US Second-hand Cars Price Prediction Based On Multiple Features

Maoyi Liao & Ting Wu

Summary of questions and results:

1. How is the car price affected by its mileage? Is there a relationship between them?

   Yes, there is a negative relationship between them: the car with larger mileage tends to be sold at a lower price.

2. Are some variables more important than others in determining the price of the car? What are they?

   Yes, we found that numerical variables such as price, model year, and mileage, as well as categorical variables such as manufacturer, model, and so on, have larger effects and strong correlations with price.

3. How accurately can we predict the price of second-hand cars?

   We created three models using machine learning, and the best model is the ridge regression, which has the smallest mean squared error of nearly 0.418.

Motivation

Buying and selling a car is something that many adults will have to do at some point in their lives. In general, people have two choices: buy a new car or a used car. A new car's price is usually available on the website, but the price of a used car can vary greatly depending on the car itself. As a result, we want to create a used-car price prediction model to assist traders in the used market in estimating the market's approximate price based on the car's characteristics such as brand, mileage, color, and other information.

Knowing what variables are more important than others can help us choose the most relevant features to train the prediction model. Knowing the relationship between mileage and the price of the car can give us a better sense of what price to expect under certain mileage. Knowing the accuracy of the model can help us determine whether we could rely on the model for future use.

Dataset

Our datasets are from Kaggle, and one was scraped from auctionexport.com, a website that sells and repo cars via auction. Each row in the datasets represents a single car, complete with its features and price. These are the links to the datasets' websites. After clicking the link, you can download/open the dataset by clicking "Download" on the right top of the page.

1. https://www.kaggle.com/datasets/doaaalsenani/usa-cers-dataset(auctionexport.com)
2. https://www.kaggle.com/datasets/deepcontractor/car-price-prediction-challenge

Method

Before considering research questions, we cleaned the dataset by removing some extraneous columns (namely ‘vin’, ‘lot’, ‘condition’ in USA_cars_datasets’, and ‘ID’ in car_price_prediction), and changed names of the columns to let the matched variables of these two datasets have the same name (e.g. price and Price, brand and Manufacturer, model and Model, etc.). Also, we dropped all rows with missing data, only considering the cars with prices over 1000 and under 100000, and the cars with mileage over 10 and under 300000 to filter outliers. Then, we combined the two dataset together to get a more in-depth dataset with more attributes, and now we began to analyze the dataset and try to answer the research questions.

For research question 1, we first used groupby method to get the average mileage of all cars based on their manufacturers, then chose the top 9 manufacturers with the longest average mileage, analyzed the mileage and price of cars and get a scatter plot of it showing their relationships (using the seaborn and matplotlib libraries) of all 9 manufacturers in one plot, while elements in the plot with different colors based on the produced year of cars. There is one thing I want to mention, we excluded all manufacturers with average mileage in integer in order to exclude the cases where there is only one car with such manufacturers. Then we can analyze the plot by stating their relationship (extent of correlation, positive or negative, the slope of the best-fitted line, etc.).

Before digging into research question 2, we splitted our dataset into two parts, with the first 6000 rows serving as "training data" for the machine learning in our project, and the rest serving as "test data" for the later testing. Training data here is used later in research question 3, where we split this training dataset into two smaller training and test data that are used when we train our machine learning model and test it. The test data will be kept away from all code operations, and used as a real-life example for our model once we finish constructing our machine learning model to show the accuracy of the model.

For research question 2, we analyzed numerical data and categorical data independently. Firstly, we plotted the histogram showing the distribution of the number of different prices of cars, and found out it is a right-skewed plot since there are more cars with lower price than cars with higher price. Then we transformed the price by applying a logarithmic scale and got a histogram with log(price) as a normal distribution and combined these two plots in one image. The reason that we have this step is to make it easier to train our machine learning model later in research question 3. After this, to analyze numerical variables (e.g. year, mileage, Cylinders, Airbags), we calculated correlation between price and all numerical variables by using corr() method in pandas, and sorting them from lowest to highest. Then we selected the variables that are most relevant to the price.

