MULTI-MODEL MACHINE LEARNING BASED APPROACH TO USED VEHICLE PRICING

Andreas Anglin **Ben Mazerolle** **Darian Morrison** **Kush Gautam**

Subah Mehrotra

Abstract

With over 12.3 billion dollars in used car sales from dealerships alone last year in Canada [1], the market for pre-owned vehicles is a massive, low-regulation "wild west" that many consumers find difficult to navigate. Buyers and sellers, especially those in the private market, have long struggled with price negotiations due to the completely free market nature of used item sales. Demand is dictated entirely by the consumer who must decide what attributes make a vehicle desirable, and most importantly, what a representative price of that value is. Our project strives to solve the problem of pricing used vehicles by determining what features determine a vehicle's value. In this report, K Means Clustering, Random Forests, and a Neural Network are modelled to attempt to accurately predict a vehicle's price given a number of common attributes. We evaluate the success of the proposed models using common metrics such as average error as well as model specific metrics such as epoch accuracy (Neural Network) and prediction accuracy (Random Forest). The results demonstrate the relative ineffectiveness of K means clustering and Random Forests for the problem and the greater effectiveness of Neural Networks for the prediction of a vehicle's value. The code for the project is available at https://github.com/andre3racks/car-value-prediction.

1 Introduction

There were over 24.6 million road vehicles registered in Canada (2018) [2]. Given that people switch cars every 6.4 years on average [3], roughly 4 million Canadians are looking for a new vehicle every year. Since the average age of a vehicle on the road is 11 years, a large percentage of the 4 million vehicles purchased are used, a massive market. Typically, the value of a used car is almost entirely driven by market demand, and a "reasonable price" is decided by what other vehicles with similar characteristics are priced at. However, this process leaves a lot of questions for buyers and sellers. What features are more important than others? Is the market price truly indicative of a vehicle's value? How old of a vehicle should be purchased?

Most vehicle price prediction models rely on three primary metrics to determine the value: the model of the vehicle, the condition of the vehicle, and its age. However, most consumers have interest in a number of other features not captured in these low-precision models. A fluorescent pink pickup truck is likely not as valuable as a black one due to the higher demand for black trucks. Humans are fickle and often illogical, and given the status symbol that a vehicle can represent, attributes of a vehicle's quality such as gas mileage and safety do not always ultimately

determine the value of a car. In this report, we aim to determine exactly what attributes are desirable, and more precisely, what dollar value we assign to those attributes.

To this end, three distinct models will be implemented in an attempt to create an accurate regression for vehicle prices. Unsupervised K-Means clustering with varied cluster numbers to produce the most accurate pricing clusters possible while maintaining the precision of price estimate. A regression-output Neural Network will be implemented with varying hyperparameters such as layer depth. Finally, a Random Forest is also modelled and run on the dataset.

To train and test the above models, three different reference datasets will be used. A kaggle-sourced dataset containing all active Craigslist vehicle ads from the United States will be the project's source of data given its "overabundance" of information, containing 25 distinct attributes per entry [4]. However, it also requires a great deal of cleaning and regularization as Craigslist allows for manual attribute entry, producing many incorrect or empty attribute entries. This large number of columns will allow the models to be fine-tuned to determine the attributes that are of utmost importance when determining value. The second dataset will only take into account the mechanical aspects of listed vehicles. As the third dataset, the full dataset will be trimmed to mimic the characteristics that autotrader.ca vehicles use for pricing data as autotrader.ca is a well-known source for pricing information.

The main topics covered by this report are as follows:

- We evaluate the regularization and selection of data to trim unnecessary features captured in the original dataset.
- We tune the hyperparameters of a Neural Network, Random Forest, and K-Means Clustering model to produce the most accurate models possible for car price prediction.
- We compare the results of the three above methodologies to propose the most effective method of predicting a user vehicle's price.

