Predicting Car Resale Price using Machine Learning

Abstract

Used car sales make up the majority of vehicular sales in the United States of America due to the affordability of these cars compared to new vehicles. By looking at certain features of a vehicle, such as its condition, title status, odometer reading and many more, we are interested in estimating the resale value of used cars. Using data aggregated from Craigslist, an online advertisement website in America, we can formulate many insights into the used car market in the USA.

1. Introduction

In recent years, the price of used cars has skyrocketed due to the onslaught of the COVID-19 pandemic and the reluctance and inability of manufacturers to continue production of silicon chips for their critical use in the manufacturing of new vehicles. As a result, there has been a shortage in the new car market, resulting in many consumers turning to the used car market for their next vehicle purchase, resulting in increased market demand and higher prices across the board for used cars. Resulting from this recent spike in used car prices, there is an incredibly strong desire amongst consumers for a return to normalcy - a market that does not so heavily favor the seller. With this desire, many consumers are desperately looking for a way to determine what the true value of used cars are in this adjusting market. With this in mind, this project aims to accurately predict the selling price of a used car, given its model year, make, model, condition, title status, size, odometer reading, and a plethora of other descriptive features.

2. Dataset Exploration

The dataset used for this prediction task is from Kaggle. It contains information on Craigslist listings for vehicles, which are primarily used cars with a couple exceptions containing dealership advertisements. This data is scraped from Craigslist websites in every county in America and is regularly updated as more listings are available. Particularly, during the use of this dataset it contained 426,880 listings from Craigslist along with 26 columns for the listings' features. The features contained in this dataset include: id, url, region, price, year, manufacturer, odometer reading, model, condition, number of cylinders, title status, fuel type used, type of drivetrain (FWD, RWD, 4WD, AWD), size of vehicle, type of vehicle, state, paint color, description, posting date, as well as more features that are less descriptive for our prediction purposes.

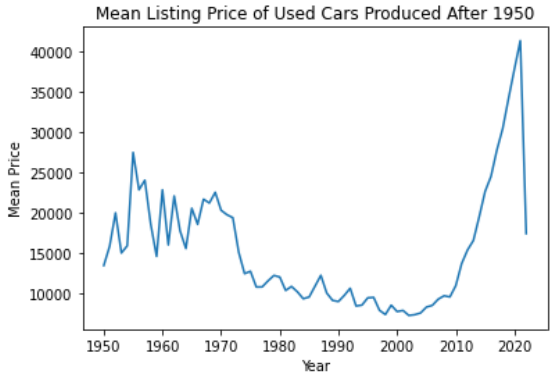

Figure 1: Average Listing Price per Car Manufacturing Year

In our motivation for this project, we talked about the rising prices of vehicles, especially during the COVID-19 pandemic and the ensuing lockdown. We can see in Figure 1 how the price of vehicles has been affected over time. We see that much older cars are selling for more than slightly older cars (1950s era vs 1990s era). This is most likely due to collectors selling older cars, like for example, the classic '69 Ford Mustang. We also see an exponential increase after about the year 2005, which is super interesting, and we intend for our model to take this into account as it does training and predictions.

Looking through the data, we at first realized that many of these columns contain a significant amount of noise and should not be related too closely to the price of the listing. Some of these columns might overfit our data into relying on certain meaningless patterns. We ended up dropping columns that did not feel relevant such as: VIN, ID, Latitude, Longitude, and Image URL.

Next, while investigating the number of null values in our dataset, we saw how abundant these nulls were in certain columns. From figure 2, we can see that the features in which they were the most apparent were in columns of county,

	num_null
county	378659
size	271951
cylinders	154532
VIN	144948
condition	142533
drive	114412
paint_color	108018
type	80496
manufacturer	14795
title_status	6551
model	4305
lat	3446
long	3446
fuel	2587
odometer	2071
transmission	1808
year	1162
description	61
image_url	59
posting_date	59
url	0
price	0
state	0
region_url	0
region	0
id	0