Secondly, we focus on the categorical variables. We first filled all null values to be None in order to make our machine learning model easier to be trained later, and explored the relationship between them and the log(price) by drawing the boxplots

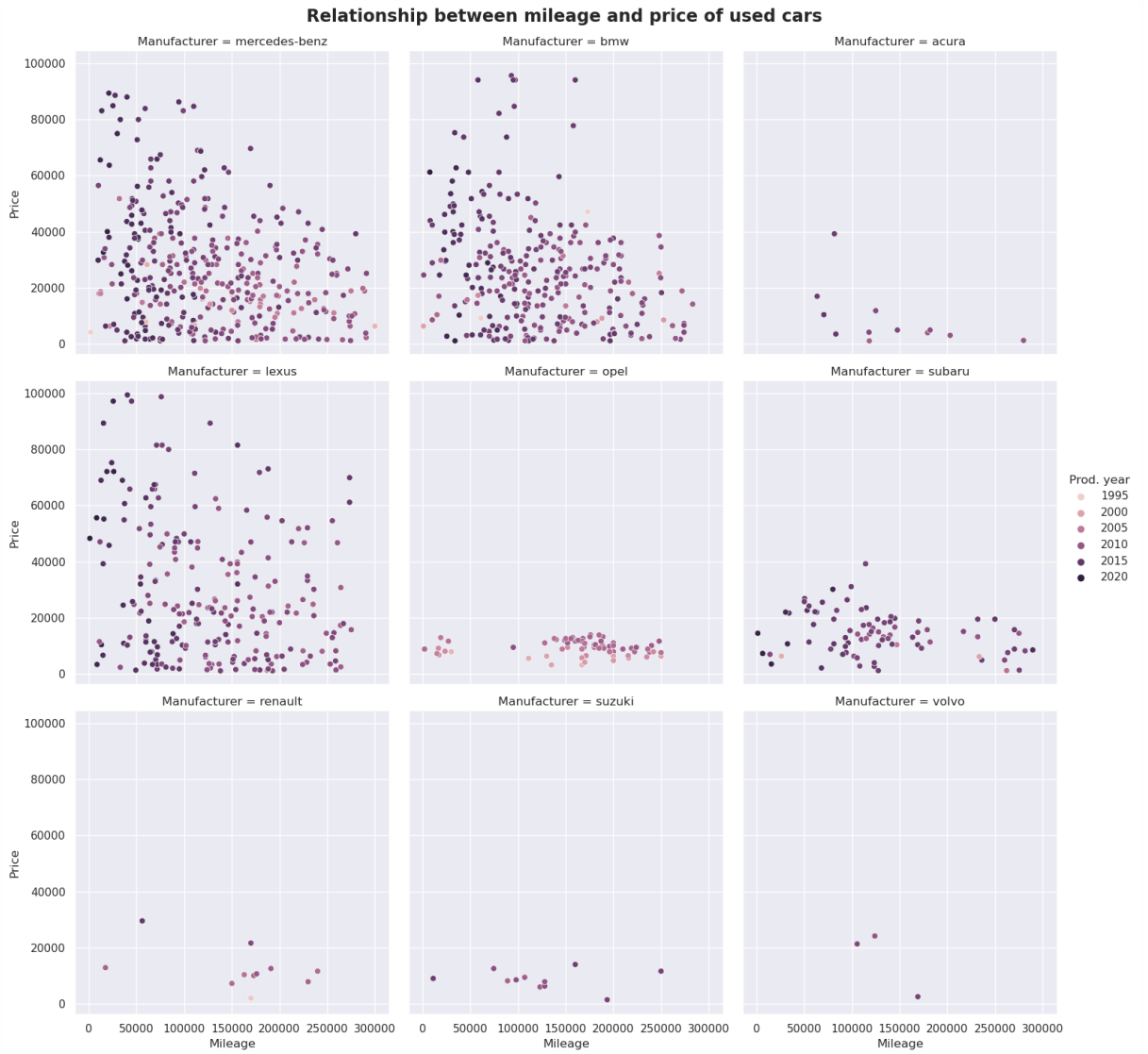
from seaborn. We first plotted the boxplot showing the relationship between the car manufacturers and log(price) as an example, then plotted all boxplots with other categorical variables and used subplots of matplotlib.pyplot to put those boxplots in the same image. By comparing the boxplot, we were able to tell which categorical variables have larger impact than others.

