do not get to be sufficiently balanced causing an overfitting problem.

Logistic Function A logistic regression model estimates the probability $\hat{p}$ that an instance X belongs to the positive class which makes the prediction of $\hat{Y}$ simple as shown in formula 3.

$$\hat{y} = \begin{cases} 0 & \text{Si } \hat{P} < ,5 \\ 1 & \text{Si } \hat{P} \geq ,5 \end{cases} \tag{3}$$

Note that $\sigma t \geq .5$ for values of x >0 since the RL model predicts 1 if $\theta_T \cdot X$ is positive and 0 otherwise.

2.1.2. Decision trees. [scikit learn.org [n.d.]] Decision trees is another supervised learning algorithm, whose objective is to predict the labels of our test set by means of decision rules that it learns from the training set.

It is a model that requires less data cleaning, its results are more intuitive (it also allows the option of visualization), the training cost is logarithmic and it supports categorical and numerical data. In addition, it can be used in multi-class problems.

$$E(X) = -\sum_{i=1}^{n} \frac{1}{3} log_2(1/3) - \sum_{i=1}^{n} \frac{2}{3} log_2(2/3) = .53 + .39 = .92[bit] \tag{4}$$

$$E(X) = -\sum_{i=1}^{n} \frac{log_2(1/2)}{2} - \sum_{i=1}^{n} \frac{log_2(1/2)}{2} = 1[bit] \tag{5}$$

2.1.3. Random trees. [Sharma 2020] It is a widely used classification and regression model that creates and joins several decision trees to create a forest. Its accuracy is not based on a single tree, but brings together the characteristics of several trees to improve its accuracy. It is less likely to over-fit as the number of trees increases. It can work with missing values and with categorical and numerical data.

2.1.4. Support Vector Machines (SVM). Support Vector Machines (SVM) is a powerful and versatile machine learning algorithm that can be applied to linear and nonlinear classification models, regression and outlier detection.

SVM models are efficient and work very well in most cases. The Scikit-Learn library has different SVM classes, but we are only going to be interested in the SVC class because, among other things, it is non-linear and adds the use of kernels.

Some of the most common kernels are:

$$Linear : K(a, b) = a^T b$$
$$Polynomial : K(a, b) = (\gamma a^T B + r)^d$$
$$GaussianRBF : K(a, b) = exp(-\gamma ||a - b||^2)$$

2.2. Data acquisition

The data were provided by the interviewer.

2.2.1. Workspace. For the working environment I used Jupyter Notebooks, Paperspace and Github (for version control); for the models I used specific Python modules such as Matplotlib, Pandas, Numpy and Scikit-Learn; and Fastai and PyTorch modules. The data and notebooks needed to replicate the results can be found in the data repository of the work.
https://github.com/jruiz971/Data-Science-Test-2021.4

All the analysis shown here was performed using a copy of the data and in the exploratory and EDA phase I made use of the pandas library.

The dataset has 398 instances and 11 attributes, of numerical and categorical type (for more details see figure 3).

Figure 3 shows a general description of the data using the *describe*() method that inherits the variable from the Dataframe, it is divided by rows and columns. In the columns we have each of the attributes of numeric type and in the rows the Standard Deviation (std) is shown, in turn in the rows 25 %, 50 % and 75 % show the corresponding percentiles. These are often referred to as the 25th percentile (or first quartile), the median and the 75th percentile or third quartile.

Some histograms extend much further to the left of the median than to the right (Figures 5 and 6), this can make it a bit more difficult for some Machine Learning algorithms to detect patterns.

```
data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 398 entries, 0 to 397
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   mpg           398 non-null    float64
 1   cylinders     398 non-null    int64
 2   displacement  398 non-null    float64
 3   horsepower    398 non-null    object
 4   weight        398 non-null    int64
 5   acceleration  398 non-null    float64
 6   model_year    398 non-null    int64
 7   origin        398 non-null    int64
 8   car_name      398 non-null    object
 9   class_mpg     398 non-null    int64
 10  is_able       398 non-null    int64
dtypes: float64(3), int64(6), object(2)
memory usage: 34.3+ KB
```