2 Related work

Many websites currently exist to appraise the value of a vehicle given its make, model, year, and condition. They vary in specificity and attributes used to determine price, producing a non-uniform dataset set of "correct prices" for buyers. Autotrader.ca [5] has a "What's my car worth?" tool that uses the make, model, year, trim, and condition of the vehicle to determine its value. The Canadian Black Book, often considered the original car appraisal tool, provides a similar service, showing an average price given the make and model [6]. Perhaps the most accurate tool available to consumers today are dealer appraisals themselves, who price a specific car that they are either selling or looking to purchase from the consumer. However, this is a black box appraisal, and different dealerships value different attributes of a vehicle, making it difficult to determine the formula, if one exists, that is used. Furthermore, there is a great deal of self-interest built into the appraisal, as dealers will often undervalue a trade-ins value and "make up the difference" by offering the consumer that add-ons to their new vehicle that are worth much less to the dealer than the the amount of money they are saving on the low trade-in offer [7].

There are also numerous machine-learning based approaches to open source used car price predictions, many of which have used random forests, ridge regression, and XGBoost [8, 9].

However, most of these approaches have lacked a clustering-based approach to their methods, which is executed and summarized in this report. Similarly to many of the listed approaches, we use Neural Network and Random Forest as comparative methods to evaluate the success of our clustering approach.

3 Proposed Methods

Data

A dataset of Craigslist listings on Kaggle with 450,000 samples is used as our source of data. The dataset is cleaned in order to handle inconsistent user input and is optimized for machine learning. The steps taken for cleaning the data are to standardize data, remove unnecessary features, prune samples with missing features and then normalize the data to values between 0 and 1.

Inconsistency among certain key features requires some method of standardization to ensure equivalent feature values aren't treated as distinct values. This is done by selecting a list of accepted values for a given feature, simplifying both the actual and accepted values by downcasing and removing symbols and then performing keyword searches on the actual values with the accepted values. This is primarily present in the "model" column.

Once the data is standardized it is split into subsets to prevent data loss when pruning samples with one or more missing features and to see if overfitting would occur. Three data sets are created as subsets: the first has nearly all the features, the second uses the features Autotrader uses for their appraisal tool and the third includes all the mechanical features, avoiding the manufacturer and model. Upon feature selection, each dataset then has samples with missing features purged.

After feature selection, we need a way to handle discrete and categorical data. Discrete columns that can be treated as linear are turned into integer values (I.e. a vehicle's condition with values of excellent, fair and poor were mapped to 2, 1 and 0 respectively). Other columns with strictly categorical data use one-hot encoding [10] to map categories to new binary features.

Finally, each feature set is normalized to values between 0 and 1 in order to optimize model performance. This is done by taking each feature column and dividing it by the maximum feature value from the entire dataset. After cleaning the data we are left with the "Full" dataset that has 14 of the original features with 117,548 samples, the "Mechanical" dataset having 9 features with 151,111 samples and the "Autotrader" dataset that has 6 features with 196,865 samples.

K Means

The first algorithm utilized for prediction is K Means, or Lloyd's Algorithm. K Means is an unsupervised clustering algorithm and requires additional logic for continuous variable prediction. Initially, the K Means model is fit for the training examples with 'K-Means++' as the initialization function. 'K-Means++' assigns initial centroid locations more intelligently than its alternative of random locations. Better initial centroid locations lead to a faster convergence. After the training data is clustered all examples in each cluster are averaged to assign a price, or label, to each

cluster. Averaging is done in two ways. The first is simple averaging, calculated by the summation of labels in each cluster divided by the number of items in that cluster. The second is a weighted average. The weights for each example are equal to the reciprocal of the euclidean distance between the given example and its cluster divided by the summation of weights for the given example's cluster. Parameter tuning is done in 3 stages. The first two utilize KFold Cross Validation to reduce the range of 'K' for finding optimal model(s) across an initial range of 0-500. The third stage trains and tests a small collection of models across 10 'K' values on the entire dataset. The best performers from this third stage are recorded by Mean Absolute Error, Model Acceptance Rate, and cost. This procedure is done for all datasets across both methods of averaging to determine an optimal model.

Neural Network

The second model used for prediction is a Neural Network, which is a supervised classification algorithm containing multiple hidden layers and neurons. Our Neural Network model is created with 6 hidden layers taking different input values for different datasets. The activation functions used are "linear" and "relu" because they are cheap to compute and converge faster. The optimization method used is "Adam", which is an algorithm for first-order gradient-based optimization of stochastic objective functions. These parameters are further discussed under Experiments. "Mean absolute error" is used for the loss function that gives the sum of absolute differences between the target and the predicted values. Using all these functions along with a subset of the training data as validation set, we performed model fitting and tuned the parameters to obtain the best results.