Figure 2: Count of Null Values in Dataset

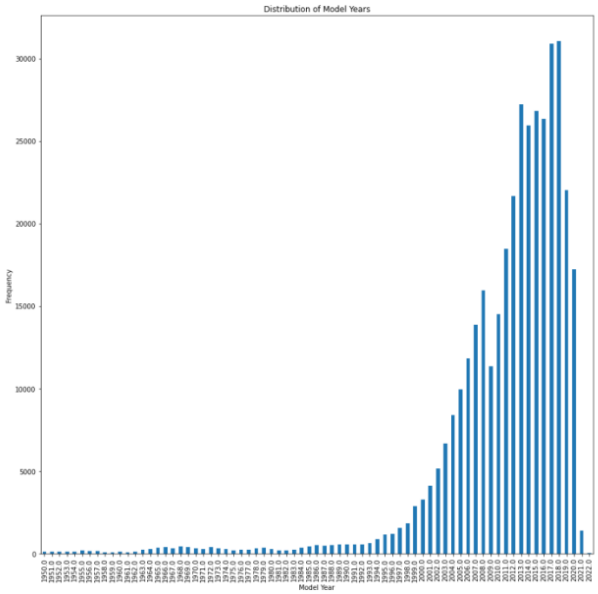

Figure 3: Distribution of Car Years

size, cylinders, vin, and condition. We wanted to use size, cylinders and condition in our predictions but we don't deem county or vin to be super important for our model as it most likely will overfit. In fact, it is certainly the case that, upon inspection, the county column is entirely filled with null values. We looked into imputing some of these columns' values, but as they were not heavily correlated with price - we decided to simply drop some of these columns as features.

Looking at the data in terms of the year of each car, we do see that almost all are used cars from previous years. Moreso, we see that most are represented from around year 1995 upto year 2020. With many of these cars falling in this range, we are expected to get more of an idea of the "typical used car", not cars such as old collectors items. Our dataset is much more represented in the typical used car way.

Furthermore, we explored each individual column and how it relates to the price of the listing. Before that, we had to look at the price column used and distinguish if these were valid listings or not. The nature is that on Craigslist, many listings are marked down to zero or one dollars to appear higher up on the results board when sorting by price, but it isn't exactly representative of the true price the seller wants. Looking at figure 4, we can see that indeed many listings fall out of the normal range that we would typically see for used car sales - there are many results falling less than one hundred dollars and results that are way above hundreds of thousands of dollars. In order to preserve our models' integrity by generating good results, we limited these listings to be at least over one thousand dollars (to make up for these low listing prices that aren't real) but less than one hundred thousand dollars (to deal with possible typos, extra zeros). Now in figure 5, we can see these prices are much more realistic and have a much better distribution than before - we will impose this threshold as we move further.

After this, we look at the descriptive data statistics. First, we look at the numerical columns and compute the normal statistics such as mean, standard deviation, quantiles, min,

Figure 4: All Price EDA

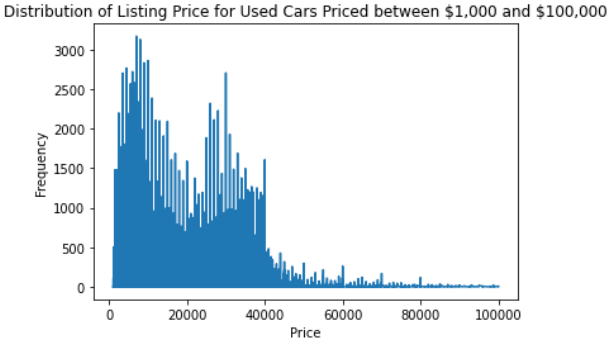

Figure 5: Limited Price EDA

	id	price	year	odometer	county	lat	long
count	3.786590e+05	378659.000000	377497.000000	3.765880e+05	0.0	375213.000000	375213.000000
mean	7.311456e+09	19416.465524	2011.001144	9.809761e+04	NaN	38.505508	-94.285332
std	4.474049e+06	14320.108898	9.567361	1.841765e+05	NaN	5.838847	18.063606
min	7.207408e+09	1002.000000	1900.000000	0.000000e+00	NaN	-84.122245	-159.719900
25%	7.308070e+09	7995.000000	2008.000000	3.833675e+04	NaN	34.708828	-110.891800
50%	7.312553e+09	15991.000000	2013.000000	8.763400e+04	NaN	39.193119	-87.994400
75%	7.315237e+09	27990.000000	2017.000000	1.360000e+05	NaN	42.350000	-80.828819
max	7.317101e+09	99999.000000	2022.000000	1.000000e+07	NaN	82.390818	167.629911

