CS523 - AI As An Application Final Report

Andrei Cozma and **Hunter Price**
University of Tennessee, Knoxville

Introduction

The price of a car changes significantly over time. This is due to many factors such as depreciation, the current car market, the economic environment, etc. This price fluctuation is hard to predict and can lead to significant loss to its owner. The used car market specifically, is prone to substantial changes in price due to the different avenues at which a car can be sold. A used car can be sold individual to individual, individual to dealership, dealership to individual, and dealership to dealership. Throughout all of these avenues the price is entirely up to the discretion of the seller. More often than not, people tend to be biased when they are selling or shopping for a used car. They let their emotions get in the way of making a rational decision. This behavior leads to the final price of the car being skewed towards the sellers prior beliefs of a valid price.

The first problem we attempt to solve is that the used car market is a dark market in terms of information. There is a considerable lack of information regarding the actual value of the car to be sold. This imposes a substantial task on the buyer to search for a proper price of a used car. This search requires a large amount of time and effort to make comparisons between many different sources. Often, these comparisons are not done in a sound way and rather the buyer gathers an intuition on what price "feels" correct and the makes a decision based on how they feel about the price. Used car dealerships and individual sellers are in the business of making money and are not always honest about a used car's actual worth. Thus, they try to exploit an over-stressed and over-worked potential buyer to squeeze out as much money as they can to make a large profit. Conversely, sellers may also not know a proper value to put on a car when it is time to sell.

The second problem we attempt to solve is that buyers may not know what kind of used car they are looking for is. With the large number of brands and models of used cars over the course of many years there are far too many choices for a buyer to make a rational decision.

This work can help people both purchase and sell used cars effectively by providing them with more accurate estimations and predictions that are derived from a much larger set of samples than an individual would be able to comb through by themselves. A large set of samples can more effectively produce more accurate averages and estimates for a used car's worth because the sample size can compensate for the randomness and variability in the data. This hopes to eliminate the need for the buyer and seller to perform many timely stages of search and comparisons on the prices of cars equipped with similar features.

Approach

Previous Work

There have been many different approaches in attempting to predict the worth of used cars. A random forest model was used on a similar used car dataset from kaggle.com. This work achieved an 83.62% accuracy on the testing data and a 95% accuracy on the training and validation data (Pal et al. 2019). Another paper used both a K-means model and a decision tree model to attempt to predict the price of a used car. This paper did not explicitly express the final accuracies of the models but the overall results were poor (Chandak et al.). A third paper attempted to predict the price of a used car with a support vector machine and an artificial neural network. The support vector machine model obtained an accuracy of 90.48%, and the artificial neural network model obtained an accuracy of 85.71% (Gegic et al. 2019). A final paper used a shallow artificial neural network model to predict the price of used cars. This model obtained a success rate of 91.38% (Karakoç, Celik, and Varol 2020).

Our approach is unique in that we use a deep artificial neural network as a regression model, rather than a simple linear regression model. Additionally, we were unable to locate any prior work classifying the type of used car a buyer should purchase based on desired features.

Datasets

We are utilizing publicly available data from kaggle.com. The datasets include data that has been labeled according to the manufacturer of the car, which includes Mercedes, Audi, BMW, Toyota, Ford, and Hyundai. The model, year, price, transmission type, mileage, fuel type, KPG (Kilometers per Gallon), and engine size are all included. There are 64131 rows and 9 feature columns in total.