Random Forest

A Random Forest Regressor is the third and final model used for price prediction on the dataset. Multiple hyperparameters are tuned to optimize the performance of the model. The maximum features being used for splitting is set to the number of features, as the datasets themselves have pre-trimmed columns. The other pre-graphing Forest attributes that are calculated are minimum samples needed to split and create a leaf, the maximum tree depth, and the minimum impurity split. Given the determination of the ideal values for these hyperparameters, two hyperparameters are varied and shown in the results below. The split criterion being used is varied between mean squared error (mse) and mean absolute error (mae). The number of trees used in the forest vary between 1 and 100000. All results are confirmed using K fold validation for a value of K between 5 and 10.

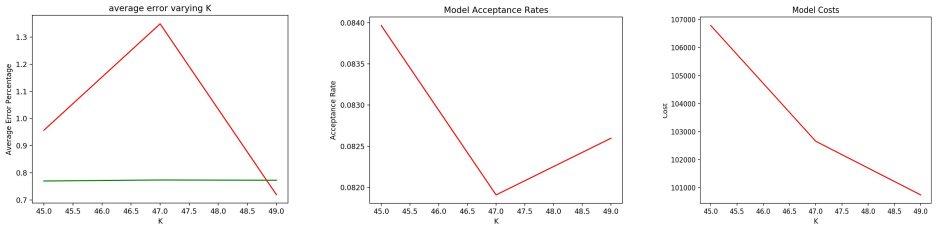
4 Experiments

K Means

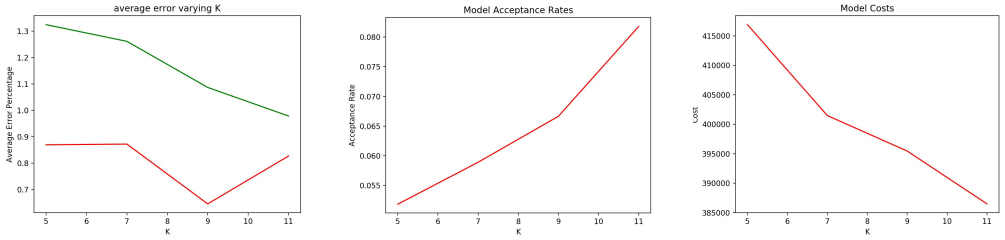
After two stages of K Fold cross validation, a range of K values (less than or equal to 10) is determined for the minimal mean absolute error of each dataset and averaging method pair. The

respective model groups are trained and tested on the entire dataset for each aforementioned pairing. Weighted averaging outperformed simple averaging for all datasets and therefore only the models which utilized weighted averaging are described below (Figures 1-9). Mean absolute error (where a value of 0.9 is equivalent to 90% mean absolute error) is graphed in red for testing performance and in green for training performance. It should be noted that the mean absolute error, for both testing and training, of the models during all stages are prone to unpredictable spikes. Due to this, only the best performers from this final stage are graphed.

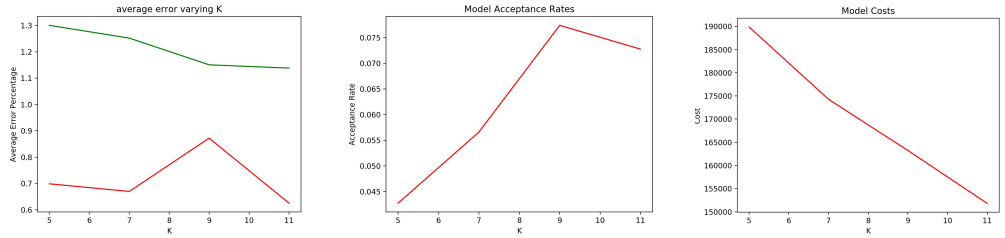
Figure 1 shows the metrics for optimal models with Weighted Averaging for each dataset.