Figure 6: Descriptive Summary Stats of Numerical Columns

	price	year	manufacturer	model	condition	cylinders	fuel	odometer	title_status	transmission	size	drive	type	paint_color	state
price	1.000000	0.262123	-0.061816	0.067151	0.011955	0.260596	-0.352256	-0.165322	-0.025603	-0.025224	-0.054421	-0.158280	0.090991	0.054775	-0.014029

Figure 7: Correlation Matrix

max and more as shown in figure 6. This gives us an idea about how our target variable, price, is spread across the data along with some features we know are going to be useful such as year and odometer. It is interesting to see that we have cars ranging from year 1900 to the present in 2022. This range will be amended to only encapsulate cars produced between 1950 and 2021, to prevent the small number of vehicles (some sold as collector items) from the excluded years significantly influencing the model's predictions.

Finally, after encoding our columns using a label encoder in order to make these categorical columns interpreted numerically, we take a look at how each of the features is correlated with price. Correlation can give us a good idea of what features are truly important in our model and some we might be able to skip out on. Looking at figure 7, we see that year and price are very much positively correlated at 0.26, while things like paint color or state have not much of a relationship with price.

At the end of our EDA, we were left with less rows and less columns than in the start as we filtered certain conditions to be true in order to only have feasible, realistically priced listings in our dataset. We ended up with 378,659 listings that will be used in our model for both training and testing. This is plenty big enough to train and get good, reliable predictions on.

3. Predictive Task

The core task of our project, as described in previous sections, is the prediction of a used car's resale value based on the plethora of features mentioned in our dataset section.

In a general sense, features were constructed in the following way: We ensured that our categorical features were able to be used by our models for predictions by label encoding them. Our numerical features, however, did not require much adaptation (besides range limitations, which are described in the previous section, based on our EDA to ensure proper predictive power) and were used directly to make predictions. Further information and finer details regarding feature implementation and usage/validity of their use is explained in greater detail below.

To evaluate the performance of our models, we decided to use R^2, which measures the strength of correlation between the models' predicted values with the actual observed values in the dataset, as our performance metric over the traditional metric of MSE due to the fact that MSE is more prone to influence of non-scaled price values, as is the case in our task, and all of our tested models for this project are either regression models or simple (linear/constant) predictions, to which the evaluation of our predictions using this metric is especially relevant.

As baselines for our predictive task, we used a simple linear regression model, along with a simple constant prediction model that predicts the mean selling price of that make and model in the training set, or the global mean selling price of the training set, if the specific make and model in the test set is not present in the training set.

The validity of the models' predictions were verified in two ways:

First, via a simple plotting of the predicted results vs. the actual observed values, we verified that the spread of the predicted data is more similar to the spread of the observed data when we obtain a larger R^2 value.

The other way in which we achieved this was that we selected the top four performing models and ran a cross validation suite on these select models, alongside a tuning of hyperparameters- varying batch size, number of layers, layer depth, number of epochs, and using a learning rate scheduler based on validation loss for a simple MLP (with linear activation) prediction model.

A third possible way, to ensure the validity of our models' predictions, which was not entirely explored, would have been to calculate the value of R^2 for our training set and check that it is relatively close to 1.0 and compare it to our test R^2 value to determine the best model.

In order to gauge a baseline of each of our models, our feature set was directly derived from the original columns of the dataset. We used the following columns to act as features for our prediction task:

1. Year - We decided to use the year column, which represents the model year of a vehicle, as a determining factor in price predictions due to its potential role in in signalling the age of a vehicle. As shown in figure 1 of our exploratory data analysis, the older a vehicle is, the cheaper it tends to be, with the exception of especially old vehicles that may be regarded as collector items, as well as cars with a model year of 2022, which may be due to the sheer lack of data of these newest cars in the dataset, as can be seen in figure 3 of our dataset (as such, we only considered cars from model years 1950 - 2021 to train and evaluate our models). Furthermore, the corresponding year in which a posting was created (derived from the posting_date column) can play a role in reflecting price increases in inflation year over year, as well as its potential in reporting economic changes as a function of time.
2. Manufacturer - The inclusion of the manufacturer column (i.e. the make of a vehicle) in our feature set allows us to use the innate characteristics of a brand's name and production (build) quality of vehicles as a factor that influences the price of a used vehicle.
3. Model - Vehicle model can be used as a price predictor

due to a wide variety of differences in the rarity, historical value, inclusion of special packages (e.x. sport vs. limited vs. base), etc. of a specific vehicle model.