Figura 2: Description of the data type by attribute, and missing values

```
data.describe()
```

	mpg	cylinders	displacement	weight	acceleration	model_year	origin	class_mpg	is_able
count	398.000000	398.000000	398.000000	398.000000	398.000000	398.000000	398.000000	398.000000	398.000000
mean	23.514573	5.454774	193.425879	2970.424623	15.568090	76.010050	1.572864	1.027638	0.133166
std	7.815984	1.701004	104.269838	846.841774	2.757689	3.697627	0.802055	0.873189	0.340182
min	9.000000	3.000000	68.000000	1613.000000	8.000000	70.000000	1.000000	0.000000	0.000000
25%	17.500000	4.000000	104.250000	2223.750000	13.825000	73.000000	1.000000	0.000000	0.000000
50%	23.000000	4.000000	148.500000	2803.500000	15.500000	76.000000	1.000000	1.000000	0.000000
75%	29.000000	8.000000	262.000000	3608.000000	17.175000	79.000000	2.000000	2.000000	0.000000
max	46.600000	8.000000	455.000000	5140.000000	24.800000	82.000000	3.000000	3.000000	1.000000

Figura 3: Summary of all numeric type attributes

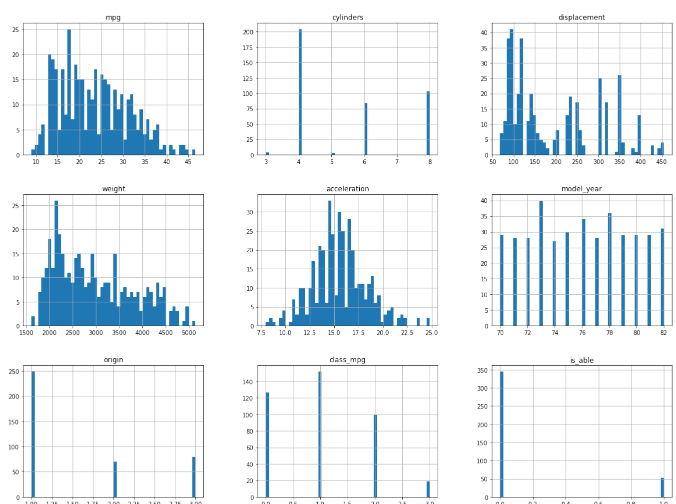


Figura 4: Histograms for each of the numerical attributes

An attempt is made to transform these attributes further to have more normal form distributions.

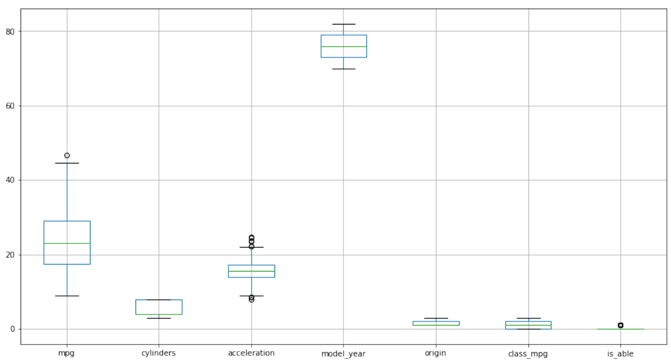


Figura 5: Outlier Detection

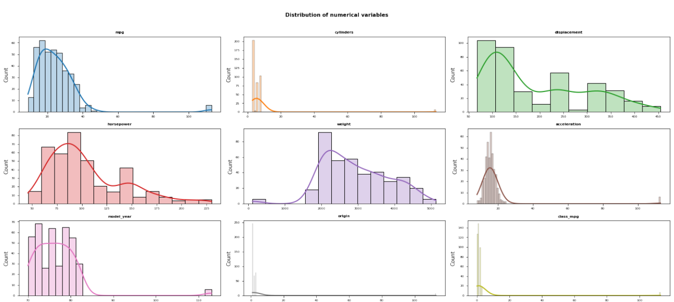


Figura 6: Distribution of the variables

2.3. EDA

Now that we have superficially seen the data, we begin an exploratory data analysis to better understand our Dataset.