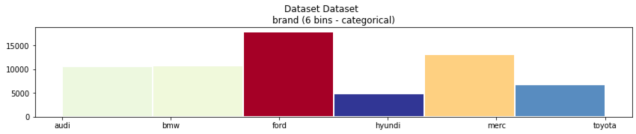

Figure 1: Distribution of Car Manufacturer

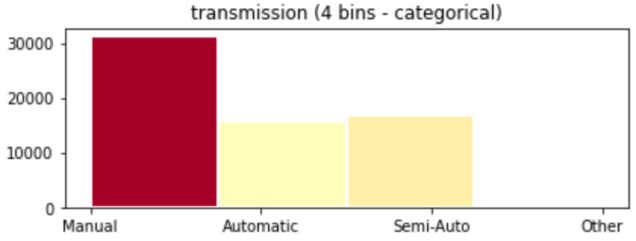

Figure 2: Distribution of Transmission

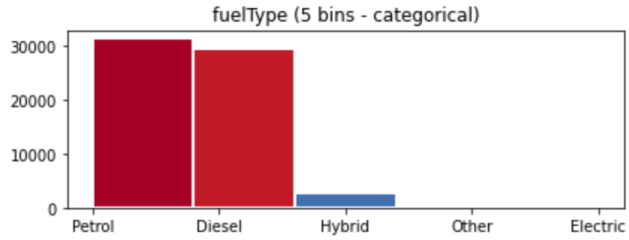

Figure 3: Distribution of Fuel Type

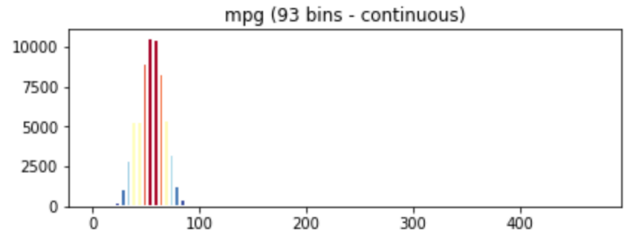

Figure 4: Distribution of Fuel Consumption

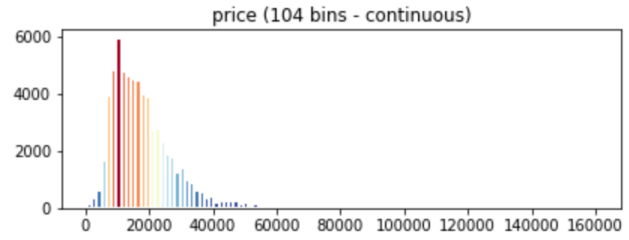

Figure 5: Distribution of Car Price

We began by analyzing the distributions of the feature columns in order to make inferences about data correlations. We examined distributions by car manufacturer as well as the combined dataset as a whole.

By the number of rows, it is clear that the Ford dataset is the largest when compared to the others. On the contrary, t he Hyundai dataset contains the fewest number of data points of any of the datasets we examined.

Surprisingly, the majority of used cars for sale were equipped with manual transmissions, nearly as many as automatic and semi-automatic vehicles combined.

Additionally, we observe that the distribution of gasoline-powered and diesel-powered vehicles is roughly equal, while hybrid and electric vehicles are the least prevalent in the data collected.

The car's fuel consumption metric follows a normal distribution with a mean of 50 kilometers per gallon, which equates to approximately 31 miles per gallon. The car's price metric is also normal, with a mean of approximately $15,000.

Implementation

To accomplish our objectives, we set out to create two distinct models that could be used in a variety of situations. To begin, a regression model would be capable of predicting the price of a car based on the vehicle's various characteristics. Second, a classification model capable of forecasting the brand and model of a car based on the customer's desired price and feature set.

We trained our neural networks using Keras and a Tensorflow backend on an Nvidia RTX3070 GPU. Both approaches make use of a sequential network that we tune in order to find the models' optimal hyper-parameters. We used the Hyperband tuning algorithm from the keras-tuner library to tune the hyperparameters.

We chose to tune the following hyperparameters: the number of hidden layers in the network, the number of neurons in each hidden layer, the activation functions for each individual layer, as well as the learning rate and kernel initializer.

The Adam optimizer was the optimization method of choice. The Adam optimizer uses an adaptive learning rate is a replacement optimization algorithm for stochastic gradient descent, and has been designed specifically for training deep neural networks.

The main differences between the classification and regression models reside in the input and output layers of the networks, which we will discuss further.

Overall, both the regression and classification models were trained on 500 maximum number of epochs. The top 10 model configurations were then saved for evaluation purposes covered in the next section.

Pre-Processing

Pre-processing of the data included encoding categorical columns using One-Hot-Encoding and standard scaling the combined dataset.

Both models had distinct target labels. The target label for the regression model was the price of each sample (car). These values have been left alone to allow the network to produce meaningful results for the user. The classification model employed a different approach. We combined the brand and model columns to form a consolidated target label. This resulted in the creation of 134 distinct target labels. Due to the loss function discussed later, these labels were then One-Hot-Encoded. We chose not to include the year in the target prediction because it would have resulted in far too many classes to predict with the given dataset. Further, we kept a training set size of 70% and 30% for testing.

Regression

We set out to build a model that can be used as a point of reference for used car buyers and sellers when making a purchasing decision. The regression model takes in the car's features and outputs an estimated price for the used car.

The network's input layer receives the pre-processed features. The output layer employs a linear activation function composed of a single neuron to compute the prediction. The rest of the network's structure is determined optimally through hyperparameter tuning. The search space for the Keras Hyperband consists of the following:

- Number of Hidden Layers: Min of 1 and Max of 10
- Hidden Layer Neurons: Min 32, Max 512, Step of 32
- Activation Functions: relu, sigmoid, tanh
- Learning Rate: 1e-2, 1e-3, 1e-4
- Kernel Initializer: random_normal, random_uniform, zeros, glorot_normal, and glorot_uniform.