4. Make + Model - The concatenation of both make and model increases specificity of the feature for training. It is used in addition to their separated counterparts to retain the innate characteristics of these features (some makes are worth naturally more than others, due to their quality and brand name, while some models are worth naturally more than others, due to their rarity).
5. Condition - A vehicle's condition (new vs. like new vs. excellent vs. good vs. fair vs. salvage), when label encoded, can be utilized as a strong predictor of price - based on the correlation matrix produced during our EDA, a better condition is expected to correlate with a higher price, and a worse condition is expected to correlate with a lower price.
6. Cylinders - When label encoded, the number of cylinders is an indicator of vehicle (engine) performance, which is positively correlated with price (figure 7) - the more cylinders a vehicle's engine has usually indicates the presence of a stronger and more performant engine, which sellers/customers may find great value in, resulting in higher prices.
7. Fuel - The type of fuel a vehicle runs on (e.x. electric vs. flex vs. diesel vs. gasoline) has a strong correlation with the price of the vehicle. Electric vehicles tend to be more expensive since they are a relatively new technology, while cars that run on diesel tend to be less expensive due to their nature of emitting higher levels of pollution, and the overall expensive nature of diesel per gallon, compared to standard gasoline. A vehicle than can take in E85, or flex fuel, is usually mid-range in terms of price, simply because it has a shorter range, but is less costly per gallon, compared to other types of fuel.
8. Odometer - The odometer reading indicates how long a vehicle has been driven for, in miles, which can be an indicator of the quality and expected lifespan of this vehicle. This measure also suggests that a vehicle is newer if the value of the odometer reading is low, which relates to vehicle condition and has a strong correlation with price.
9. Title Status - When label encoded, a vehicle's title can be used as a strong categorical predictor of price, as it describes the status of its accident and damage history. A salvage or rebuilt car (i.e. a car that has been deemed by insurance to be a total loss), for example, is expected to have a lower selling price, compared to a vehicle that has a clean title (has never been in an accident reported to insurance).
10. Transmission - When label encoded, the transmission type of a vehicle is indicative of its popularity amongst consumers. Consumers, especially in the United States, tend to prefer the ease of use of an automatic car over a manual car, which reflects in their pricing. Manual cars tend to be cheaper (and usually older, from a time when manual cars were more abundant on the road) than automatic cars.
11. Size - When label encoded, the size of a vehicle reflects upon its production cost, storage capacity, and luxury, which is often reflected in the price passed onto the consumer. An full size SUV or truck tends to be more expensive than a midsize sedan, or a compact car.
12. Type - Similar to size, the type of the vehicle that is being sold, when label encoded, is used to more generally reflect on the storage capacity and luxury of that vehicle. Likewise, an SUV will generally tend to be more expensive than a sedan, ceteris paribus.
13. Drivetrain - When label encoded, the drivetrain of a vehicle indicates its suitability for different terrain and inclement weather. Offroading vehicles and 4wd vehicles tend to be suitable for all types of terrain and weather since they deliver power to all four vehicles simultaneously and retain grip, while 2wd vehicles tend to be suitable for dry, paved roads. As such, 4wd drivetrains tend to be used more in SUVs and trucks; whereas, 2wd drivetrains tend to be used in sedans and compact cars. As a result, for a similar reason as vehicle type, vehicles with 4wd drivetrains tend to be more expensive than those with 2wd drivetrains.
14. Paint Color (Label Encoded) - Sellers and buyers alike may prefer one color over another, popular colors, limited colors, or colors that just overall indicate luxury may be more expensive - e.x. chromaflair, chrome paint, gold, thermochromic paint, etc. may be part of a more expensive package with a higher selling price compared to a vehicle that is simply painted black or white, which are in greater supply (induced demand).
15. State - Markets within states may have their own price ranges for vehicles due to the level of competition (availability of vehicles), tax rates, and overall cost of living, making the state in which a vehicle is listed, when label encoded, a good, but possibly a slightly weak, predictor of price.