We can observe several details from the Figure 6 of the distribution of our variables. First, that most of the cars have a performance close to 20 MPG, that most of the cars belong to the 70's decade. Also, Figure 7 shows the positive correlations between MPG and car model year, saying that newer registered cars have higher fuel efficiency. Negative correlations are also seen, such as horsepower vs. MPG, where the more horsepower, the lower the fuel efficiency.

The MPG vs Cylinders graph is curious, because it seems that the more cylinders, the more MPG, but it is not so, you see the distributions tend to go downwards, but the correlation line goes upwards due to the presence of an outlier.

Because of the interesting things we could observe, I made a deeper study among the correlations between variables (Figure 9). In the heat map are the indexes that indicate how much one variable influences another. We see again interesting details, such as that acceleration is strongly related to MPG, the more cylinders, the less travel, or that weight has a strong influence on the car's displacement.

The range of values within a linear correlation varies between -1 and 1, if this value is close to 1, it means that the variable is positively correlated, on the other hand if the variable is close to -1 it means that it is negatively correlated and if it is 0, there is absolutely no correlation. The attributes with the highest positive and negative correlation are plotted (see figure 9) because they are the most promising attributes.

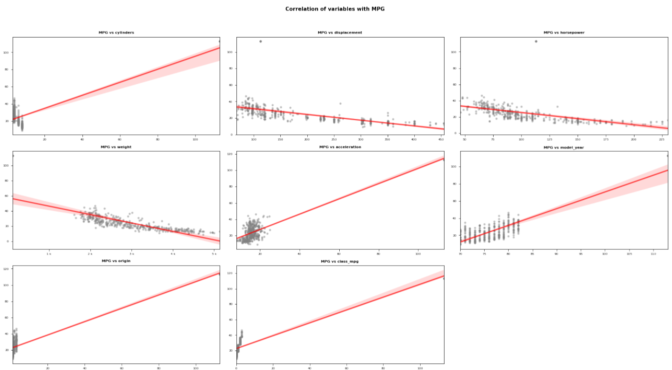


Figura 7: Correlation of variables with MPG

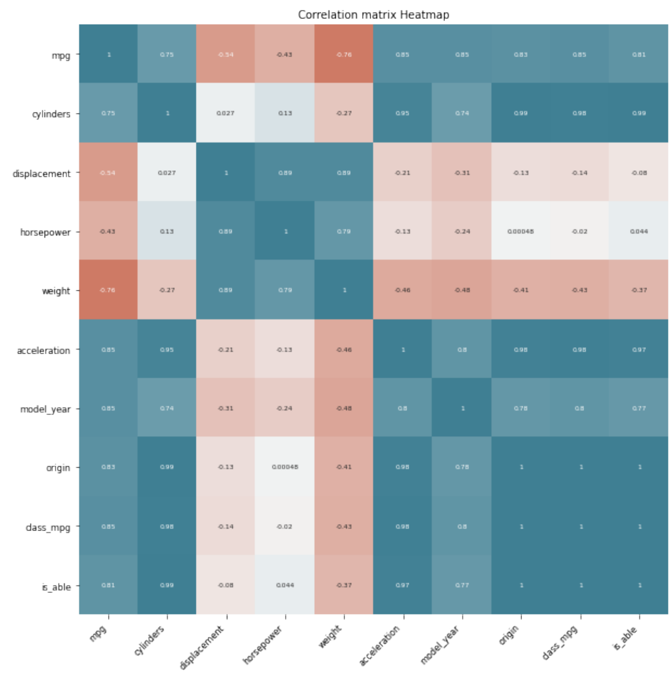


Figura 8: Correlations between variables

With Exploratory Data Analysis we can have information and intuitions about the data.

2.4. Data preparation

First, in order to make better predictions about the cars, I decided to encode them to take advantage of the name columns.

The horsepower variable was in object type because it contained null values, which I filled with the mean and changed the data type to int.