(a) Mean Absolute Error, Model Acceptance Rates, and Cost for optimal models for **'Autotrader'** Dataset with Weighted Averaging



(b) Mean Absolute Error, Model Acceptance Rates, and Cost for optimal models for **'Full'** Dataset with Weighted Averaging

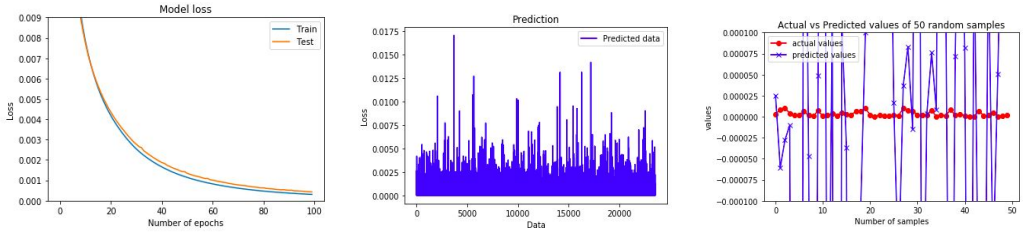


(c) Mean Absolute Error, Model Acceptance Rates, and Cost for optimal models for **'Mechanical'** Dataset with Weighted Averaging

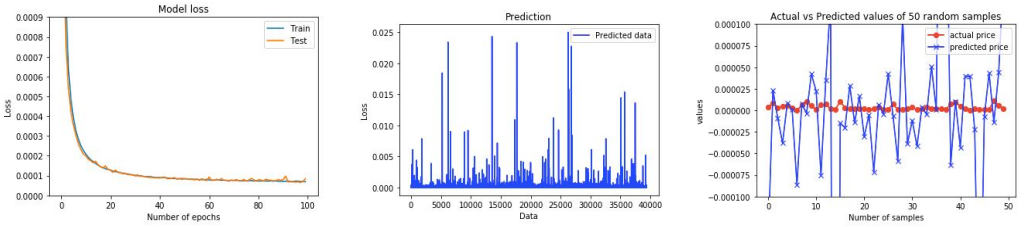
Figure 1: (a) An optimal MAE of 73% is reached through K=49. (b) An optimal MPE of 65% is reached through K=9. (c) An optimal MPE of 63% is reached through K=11.

Neural Network

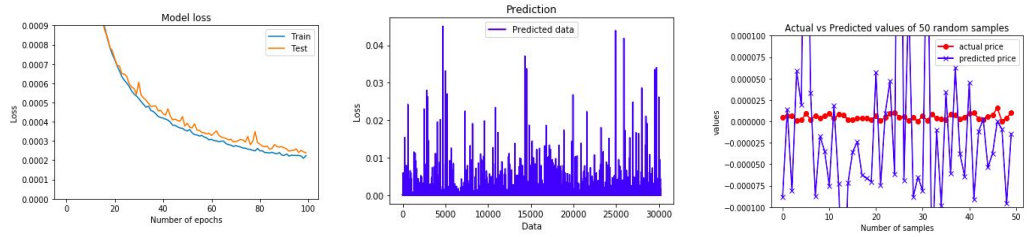
After parameter tuning the best results are seen with 6 hidden layers, with the first 4 layers using Rectified Linear Units (Relu) and the last 2 layers using a linear function as the activation functions. Different activations allow for different nonlinearities that work better for this problem. Learning rate also has a significant effect on the prediction and is set to a small value of 0.00001. The model is run on three different datasets and the following graphs are plotted for each dataset: Loss/Accuracy vs Number of epochs, Mean absolute error of prediction vs Data and Actual vs Predicted values of 50 random samples. The loss value decreases drastically as the number of epochs increase from 0 - 40 and then the rate of decrease slows down for all the datasets. The mean absolute error graph represents the amount of loss/error in the prediction. For the Actual vs Predicted values graph, some of the predictions are seen to be less than 0. This is because the last 2 layers using the linear activation function resulted in a few negative weights in the network. "Autotrader" dataset resulted in the best prediction with the least amount of loss and most amount of predicted values lying near the actual values.