Furthermore, we also employed feature engineering to make a separate set of predictions in a simple linear model. To do this, we constructed, for each data point in the training set, a vector containing:

1. The number of words in a description - the more words in the description, the more likely a seller may need to explain why their car is worth buying (indicating desperation), which may be correlated with a lower selling price.
2. The average price of a certain make and model if it was included in the training set, or the global mean training price if that specific make and model was not present in our training set, which was also the method used to make predictions in one of our baseline models.
3. Our standard set of features, as described above.

However, this set of added features did not significantly improve results over our standard set of features, and thus, was excluded from our analysis of model results.

Features that were not used for this predictive task include posting id, url, VIN, image_url, and region_url, since they

were not deemed to be accurate predictors of price, based on our EDA, as described in the Dataset section of this report.

4. Model

In order to get the best results of our prediction, we initially trained on 13 different models: Linear Regression, Lasso Regression, Ridge Regression, a simple mean predictor, Decision Tree Regressor, Linear SVR, Nu SVR, SVR, Ada Boost Regressor, Gradient Boosting Regressor, Random Forest Regressor, Extra Trees Regressor, and a Multilayer Perceptron with a linear activation function, a learning rate scheduler based on validation loss, and manual hyperparameter tuning (of the batch size, number of epochs, layer number, and layer depth), as described above. To train and test our model, we split the data for 75% of our data was training data and 25% of our data was test data. We used shuffle split to split our training and testing data to eliminate potential biases in the ordering of the dataset.

From the results of our 13 initial tests and evaluations, we found that the Decision Tree Regressor, Gradient Boosting Regressor, Random Forest Regressor, and Extra Trees Regressor outperformed the rest of the models significantly, being the only models to consistently result in an R^2 score of above 0.7. We decided the class of models that performed the best were models that utilized gradient boosting and decision trees.

Aside from the top 4 considered models, the rest of the models did not display signs of high performance. Linear regression had an R^2 score of 0.28744, while the rest of the seven models had R^2 scores of below zero, or did not significantly improve upon our baseline R^2 value obtained via linear regression. Some possible reasons for why these models performed poorly are that they were too complex and overfit to the training data (as was most likely the case of our MLP network), they were too overly simplified to make good predictions (as was most likely the case for our simple mean predictor, which simply predicted the average training price of a vehicle based off of its make and model), or they simply had a non-linear structure, for which R^2 was not the most suitable metric, and pursuing further investigation into these models proved to be unfruitful. As a result, these models were not considered, and the signal that these models did not use gradient boosting or decision trees helped with the decision to further explore the better performing models.

Hence, we found two more models to evaluate: XGBoost Regressor and LightGBM Regressor. XGBoost (extreme gradient boost) is a method that improves upon gradient boost by improving regularization and speeding up training time. LightGBM (light gradient boosting machine) improved upon a gradient boosting decision tree algorithm by growing trees vertically instead of growing trees horizontally.

To gain better insight on which model performs best in our dataset, we performed cross validation on the Random Forest Generator, Extra Trees Regressor, XGBoost Regressor, and LightGBM Regressor. We used a cross validation k value of 5 for all four models, and tuned the set of hyperparameters for each of the models as shown in table 1. Our cross validation method was stratified k-fold, just like when testing without cross validation, we wanted to train on as balanced data as we possibly could.

As reflected in the results shown in table 2, XGBoost performed the best after our cross validation, scoring an R^2 score of 0.90144. The hyper-parameters used to tune this model were to use a subsample of 1, large n_estimators (we used 1,500), a maximum depth of 6-7, learning rate of 0.3, and column sampling by tree of 0.7.

As XGBoost offers more advanced regularization techniques, overfitting was not a large concern in our training. XGBoost has not shown signs of difficulty when training, and has overall been a solid choice for this task. Although there has been little literature showing the importance of tuning XGBoost's hyper-parameters for significant performance improvements, the hyper-parameters chosen for this task have shown to be successful over the default parameter choices supplied by scikitlearn [CG16]. On the other hand, literature showed that LightGBM had overfitting behavior on smaller datasets, which led us to tune our hyperparameters conservatively [Ke+17]. The rest of the 13 models were not taken greatly into consideration of overfitting, as we discovered outstanding results from the XGBoost and LightGBM models.