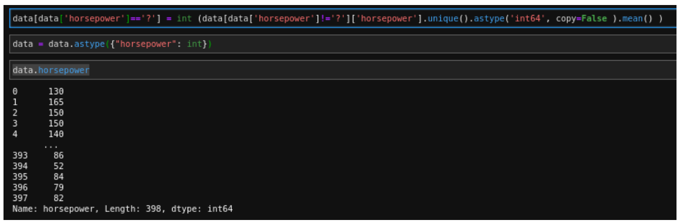
2.4.1. One-Hot Encoding. This is an encoding method that can be applied to the label that has your data set and thus obtain a vector

```
data = pd.concat([data,pd.get_dummies(data.car_name)],axis=1)
data
```

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	car_name	class_mpg	...	volvo 145e (sw)	volvo 244dl
0	18.0	8	307.0	130	3504	12.0	70	1	chevrolet chevelle malibu	0	...	0	0
1	15.0	8	350.0	165	3693	11.5	70	1	buick skylark 320	0	...	0	0
2	18.0	8	318.0	150	3436	11.0	70	1	plymouth satellite	0	...	0	0
3	16.0	8	304.0	150	3433	12.0	70	1	amc rebel sst	0	...	0	0
4	17.0	8	302.0	140	3449	10.5	70	1	ford torino	0	...	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
393	27.0	4	140.0	86	2790	15.6	82	1	ford mustang gl	1	...	0	0
394	44.0	4	97.0	52	2130	24.6	82	2	vw pickup	3	...	0	0
395	32.0	4	135.0	84	2295	11.6	82	1	dodge rampage	2	...	0	0
396	28.0	4	120.0	79	2625	18.6	82	1	ford ranger	2	...	0	0
397	31.0	4	119.0	82	2720	19.4	82	1	chevy s-10	2	...	0	0

398 rows × 313 columns

Figura 9: Names encoder



```
data[data['horsepower']=='?'] = int (data[data['horsepower']!='?']['horsepower'].unique().astype('int64', copy=False ).mean() )

data = data.astype({"horsepower": int})

data.horsepower

0      130
1      165
2      150
3      150
4      140
      ...
393     86
394     52
395     84
396     79
397     82
Name: horsepower, Length: 398, dtype: int64
```

Figura 10: Change type for horsepower

of 0's and only a 1 in the position of the correct or predicted class, ie, in our data set we have the following classes [0, 1,2,3,4,5,6,7,8,9], with OHE you can encode for example an image that has as class a 6 the following vector [0,0,0,0,0,0,0,1,0,0,0,0,0,0] and thus only activate the correct or predicted class [Gerón 2017].

Following this point we apply the OHE to our categorical attributes in such a way that we generate new attributes for each categorical value

2.5. Afinación de los parámetros del modelo

As a preliminary analysis, the various ML models were trained with the default parameters. Once the best combination was obtained, the parameters of the models were fine-tuned, obtaining the following results.

Tuned Models Results	
Model	Acc
Logistic Regression	98.75 %
Decision tree	92.5 %
Random trees	0.975 %
SVM	0.975 %

3. RESULTS

3.1. ROC Curves,AUC

For the evaluation and comparison of our models, we will rely on the ROC curves presented in the following section.

This is one of the most important evaluation metrics to verify the performance of classification models and allows us to visualize the results. The ROC curve is created by plotting on the *xaxis* the false positives and on the *yaxis* the true positives. The AUC curve

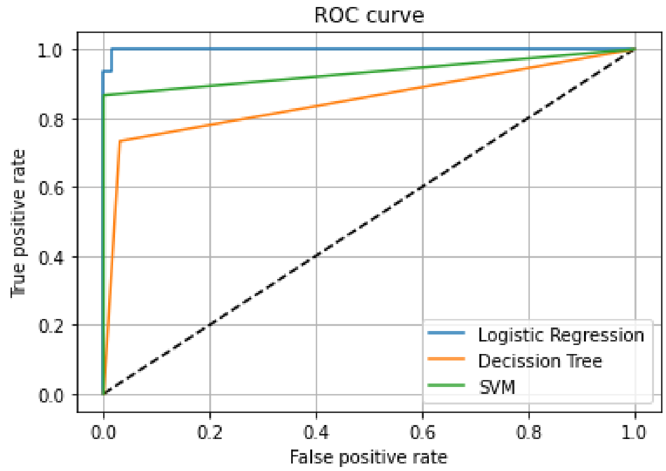


Figura 11: ROC Curve

indicates how good our model is at distinguishing between classes. AUC values to take into account:[Narkhede 2020]

- The closer AUC values are to 1, the better the model will be for classification.
- If the value is 0, sort the labels inversely, i.e., 0->1; 1->0.
- If the value is 0.5, the model does not have the ability to separate classes.