For research question 3, we refined our data before training the model by selecting the features most relevant to the price found in the previous question, which included "Price", "Prod. year", "Mileage", "Manufacturer", "Model", "title_status", "state", "Category", "Fuel type". We checked the skewness of our price and numerical variables, and transformed the data by applying logarithmic scale to the highly skewed features using scipy.stats. Then, we created dummies for the categorical variables.

Following that, we began the machine learning process. We separated the features with labels("price") and used train test split from sklearn.model selection with test size 0.3 to split our data into training and testing. We chose 70% of the total data as test data and the rest as train data because we needed enough data to train our model. Initially, we trained our model with train data using the scikit-learn library's simple linear model LinearRegression. We used function argsort() in numpy to print out the features and their coefficients in this linear regression model in order to determine which features have the strongest impact in our model. In addition, we investigated other machine learning models and selected the Lasso Regression model and the Ridge Regression model for our data. We used Lasso() and Ridge() to train two more models to figure out which model best fit our data. After generating each model and obtaining the predicted price, we evaluated our model by computing the mean squared error (MSE) between the true and predicted prices in our test data and comparing the MSE between these three models to determine which one was best.

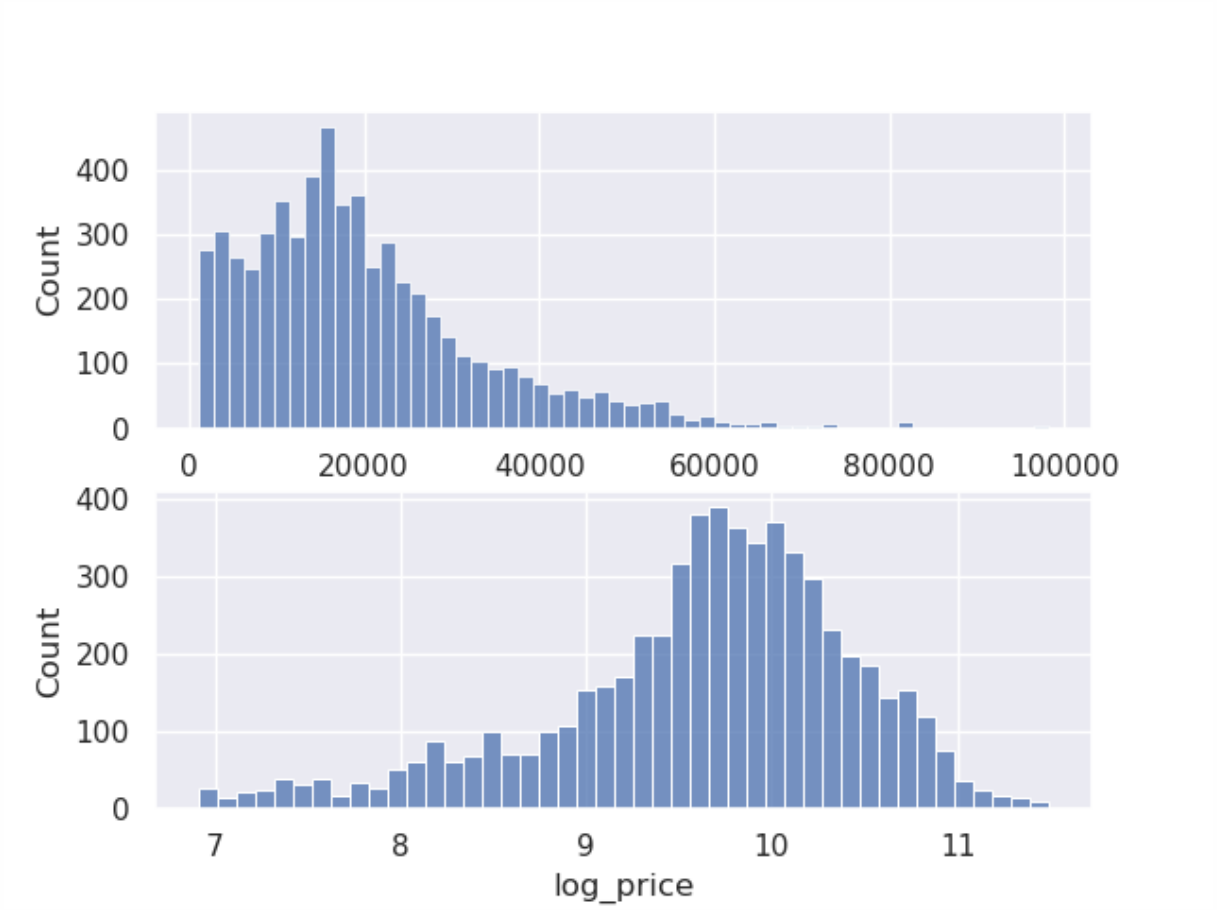
Results

For **research question 1**, we get the plot below:



As we can see, there are some plots containing less dots, meaning that few cars are manufactured by some manufacturers in the image such as renault and volvo. Despite of this, in plots showing the relationship between mileage and price of used cars with manufacturers mercedes-benz, bmw, lexus, and subaru, we can see a negative relation - the car with larger mileage tends to be sold at a lower price - although the relation is not very clear and strong. However, there is no clear relation of the Prod. year of cars and their sold price shown in these plots, which is unexpected since we previously assumed older cars will be sold at a lower price. One of the explanations is that the number of cars produced recently (in 2015 and 2020) are way too many compared to older cars, which makes the dataset more biased thus cannot get a clear relation between the produced year and the sold price of used cars.

For **research question 2**, here is the image contains the distribution of the number of different prices of cars(upper one) and the log_price after our manipulation(lower one):



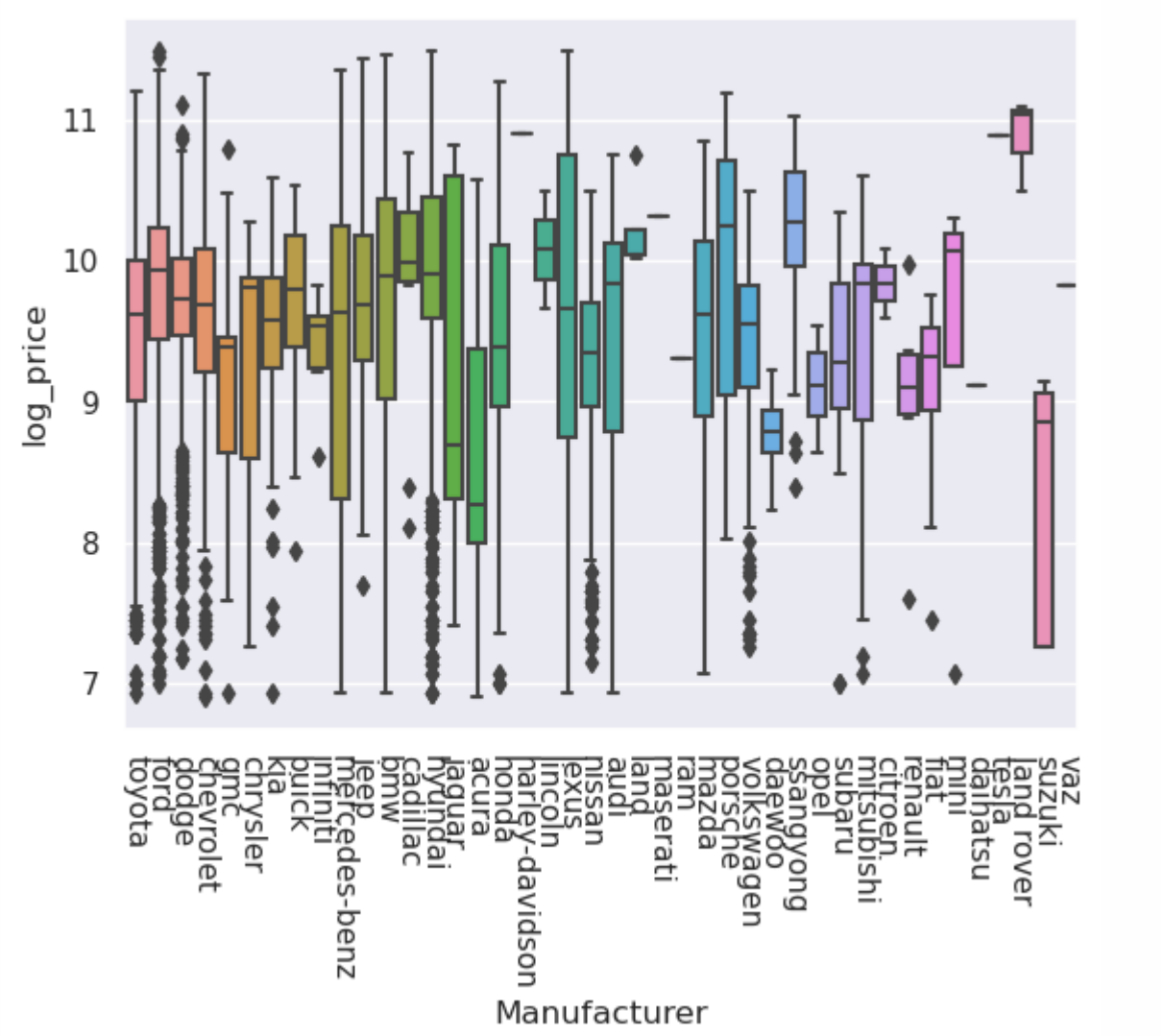
where the upper plot is the distribution of price which is right-skewed, and the lower plot of log(price) after logarithm has normal distribution. Hence, since we would prefer using a linear regression model to train our machine learning model later, we chose the bottom one to satisfy the normal distribution requirement.

