# Predicting the price of the 70 best-selling US cars

Paolo Mariani[1], Matteo Licciardello[2]

**Abstract**

In this paper is reported a study conducted on prediction of listed price in online vehicles advertisement, with particular reference to Craigslist data. In the paper's first section, a discussion regarding predictive models and causal models will take place, immediately followed by an overview about analysed data and most of the preprocessing techniques used to clean and, generally, manipulate it. In the second part will be presented two different machine learning approaches used to achieve the established goal and the results obtained.

**Keywords**

Machine Learning – US Cars – Price Prediction – Project Report – Classification – Regression

[1,2] *Dipartimento di Informatica Sistemistica e Comunicazione, CdLM Data Science, Università degli Studi di Milano-Bicocca*

1 - p.mariani20@campus.unimib.it

2 - m.licciardello@campus.unimib.it

## Contents

## Introduction

Nowadays advertising of used cars on specialized websites is a common sales activity and choosing the correct price for online sales items is a crucial step for the outcome of the sale. Craigslist [1] is an American classified advertisements website which offers sales services for several different contexts, created in 1995 by Craig Newmark [2] as e-mail lists distribution system. Today, Craigslist is a web-based service that counts over 70 different countries across the globe, used to advertise houses, items wanted, services and events. One of the most used services is the cars & trucks advertising section, which guarantees to a lot of US citizens to promote the sale of their vehicle in a specified US state or region. An interesting task, based on the data collected on these websites, is to perform an analysis of which models are the most advertised and to discover at which price they are listed. Kaggle's [3] community managed to extract a huge amount of data related to this advertisement activity by web-scraping the Craigslist: the dataset has been published on the official website in 2017 and is continuously maintained with new update. In this paper, will be proposed a classification task about the price of each vehicle given the information of other attributes, such as geographic location and specs of the car. In particular, task will be carried out with two different approaches: in the first one, the classification of the price will be evaluated using accuracy and grouping the price attribute; in the second one, the result of the prediction will be evaluated with a distance measure.

### Prediction vs Causal Inference

The research question treated in this report is about predicting the correct price of a vehicle with several machine learning algorithms, in such a way that is possible to integrate the predicting tool in a support system that helps users establish the correct price for the vehicle that they want to advertise or purchase. This kind of task is a classic prediction activity that answers to the research question ”*What is the ideal price for this particular vehicle*?”, there is no theoretical model which is used to establish the amount of a causal impact between the variables, but only a mathematical model applied on a given dataset to predict new or future observations using a set of attributes. Unlike what just described, a causal inference task is based on a theoretical model that implies the study of a causal effect between a set of variables and a dependent outcome: to measure impact of explanatory variables on a dependent variable is required that one explanatory variable at a time is altered in its values while the other ones are kept constant. If the treated problem was a causal inference problem, then the research question should have been different from the actual one, it would have been something like ”*How does the price change given a variation of one or more specs of the vehicle?*”.

## 1. Explanatory Data Analysis

The considered dataset contains over 500 thousands records and 25 class attributes, for a data total size of 1GB. Only a subset of variables is used for the prediction task, while others are dropped: information such as web-page *URLs*, *Region URLs* and *Image URLs* are not relevant for the chosen task; the textual *Description* is dropped because there is no interest in carrying out a text mining task, *VIN* and *IDs* are identifier variables that must be deleted to obtain an unbiased classifier. After a brief analysis even the variable *Size* is dropped due to a massive percentage of NaN values, *Latitude* and *Longitude* are also dropped because the information about the position is already explained by other variables. The variables considered that have been kept are the followings:

- *Price*: target variable, price established in the advertisement.
- *Region* and *State*: indicates the geographic area.
- *Year*: year of production of the vehicle
- *Manufacturer* and *Model*: car’s manufacturer and model
- *Condition* and *Title Status*: external and structural condition as declared by owner.
- *Cylinders*: number of cylinders of the engine.
- *Fuel*: type of fuel.
- *Odometer*: miles traveled by the car.
- *Transmission*: type of drive actuator of the car.
- *Drive*: traction wheels output.
- *Type*: vehicle’s design shape.
- *Paint Color*: car’s color.