After evaluation with ROC and AUC, the results were as follows:

Evaluación usando AUC	
Model	Acc
Logistic Regression	96.66 %
Decision tree	85.12 %
Random trees	93.33 %
SVM	93.33 %

3.2. Neural Network

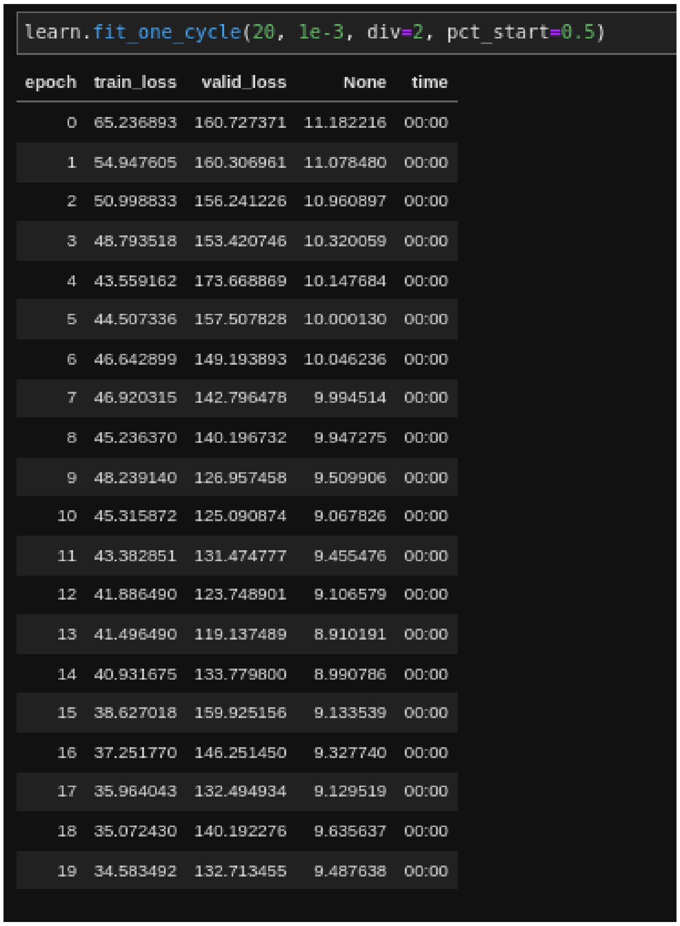
The Neural Network results for the Regression, using the L1 metric and the MSELoss function as the loss function, after 90 training cycles, had a performance accuracy of +- 9.48 MPG (rerunning the Notebook will have similar, but not the same results).

3.3. Predictions

After making the models and training them, they can be used to make predictions about new data. For this I developed code that gives the input format to the properties of the learned models to make the prediction on them. In the case of $is_{a}ble$ it prints the probability prediction of the classes and for MPG, it prints a single value, which is the predicted one.

4. CONCLUSIONS

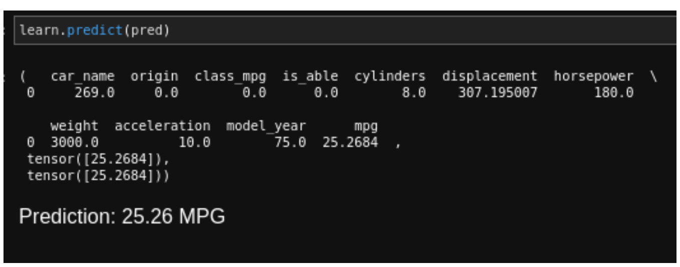
When conducting a study on a Dataset it is important to perform an Exploratory Data Analysis to apply Attribute Engineering techniques and obtain important information about the behavior of the data, as well as to preprocess them to improve the performance of the models. And regarding these, it is important to know them and their parameters, to adjust them to the problem to be solved.



```
learn.fit_one_cycle(20, 1e-3, div=2, pct_start=0.5)
```

