Building machine learning models to estimate vehicle prices

Satya Shah
Mentored by Mr. Cammarato

Introduction

The purpose of the project was to build a machine learning model that can accurately estimate used car prices. It was hypothesized that a six layered artificial neural network (ANN) run under 1,000 epochs will be significantly more accurate at predicting used car prices than other machine learning algorithms.

Machine Learning (ML) is a branch of artificial intelligence in which data and algorithms are used to simulate human learning (Radisa et al., 2015). It has become the forefront of scientific inquiry for the last decade due to its variety of applications from predicting building energy consumption (Radisa et al., 2015) to aiding in chronic disease diagnosis (Battineni et al., 2020). Researchers must apply different models to find which works best to tackle today's challenges (Diller et al., 2019). Such models can assist retail traders by accurately estimating a car's value, allowing for greater transparency in automotive trading. A variety of factors such as miles driven, year, make, model, and components contribute to a car's price. Multiple ML algorithms could be employed to determine a vehicle's market value.

Materials and Methods

The data was collected from CarMax using an API key, consisting of 25,000 cars including their components and respective prices. The set was randomly split into training (80%) and testing data (20%). TensorFlow was used to build each algorithm in Google Collab.

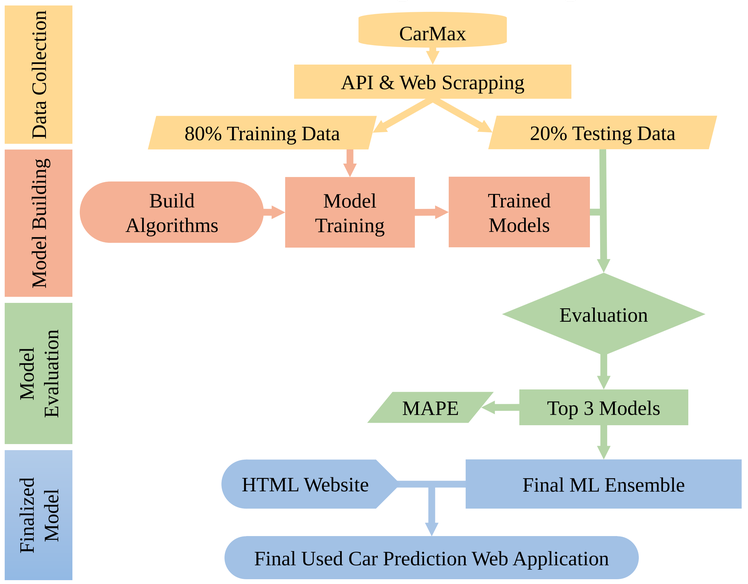


Figure 1 (above): Flowchart outlining the project's methods which show initial data collection, building/evaluation of the models, and the final HTML website deployment.

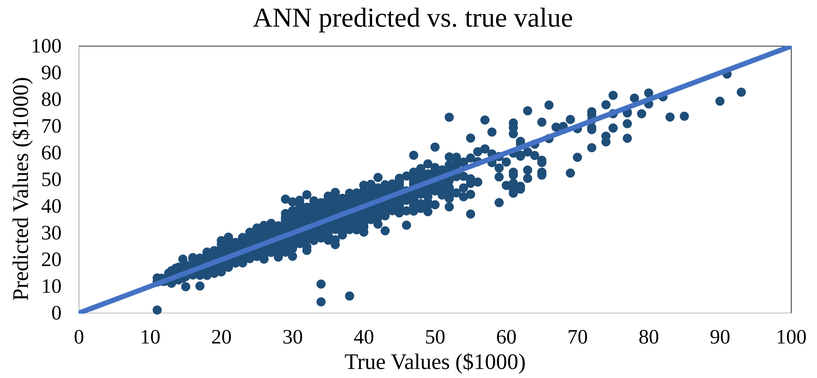
Materials and Methods (cont.)

Each algorithm was fit to the training data and validated by evaluating its prediction accuracy of the testing data. The mean absolute percent error (MAPE) between predicted and actual prices was calculated for each algorithm. An ensemble of the models with a significantly high predictive accuracy was created and their hyperparameters were optimized. Finally, a website employing HTML, CSS, and PHP (Table 1) was created. The complete model was then uploaded to the site for public usage (Figure 1).