The dataset analysis has leaded to discover important insights and to consider some preprocessing operations. Regarding the target variable, a lot of observations were equal to zero or have excessive or improbable prices: those have been removed together with the outliers values. Only for the first prediction task, the price variable has even been binned with equal size binning (5000$ width each one), instead, for the second one it has been rounded to the thousands, because the goal is to predict an estimate and not a specific value. For the other variables have been conducted some preprocessing operations which include NaNs removal and recover, with techniques that involve replacing the missing values based on some conditions: for example the NaNs in the variable *Cylinders* have been fixed with a value that is given by a set of conditions based on the vehicle model and the engine fuel; other attributes that have been fixed with this technique with other parameters are: *Drive*, *Fuel*, *Type* and *Transmission*. Other variables, instead, have been fixed by binning its values by different considerations, like the variable *Odometer*; others that were in string format, like the *Model*, were uniformed to unique values by the value of the variable *Manufacturer* (Eg. the value ”1500” for the manufacturer ”Chevrolet” is replaced with the specific value ”Silverado 1500”). The dataset has been filtered with the top-70 models listed on the Craigslist website, and the final dataset obtained contains 140503 records by 15 different features. Finally, has been computed a correlation matrix which showed no particular correlation between the selected features. All the steps of the analysis and the plots that have been produced for the discussed variables are available in the preprocessing notebook [4], under the dedicated section *Explanatory Data Analysis*.

## 2. Price Prediction

As discussed before, the focus of this study is to conduct a prediction task of the price given different features, in two different ways. In this particular task the target variable is split in 15 bins, the goal is to classify the correct bin for each observation. As can be observed from the image 1, the *Price* variable (each bin represents 5000$ more than the previous) is extremely unbalanced: this can lead to multiple problems during the execution of the classification.

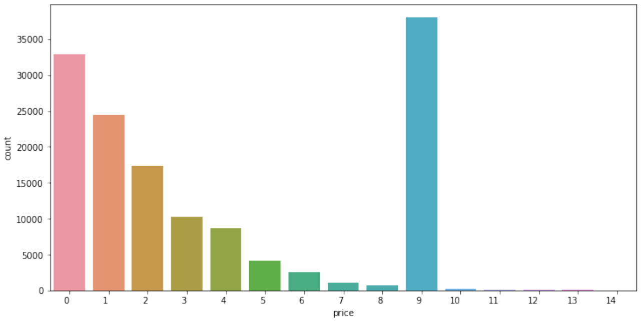

**Figure 1.** Binned price distribution

From a large set of different classification techniques and algorithms, only three of them are selected. All tests are replicable by setting the same random seed for all the algorithm.

### 2.1 Random Forest

The Random Forest classifier, taken from Sklearn library [5], is an heuristic-based algorithm which uses a set of Decision Trees to make a prediction, and selects the outcome as the most frequent class predicted by its forest. Each Decision Tree of the forest is uncorrelated to each other, is able to make a prediction by splitting the feature space with different conditions, in which each one creates two branches. Parameters used to train this classifier are:

- Number of trees: 100
- Max Depth of each tree: 24

The result obtained by the classifier on the split dataset (80-20 train-test split) by testing on the test set is 75.0%. The same classifier is then trained and tested on a 10-fold cross validation, the mean result obtained is 75.2%±0.004%

### 2.2 XGBoost

The XGBoost classifier is an iterative algorithm based on the boosting approach and the Decision Tree classifier, this means that it is composed by many Decision Trees as the Random forest but (as is done on an ensemble technique) tries to get better iteration after iteration by boosting trains models in succession, with each new model being trained to correct the errors made by the previous ones. This particular classifier is trained with the following parameters:

- Max Depth: 30
- Number of trees: 50
- Learning rate: 0.1
- Gamma: 0
- Loss function: ”Multi:softprob”

The result obtained by the classifier on the split dataset (80-20 train-test split) by testing on the test set is 74.0%. The same classifier is then trained and tested on a 10-fold cross validation, the mean result obtained is 74.2%±0.004%

### 2.3 Artificial Neural Network

The last classifier applied is the Artificial Neural Network, taken from Keras library [6]. This model simulates the training method of a natural brain, learning more and more as the algorithm iterate. The neural network training phase is very heavy in terms of computations due to his huge number of parameters to train. The parameters selected for this model are:

- Optimizer: Adam, with learning rate set at default value
- Loss function: ”Sparse Categorical Cross-Entropy”
- Metrics: Accuracy

The result obtained by the classifier on the split dataset (80-20 train-test split) by testing on the test set is 64.0%. The same classifier is then trained and tested on a 10-fold cross validation, the mean result obtained is 64.5%±0.36%.