epoch	train_loss	valid_loss	None	time
0	65.236893	160.727371	11.182216	00:00
1	54.947605	160.306961	11.078480	00:00
2	50.998833	156.241226	10.960897	00:00
3	48.793518	153.420746	10.320059	00:00
4	43.559162	173.668869	10.147684	00:00
5	44.507336	157.507828	10.000130	00:00
6	46.642899	149.193893	10.046236	00:00
7	46.920315	142.796478	9.994514	00:00
8	45.236370	140.196732	9.947275	00:00
9	48.239140	126.957458	9.509906	00:00
10	45.315872	125.090874	9.067826	00:00
11	43.382851	131.474777	9.455476	00:00
12	41.886490	123.748901	9.106579	00:00
13	41.496490	119.137489	8.910191	00:00
14	40.931675	133.779800	8.990786	00:00
15	38.627018	159.925156	9.133539	00:00
16	37.251770	146.251450	9.327740	00:00
17	35.964043	132.494934	9.129519	00:00
18	35.072430	140.192276	9.635637	00:00
19	34.583492	132.713455	9.487638	00:00

Figura 12: l1 Loss, MPG



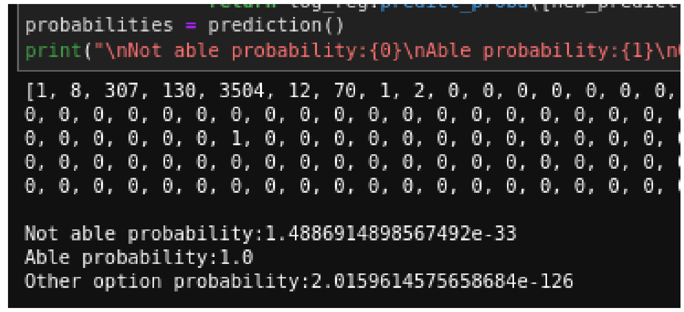
```
learn.predict(pred)

(   car_name  origin  class_mpg  is_able  cylinders  displacement  horsepower  \
 0     269.0     0.0        0.0      0.0        8.0    307.195007       180.0

    weight  acceleration  model_year      mpg
 0  3000.0          10.0        75.0  25.2684  ,
 tensor([25.2684]),
 tensor([25.2684]))

Prediction: 25.26 MPG
```

Figura 13: MPG Prediction

Once the results of the models are obtained, they must be evaluated to identify if the problems of False Positives and True Positives are solved.

REFERENCIAS

[n.d.]. *La importancia del ahorro de combustible.* https://www.4gflota.com/blog/la-importancia-del-ahorro-de-combustible/
[n.d.]. *¿Por qué es importante ahorrar combustible?* https://www.fueleconomy.gov/feg/eswhy.shtml
Nuno Antonio, Ana De Almeida, and Luis Nunes. 2010. Predicting hotel booking cancellations to decrease uncertainty and increase revenue. *Tourism Management Studies* (2010). https://doi.org/10.1016/j.ejor.2009.06.006



```
probabilities = prediction()
print("\nNot able probability:{0}\nAble probability:{1}\n

[1, 8, 307, 130, 3504, 12, 70, 1, 2, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

Not able probability:1.4886914898567492e-33
Able probability:1.0
Other option probability:2.0159614575658684e-126
```

Figura 14: ïs$_{a}$*ble”Prediction*

Aurelín Gerón. 2017. *Hands-On Machine Learning with Scikit-learn and Tensorflow.* ='O'Reilley Media Inc, 1005 Gravenstein Highway North, Sebastopol, CA 95472.
Sarang Narkhede. 2020. Understanding AUC - ROC Curve. https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5. [Online; accessed 9-July-2020].
Ayush Pant. 2019. Introduction to Logistic Regression. https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148. [Online; accessed 6-July-2020].
scikit learn.org. [n.d.]. 1.10. Decision Trees. https://scikit-learn.org/stable/modules/tree.html. [Online; accessed 6-July-2020].
Abhishek Sharma. 2020. Decision Tree vs. Random Forest – Which Algorithm Should you Use? https://www.analyticsvidhya.com/blog/2020/05/decision-tree-vs-random-forest-algorithm/. [Online; accessed 5-July-2020].