(a) Model Loss, MAE, Actual Vs Predicted Values for the **'Full'** dataset



(b) Model Loss, MAE, Actual Vs Predicted Values for the **'Autotrader'** dataset

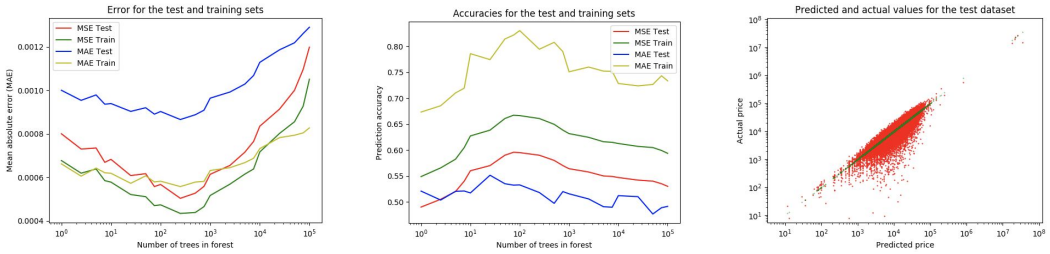


(c) Model Loss, MAE, Actual Vs Predicted Values for the **'Mechanical'** dataset

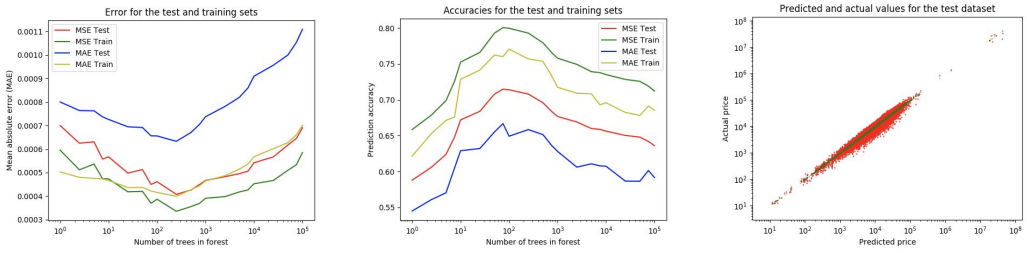
Figure 2: Qualitative visualization of the effectiveness of optimal Neural Network model on each test datasets. (a): The full dataset with 14 attributes achieved an accuracy of 56.23%. (b): The autotrader dataset with 6 attributes achieved an accuracy of 93%. (c): The mechanical dataset achieved an accuracy of 74.63%.

Random Forest

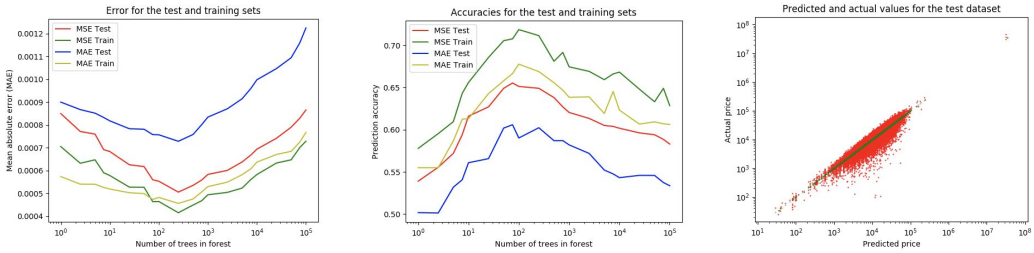
As mentioned in the proposed method, multiple hyperparameters were tuned in advance of the plotted results below to limit the size of the report. These parameters were minimum samples needed to split and create a leaf, which were set to 87, the maximum tree depth, which is set to 122, and the minimum impurity split which is set to .08. The cost complexity pruning parameter is dynamically calculated on each dataset as the model is generated by comparing the accuracies produced by varying values, graphing the most successful model. The leftmost graph depicts the mean absolute error produced by the random forest relative to the number of trees used. The graph in the center plots the accuracy within 10% of the correct price plotted against the number of trees used. The rightmost graph plots the expected prices relative to the actual prices produced by the model, with the green plot representing a 100% accuracy model.