### 2.4 Explainable AI

The framework *Skater* [7] is a library introduced to help developers to understand better what the prediction models do globally when they have to take decision based on the set of explainable features. This software gets in input a set of data (in this study nearly 10% of the original dataset is given) and, after perturbing the original data, gives in output the feature importance and relative plot for each classifier. These plots show that the most important features for the prediction are *Year* and *Odometer* for all three models, the extended results can be seen in the section *Expainable AI* of the notebook [8].

## 3. RMSE Evaluated Prediction

The second presented approach is a metric distance-based evaluation, in particular referred to Root Mean Square Error: it represents the quadratic mean of the differences between predicted and observed values and is commonly used to assess model’s prediction with actual observation. In this particular case, the price attribute is rounded to the nearest thousand to have less possible values, to help algorithm in computation at cost of a small systematic error and return an indicative value, not a specific one. The distribution of the price can be seen in the image 2.

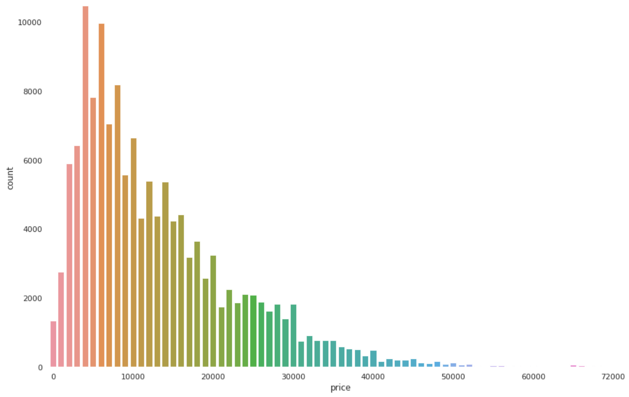

**Figure 2.** Price distribution

### 3.1 Classification models

The selected classifiers for this second approach are the same as the first approach but with different parameters: in this case all the algorithms are set to regression mode with specific metric to minimize RMSE and a different loss function to optimize: *reg:squarederror*. The classifiers also have the same hyper-parameters, except the neural network which has a different architecture [8] and different hyper-parameters to better fit the task. It is also trained with the ”MSE” loss function and Adam optimizer with learning rate set to 0.001. In this case, K-fold cross validation was not operated due to high computational cost, but all tests are replicable by setting the same random seed for all the algorithm. The RMSE obtained by the three models are:

- Random Forest: 2924.05
- XGBoost: 2912.59
- Neural Network: 3516.08

## 4. Conclusion

As shown in the first approach, the classification of the price gives very close results for Random Forest and XGBoost, with the Neural Network farther; the first two classifiers could be indifferently used to predict the price of a listed vehicle. Analysing the results of the second approach, it can be seen that the RMSE is sufficiently low for all of the three classifiers that are really close together, with XGBoost and Random Forest newly at non-significance distance followed by the Neural Network.

In conclusion, we can establish that both approaches are a valid solution to predict the price of a vehicle listed on advertising lists such as Craigslist. Talking about classifiers, both Random Forest and XGBoost represent good prevision models despite the class unbalance, the same cannot be said on the Neural Network that is more difficult to adapt to the problem, it should require further studies.

## References

[1] Craigslist. craigslist.org website. craigslist.org.

[2] Craig Newmark. Craig newmark philanthropies. craignewmarkphilanthropies.org.

[3] Kaggle. Kaggle: Your home for data science. kaggle.

[4] Matteo Licciardello, Paolo Mariani. Preprocessing notebook. NB1.

[5] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, and Alexandre Gramfort. scikit-learn project. pages 108–122, 2013.

[6] François Chollet et al. Keras. keras, 2015.

[7] Aaron Kramer and Pramit Choudhary. Skater xai. github.com/oracle/Skater.

[8] Matteo Licciardello, Paolo Mariani. Preprocessing notebook. NB2.