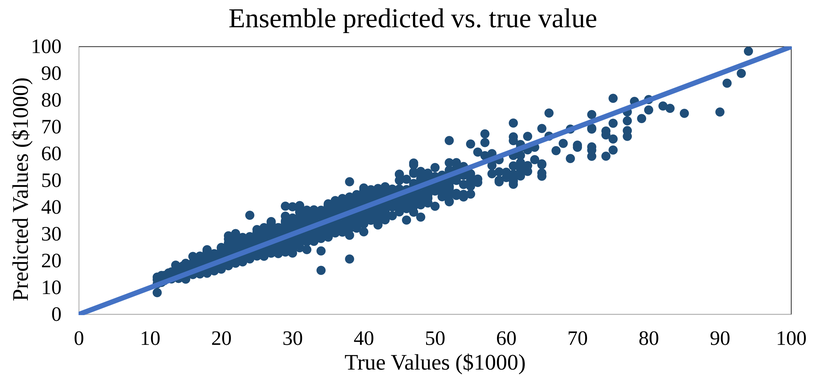
Editors	Libraries	Languages
Google Collab	Tensor-Flow	Python
Repl.it	Selenium	HTML
PyCharm	Pandas	CSS
	Requests	JavaScript
	Matplotlib	PHP

Table 1 (above): The summary of editors utilized to write the code, installed libraries, and different languages used to build the algorithms and website.

Results



Graph 1 (above): Validation for the effectiveness of the individual ANN ensemble. The blue line represents optimal prediction values. The model had a MAPE of 6.38%.



Graph 2 (above): Validation for the accuracy of the ANN and random forest (RF) ensemble. The blue line represents optimal prediction values. The model had a MAPE of 5.86%.

Results (cont.)

Table 2 shows the MAPE of each model after multiple trials. Each trial consisted of changes in code, adjustment of hyperparameters, and/or addition of new data parameters. Each model was built and trained on the same group of training and testing data for consistent results. The best model with a 5.86% MAPE was the ANN/RF ensemble (Graph 2), each of which had isolated MAPE's of 6.38% (Graph 1) and 6.77% respectively. The ensemble model was created using simple averaging.

Model Type	Best Trial	Trial #1	Trial #2	Trial #3	Trial #4
Ensemble	5.86	8.91	5.86	-	-
ANN	6.38	14.56	8.54	7.78	6.38
RF	6.77	40.90	33.49	8.75	6.77
KNN	24.34	41.50	40.32	24.34	25.60

Table 2 (above): The improvement of the top four models, artificial neural network (ANN), *k*-nearest neighbor (KNN), random forest (RF), and ANN/RF Ensemble over multiple trials and the MAPE recorded for each trial, in percent error. The best model was the ANN/RF ensemble with a 5.86% MAPE.

Conclusion

The project's objective was met. An ensemble model greatly outperformed an individual ANN model. A robust model with a mean absolute percent error of 5.86% was constructed by integrating an ANN and an RF model using simple averaging. Consistently, ANNs were the most accurate individual models. However, the RF algorithms improved the most over the course of the project. The KNN model was the least accurate model.

In the future, techniques such as support vector machines (SVMs) could be utilized in the prediction process. Second, the ensemble could be constructed using weighted averages, or it could be used as an input to a more encompassing model to see if this reduces the MAPE.

References

Battineni, G., Sagaro, G. G., Chinatalapudi, N., & Amenta, F. (2020). Applications of machine learning predictive models in the chronic disease diagnosis. *Journal of Personalized Medicine*, *10*(2), 21. https://doi.org/10.3390/jpm10020021

Diller, G. -.P, Kempny, A., Babu-Narayan, S. V., Henrichs, M., Brida, M., Uebing, A., Lammers, A. E., Baumgartner, H., Li, W., Wort, S. J., Dimopoulos, K., & Gatzoulis, M. A. (2019). Machine learning algorithms estimating prognosis and guiding therapy in adult congenital heart disease: data from a single tertiary centre including 10,019 patients. *European Heart Journal*, *40*(13), 1069–1077. https://doi.org/10.1093/eurheartj/ehy915

Radisa, Z. J., Aleksandra, A. S., & Branislav, D. Z. (2015). Ensemble of various neural networks for prediction of heating energy consumption. *Energy and Buildings*, *94*, 189–199. https://doi.org/10.1016/j.enbuild.2015.02.052