We considered several metrics when determining the neural network's loss function. Mean Squared Error (MSE) and Root Mean Square Error (RMSE) both penalize large prediction errors relative to Mean Absolute Error (MAE) (MAE). MAE is generally more robust to outliers in the input data. In general, a smaller MAE, MSE, and RMSE value indicates a more accurate regression model. We chose MAE as the loss function for this model while also keeping track of the other metrics.

The results of the regression model's hyperparameter tuning indicate the optimal values for achieving the best results. The majority of the top ten best models used four hidden layers with an activation function combination of relu and sigmoid. Random uniform initialization was the optimal kernel initializer for the overall best model, while glorot uniform initialization was used for the second and third models. The Adam classifier's optimal learning rate was 0.0001. Finally, the ideal number of neurons in each hidden layer is between 200 and 400. We will explore and evaluate the best overall models in the next section.

Classification

The purpose of this model was to predict the brand and model of car that a buyer would most likely want to purchase based on their desired features and price. This model takes the characteristics of a vehicle and predicts the brand and model of the used vehicle.

Model #	R^2	MAE	MSE	RMSE
1.0	0.997988	46.253918	240898.000000	490.813599
2.0	0.999513	55.259926	58348.859375	241.555084
3.0	0.999598	44.948151	48147.414062	219.425186

Figure 6: Comparison of Top 3 Regression Models

The network's input layer accepts normalized values for transmission, fuel type, mileage, tax, miles per gallon, engine size, and price. The model's output layer employs a softmax activation function composed of 134 neurons.

We chose Categorical Cross Entropy as the loss function for this model (CCE). To use this loss function in Keras, you must first hot encode your target labels, resulting in the shape (n_samples, 134). CCE appeared to be the de facto standard for problems involving multiple classes of classification. This loss function is ideal for our problem because it applies the same penalty to all misclassifications.

The search space for the Keras Hyperband consists of the following:

- Number of Hidden Layers: Min of 5 and Max of 15
- Hidden Layer Neurons: Min 256, Max 2048, Step of 32
- Activation Functions: relu, sigmoid, tanh
- Learning Rate: 1e-4
- Kernel Initializer: glorot_normal and glorot_uniform.

The results of hyper-parameter tuning revealed a high degree of similarity among the top ten models. The number of hidden layers varied between five and nine, with nine being the optimal number. The total number of neurons varied significantly. We discovered layers with between 256 and 2048 neurons in a single layer. The most frequently used activation functions were tanh and relu.

Evaluation

Regression

The best models were individually evaluated on the testing dataset. Out of the original ten models, we were able to determine that the top three models were the best performing based on multiple different evaluation procedures described in this section. First, we evaluated the Mean Absolute Error, Mean Squared Error, and Root Mean Squared Error metrics. The results for these metrics on the top 3 models are shown in Figure 6. All three models were able to reach excellent scores for these metrics. On average, the model's predictions would be within $45 of the actual price based on the MAE metric. The best model in terms of MAE was model #3. RMSE which simply the square root of MSE, gives more weight to large errors. The best model in terms of both MSE and RMSE was also model #3.

Further, we plotted histograms to be able to compare the price distributions between the predicted price and the actual labeled price of the car. As an example, Figure 7 shows the distributions of the best model we determined previously (model #3). Here we can notice that the model was able to