We were able to confidently decide on XGBoost as our model for this prediction task because our initial set of models signaled higher performance with gradient boosting and decision trees. These signals were significant as the models that used those methods consistently outperformed other models by a large margin, enough to solidify a top 4 from our first 13 models.

5. Literature

There has been a lot of applications of machine learning and data mining to specific use cases in a variety of fields, in this case, prediction of prices given a plethora of features. Typically, a large portion of researchers attempt to first extract meaningful data, and perform data cleaning. In doing this, and preparing data in other miscellaneous ways, researchers can deploy models using it.

The dataset we used already existed on Kaggle and was fairly complete, meaning that we weren't required to combine a few different datasets and have a "master" dataset. Our usage of the dataset varied due to the modifications we made upon it. As described in the "Dataset Exploration" section, we modified it by dropping columns that were irrelevant in our analysis, and in doing so, the result was a dataset with fewer rows and columns. This left us with a reasonable amount of data to work with, which was around 380,000 listings/rows that we could feed to our model.

There has been quite a large amount of research done into this topic, of car resale, in the past involving machine learning. A few datasets that have been researched in the past include the resale value of different vehicles in various different regions. A few examples we looked at included websites from the UAE [AlS21] and Bosnia and

Cross-Validated Model	R^2 Hyper-Parameters
Random Forest Regressor	n_estimators, max_features, min_samples_leaf, bootstrap
Extra Trees Regressor	n_estimators, max_features, min_samples_leaf, bootstrap
XGBoost Regressor	n_estimators, max_depth, eta, subsample, colsample_bytree
LGBM Regressor	num_iterations, max_depth, num_leaves, bagging_fraction, feature_fraction

Table 1: Hyper-parameters used for cross-validation of models

Herzegovina [Geg+19] a German e-commerce website [Mon+18], and an Indian resale service [SK20].

These datasets were homogeneous to the dataset we used, in the sense that a lot of the features used were identical or very similar to the ones our data contained. The approach behind the analyses of these datasets is complicated, as this prediction task can sometimes be a complex problem due to the large amount of features. Research shows that people try to utilize generic, fairly trivial machine learning techniques. This is done in order to create a baseline model, that is, a model that provides results using a trivial prediction method. This is clear in papers such as a comparative study on regression based prediction [Mon+18], where the authors use linear regression and multiple linear regression as a baseline.

Building off of those baseline models, researchers then employ state-of-the-art methods, that is, more complex models such as gradient boosted regression trees, and Forests to improve their predictions, as is evident in many papers.

XGBoost is a scalable boosting system introduced to approximate tree learning, in order to improve results of machine learning algorithms. We noted that this boosting system may be of benefit to us with respect to improving our R^2 score [CG16]. Moreover, LightGBM was another machine learning algorithm we studied; it achieves state-of-the-art performance by randomly dropping data instances with smaller gradients. We suspected that this may benefit us for our predictive task compared to more trivial machine learning algorithms [Ke+17].

Based on the existing work on this topic, we believe our work and the existing work has fairly similar conclusions. We believe more advanced, or complex models like Random Forest Regressor(s), XGBoost Regressor, or LightGBM Regressor are a better approach to the task at hand than simpler, more trivial models such as Linear Regression.

6. Results/Conclusion

In this report we used various regression models, and other machine learning algorithms to tackle the problem of predicting car resale values, using the "Used Cars Dataset" from Kaggle. These different regression models use characteristics such as year, manufacturer, mileage, etc., to use as features to predict a car's resale value given its features. Our baseline for this project as using a simple mean predictor to predict car resale value, which performed quite poorly, giving us R^2 = 0.5797. While experimenting with various different models, we noticed that many produce an R^2 score below zero. This introduces the idea that particular models such as Lasso and Ridge regression, as well as many of the support vector regression models fit our data extremely poorly, even worse than a horizontal line. Through our experiments with various different regressive machine learning models, we found that the XGBoost and LightGBM were the two modeling families that were high performing, and can predict the resale value of a car fairly accurately. Amongst all the models, we found that the XGBoost regressor has the best performance compared to a simple mean predictor linear regression, random forest regressor(s), and LightGBM regressor. After fine tuning the models and cross validating, we obtained an R^2 = 0.9014450888493033, which was our best performing model.