(a) MAE, Accuracies, and predicted values scatterplot for the **'Full'** dataset



(b) MAE, Accuracies, and predicted values scatterplot for the **'Autotrader'** dataset



(c) MAE, Accuracies, and predicted values scatterplot for the **'Mechanical'** dataset

Figure 3: Qualitative visualization of the effectiveness of the optimal random forest model on each test dataset. (a): The full dataset with 14 attributes achieved 59.82% accuracy and .00044 MAE using MSE as the split criterion with 62 trees. (b): The autotrader dataset with 6 attributes achieved 72.73% accuracy and .00035 MAE using MSE with 12 trees . (c) The mechanical dataset with 9 attributes achieved 65.21% accuracy and .00051 MAE using MSE and 26 trees.

5 Conclusion

We present a novel approach to the prediction of used vehicle prices using three distinct regression models: neural networks, random forests, and k-means clustering. Given datasets of varying attribute specificity, the neural network consistently produced the most accurate results both in terms of error and prediction accuracy. The K-means clustering approach did not produce any viable prediction models. Regardless of the averaging method, or dataset at question, the optimal mean absolute error is 63%. During the testing and training procedures these models were also prone to large spikes in MAE. This suggests that clustering is an unstable solution for the problem at hand. Random Forest models proved to be promising, but in need of a further refined dataset due to its high dependency on class weights. Across all three models, the dataset containing just 6 attributes: make, model, year, odometer, condition, and price, produced the most accurate prediction results.

While our paper mainly focuses on finding the best model based on lowest mean absolute error, further exploration can refine the data set by providing more comprehensive mappings, and with increased cleaning and outlier detection, and the addition of class weights. Data from other sources could also be used to further legitimize the results and provide a bigger training set to our models. Furthermore, finely tuning hyperparameters for all models can be done once our dataset is improved.

References

[1] E. Bedford, “Retail: used car dealers Canada 2019,” *Statista*, 26-Mar-2020. [Online]. Available: https://www.statista.com/statistics/431937/retail-sales-of-used-car-dealers-in-canada/. [Accessed: 07-Apr-2020].

[2] *Statistics Canada: Canada's national statistical agency / Statistique Canada : Organisme statistique nationale du Canada*. [Online]. Available: https://www150.statcan.gc.ca/t1/tbl1/en/tv.action?pid=2310006701. [Accessed: 07-Apr-2020].

[3] A. Law, “How long do Canadians keep their cars?: Car News: Auto123,” *auto123.com*. [Online]. Available: https://www.auto123.com/en/news/how-long-do-canadians-keep-their-cars/47001/. [Accessed: 07-Apr-2020].

[4] A. Reese, “Used Cars Dataset,” *Kaggle*, 22-Mar-2020. [Online]. Available: https://www.kaggle.com/austinreese/craigslist-carstrucks-data. [Accessed: 07-Apr-2020].

[5] “Buying, selling or trading-in your vehicle? Find out what it is worth first!,” *autoTRADER.ca*. [Online]. Available: https://www.autotrader.ca/valuations/?Icoenabled=true. [Accessed: 07-Apr-2020].

[6] Canadian Black Book, 2020. [Online]. Available: https://www.canadianblackbook.com/. [Accessed: 07-Apr-2020].

[7] G. Fidan, "How Car Dealerships Really Make Money", Realcartips.com, 2020. [Online]. Available: http://www.realcartips.com/newcars/135-how-car-dealers-really-make-money.shtml. [Accessed: 07-Apr-2020].

[8] E. Gokce, "Predicting Used Car Prices with Machine Learning Techniques", Medium, 2020. [Online]. Available: https://towardsdatascience.com/predicting-used-car-prices-with-machine-learning-techniques-8a9d8313952. [Accessed: 07-Apr-2020].

[9] S. Yildirim, "Predicting Used Car Prices with Machine Learning", Medium, 2020. [Online]. Available: https://towardsdatascience.com/predicting-used-car-prices-with-machine-learning-fea53811b1ab. [Accessed: 07-Apr-2020].

[10] D. Yadav, "Categorical encoding using Label-Encoding and One-Hot-Encoder", Medium, 2019. [Online]. Available: https://towardsdatascience.com/categorical-encoding-using-label-encoding-and-one-hot-encoder-911ef77fb5bd. [Accessed: 07-Apr-2020].