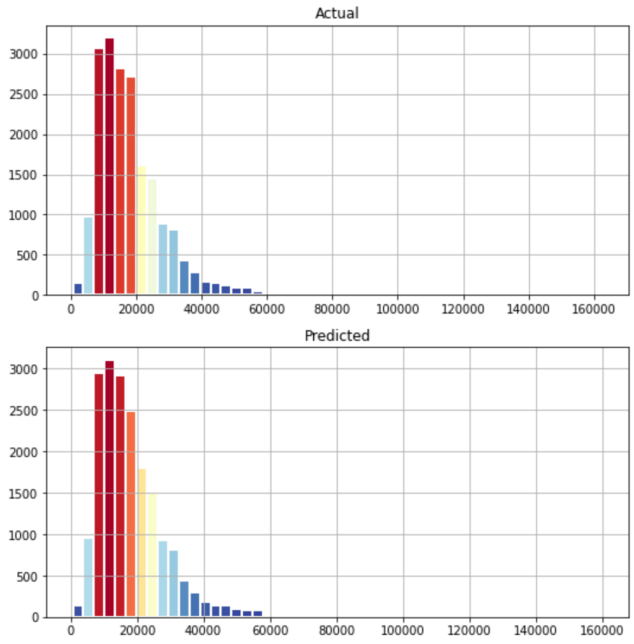

Figure 7: Regression Histograms

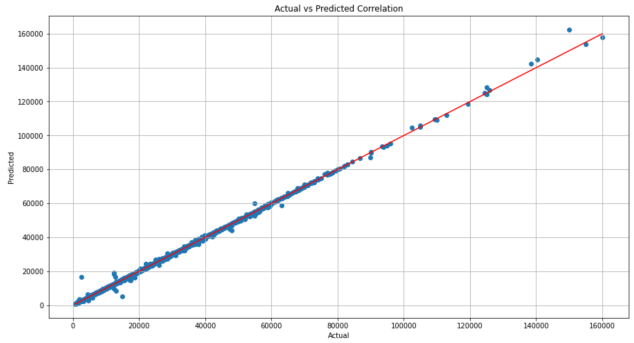

Figure 8: Regression Actual vs Predicted

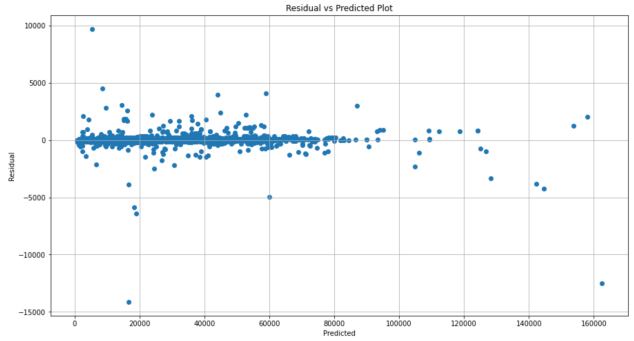

Figure 9: Regression Residual Plot

	categorical accuracy	loss	precision	recall
0	0.760551	1.136896	0.779563	0.751403
1	0.756913	0.979351	0.789818	0.740229
2	0.756913	1.150967	0.779090	0.744023

Figure 10: Comparison of Top 3 Regression Models

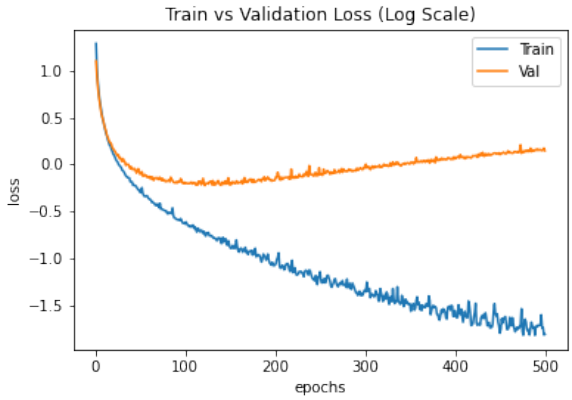

Figure 11: Training vs Validation Loss

generate predicted car prices with a very similar normal distribution as compared to the actual values. However, it is worth noting the very slight shift of the tail of the distribution in the predictions. This observation shows that the model most likely suffers from a lack of training data for more expensive cars around the $60,000 price range.

As an additional evaluation metric, we also used the Coefficient of Determination (r^2). The scatterplot between the actual and predicted data points in Figure 8 allows us to evaluate how close the data are to the fitted regression line which is shown in red. Here we can notice that while there are a few outliers, the model was able to fit the data really well overall. The best model overall has an r^2 coefficient of 0.9996.

However, we cannot use r^2 alone as it cannot determine whether the coefficient estimates and predictions are biased. Generally, a higher R-squared is desired for precise predictions. In order to check the assumptions of a regression model we also evaluate the residual plots, which are shown in Figure 9. The residual plot summarizes the difference between the actual and predicted values. We can observe yet again that cars priced above $80,000 tend to have much higher prediction errors likely due to the lower number of training samples for those price ranges.