Model [Regressors]	R^2 Score
Linear Regression	0.28743996917730896
Lasso Regression	-1.6086729833458806
Ridge Regression	-1.6069951319201117
Simple Mean Predictor	0.5797508843436932
MLP with LR Scheduler and HT	0.4088143604146388
Decision Tree	0.7787174656400603
Linear SVR	-0.08715362777881563
Nu SVR	-79.69506193729143
SVR	-50.32750819925044
Ada Boost	-1.5589637932299687
Gradient Boosting	0.7135044883412069
Random Forest	0.8591335608052795
Extra Trees	0.8677954391429455
XGBoost	0.8570208361908647
LGBM	0.8153352011957764
Cross-Validated Random Forest	0.8608358254945203
Cross-Validated Extra Trees	0.8602015606364788
Cross-Validated XGBoost	0.9014450888493033
Cross-Validated LGBM	0.8910888852721713

Table 2: R^2 Scores of evaluated models

From the analysis of model parameters (weights) for our linear regression model and the final layer of our MLP network, it can be seen that certain features, such as year and condition have a high associated weight attributed to them, meaning that they are good predictors of used car prices, while other features, such as paint color and state have a lower weight associated with them, meaning that they don't make as good predictions on used car prices.

With our final and best performing model being XGBoost, we used cross validation to optimize the hyperparameters of the model. Through this method of optimization, we obtained a value of one for the optimal subsample parameter value, meaning the data will be subsampled fully once per iteration. This limits the possibility of batching leading to overfitting in our model. The optimal number of estimators, which represents the total number of trees our model is creating, was determined to be 1500, which was the maximum amount specified in our cross validation setup. This means that there is the possibility that a higher value of this parameter may have been more optimal for our use case; however, this would extend training time significantly and may lead to overfitting. Our model's eta parameter, which determines how quickly the step size of our model shrinks (for gradient updates) was optimally determined to be 0.3. This means that our step size shrinks by 30 percent each update iteration, resulting in infinitely finer improvements to the gradient until convergence. Penultimately, colsample_bytree, which is the fraction of features that are randomly selected/sampled to be used for training in each tree, was optimally determined to be set to 0.7, meaning 70 percent of features were randomly selected each iteration for tree construction, resulting in a reduced chance of overfitting to a specific set of features, as each tree should be slightly different. Finally, a finer parameter that was tuned and applies to each tree individually was max_depth, which was optimally set to 6 after running cross validation. This value represents how many levels deep each of our trees can maximally fan out to. This value is in a Goldilocks position in that it isn't too small that it creates limited predictions on a limited set of feature characteristics, but it is also not too big, where there could be a higher possibility of overfitting to the finer characteristics of each feature in the training set.

References

[CG16] Tianqi Chen and Carlos Guestrin. "Xgboost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016, pp. 785–794. URL: https://www.kdd.org/kdd2016/papers/files/rfp0697-chenAemb.pdf.

[Ke+17] Guolin Ke et al. "Lightgbm: A Highly Efficient Gradient Boosting Decision Tree". In: *Advances in neural information processing systems* 30 (2017). URL: https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

[Mon+18] Nitis Monburinon et al. "Prediction of Prices for Used Car by using Regression Models". In: *2018 5th International Conference on Business and Industrial Research (ICBIR)*. IEEE. 2018, pp. 115–119. URL: https://ieeexplore.ieee.org/abstract/document/8391177.

[Geg+19] Enis Gegic et al. "Car Price Prediction using Machine Learning Techniques". In: *TEM Journal* 8.1 (2019), p. 113. URL: https://www.temjournal.com/content/81/TEMJournalFebruary2019_113_118.pdf.

[SK20] K Samruddhi and R Ashok Kumar. "Used Car Price Prediction using K-Nearest Neighbor Based Model". In: *Int. J. Innov. Res. Appl. Sci. Eng.(IJIRASE)* 4 (2020), pp. 629–632. URL: https://web.archive.org/web/20210319072446id_/https://www.ijirase.com/assets/paper/issue_1/volume_4/V4-Issue-2-629-632.pdf.

[AlS21] Abdulla AlShared. "Used Cars Price Prediction and Valuation using Data Mining Techniques". In: (2021). URL: https://scholarworks.rit.edu/cgi/viewcontent.cgi?article=12220&context=theses.