Classification

We collected a list of the top 10 classification models given by the hyperband tuner; for each model we tested the classification loss, accuracy, precision, and recall on the testing data. The models with the top 3 accuracies are shown in Figure 10. The accuracies on the testing data was ultimately

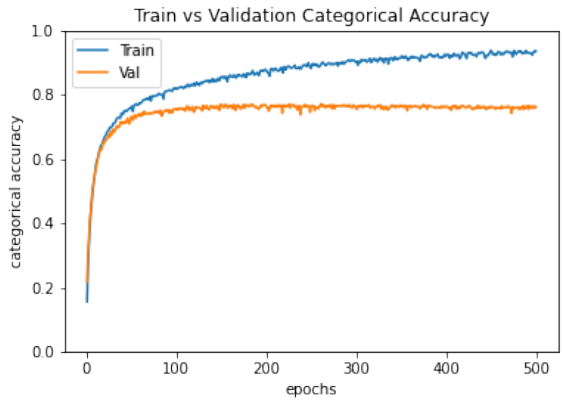

Figure 12: Training vs Validation Categorical Accuracy

poor for all models and the loss was relatively high. This reason for this result become apparent when the training and validation loss per training epoch is shown side by side in Figure 11. The training loss constantly decreases throughout training while the validation loss quickly decreases, then begins increasing. This is indicative of overfitting. Further, the categorical accuracy for the best model, shows the training accuracy increasing throughout every epoch while the validation accuracy levels off shortly after 100 epochs. This again is another sign of overfitting. These curve looks similar for all of the top models.

The best models testing accuracy ended with an overall accuracy of 76.1% accuracy, a loss of 1.14, a precision of 78.0%, and a recall of 75.1%. The model had 9 hidden layers with the following number of neurons: 768, 1920, 1792, 896, 1408, 1024, 1152, 1408, and 1408. Each layer had the following activation functions: tanh, relu, relu, relu, relu, tanh, tanh, relu, tanh, sigmoid, and softmax. The learning rate that was used was 0.0001. And the kernel initializer used was glorot_uniform.

To improve this model, we need to avoid overfitting. One way to accomplish this is to decrease the complexity of the model. We could decrease the number of layers as well as the number of neurons in each layer to achieve a more generalized model that will perform equally with both the training and testing data.

Future

This work could be pursued in a variety of different directions. To begin, we can work on improving both the regression and classification models' performance. Due to the fact that both models were built using neural networks, we can experiment with different configurations with fewer or more layers and nodes in an attempt to improve performance. Additionally, we can improve the classification model's performance and attempt to avoid overfitting the training data. We can investigate different types of models for the classifier model, such as a support vector machine or a random forest model. Additionally, we can experiment with various model fusion techniques, such as the naive Bayes model. Second, we can attempt to collect additional data. We can collect additional features or samples in order to improve our results. Third, we can build the classification model in such a way that it returns a list of the most likely cars that a user would want based on the features they desire. We can further filter this list based on the user's specifications to assist them in determining which car to purchase. Next, we can create a classification model that predicts the brand, model, and year of a product that a user may wish to purchase. Following that, we can create an application that is built around our models and allows users to enter their own data. This enables a user to quickly determine whether or not to purchase a car and how much to sell it for.

Contributions

Andrei

Andrei conducted analysis on the individual car datasets and produced composite charts for each of the six car manufacturers, as well as the combined dataset comprised of all the samples. Further analysis was conducted on the cleaned and encoded data as well as on the train-test split to generate charts. This allowed comparison of distributions between the actual values and predicted values of the models. Further research and implementations followed with the initial neural network models, the Keras Hyperband tuning approach, as well as optimizing the search space of hyperparameters for the network configurations. The main focus was the regression model and performing evaluations and comparisons between the top 10 models. Lastly, he trained both models on the GPU to be able to achieve faster results with the deep neural network approaches.

Hunter

Hunter wrote the preliminary pre-processing code to read in the multiple csv files and combine the dataframes. He also wrote the initial code to one hot encode all non-numeric columns to numeric columns that Andrei later added to. Additionally, he wrote the code to encode the classification target values in a format needed for the Keras model. Hunter did research on previous models that were used for the same and similar datasets that were used in this work and created the citations used in the proposal and final report. Hunter also wrote the code for the classification model that was later run on Andrei's GPU. Additionally, he created the model evaluation charts and graphs for the classification model.

References

Chandak, A.; Ganorkar, P.; Sharma, S.; Bagmar, A.; and Tiwari, S. Car price prediction using machine learning. 7:444–450.

Gegic, E.; Isakovic, B.; Keco, D.; Masetic, Z.; and Kevric, J. 2019. Car price prediction using machine learning techniques. *TEM Journal* 8(1):113–118.

Karakoç, M. M.; Celik, G.; and Varol, A. 2020. Car price prediction using an artificial neural network. *Eastern Anatolian Journal of Science* 6:44–48.

Pal, N.; Arora, P.; Kohli, P.; Sundararaman, D.; and Palakurthy, S. S. 2019. How Much Is My Car Worth? A Methodology for Predicting Used Cars' Prices Using Random Forest. 886:413–422. Series Title: Advances in Intelligent Systems and Computing.