Then, we split the variables of our dataset into numerical and categorical variables to find out the correlation between each numerical variable and the price and get the result below:

Mileage	-0.238905
Airbags	-0.215026
Cylinders	0.067166
Levy	0.076660
Engine volume	0.110938
Prod. year	0.286775
log_price	0.872535
Price	1.000000

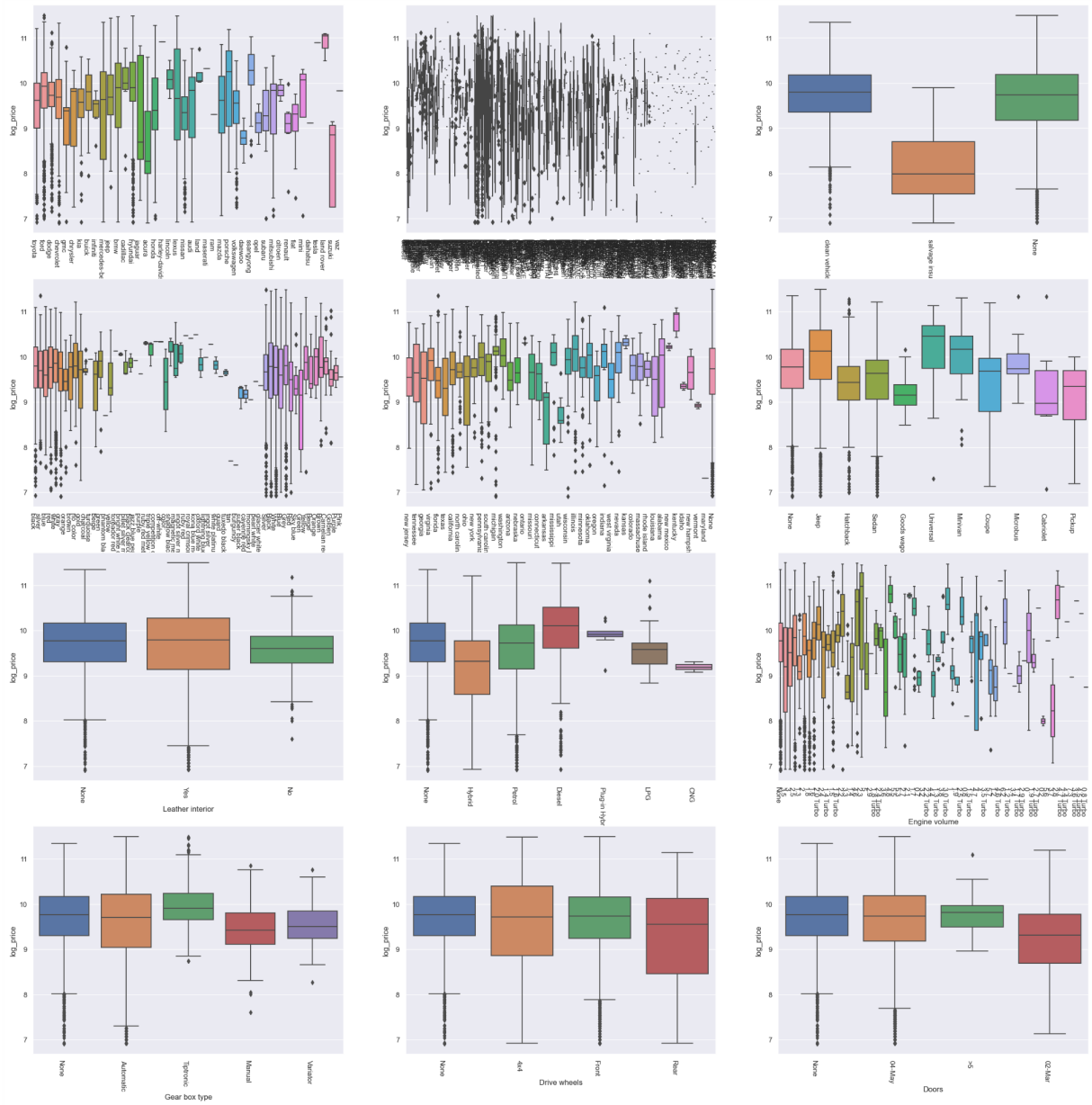
As we can see, the correlation between these variables and price are not very strong, with no correlation coefficient larger than positive or negative 0.3 except for log(price) and price itself. The strongest relationships are between Prod. year & Price, and Mileage & Price where the former is positive and the latter is negative. Therefore, we choose these two variables as “more important variables” and will be used in our next research question. It is important to mention that the larger the Prod. year is, the newer the car is, since in our function, a car produced in 2020 has a larger Prod. year than the car produced in 2010, but the former car is newer than the latter one, therefore causing the positive correlation between Prod. year and price since newer cars tend to be sold at a higher price.

For categorical variables, here is the boxplot showing the relationship between the car manufacturers and log(price):



There are many car manufacturers in our dataset, and the larger the length of the box is, the stronger correlation these two variables have. Therefore, in this plot, the correlation between manufacturers and log(price) is strong based on our observation.

Then, we put all boxplots that show the relationship between categorical variables and log(price) in one image:



where the categorical variables for each subplots are:

Manufacturer	Model	title_status
Color	state	Category
Leather interior	Fuel type	Engine volume
Gear box type	Drive wheels	Doors

As we can see in the image, the boxplot representing the relationship between car models and log(price) is very messy, that's because there are way too many different car models, and the limited length of x-axis cannot contain all their names, thus

causing the mess. Although the model plot looks messy, the length of the boxplot for each model is actually very long, meaning that there is a strong relationship between car models and their sold price.

Conversely, the length of the boxplots showing title_status and Leather interior are short, meaning that the value of log(price) don't change a lot within certain categories in these two variables, so they are less important variables that determine cars' sold price.

Therefore, combining both analysis results of numerical and categorical variables, we find out that Prod. year & Mileage as numerical variables, and Manufacturer, Model, state, Category, and Fuel type as categorical variables, are more important and can better affect the price than other variables. We are going to filter the dataset only with these variables and train our machine learning model later to help us to investigate research question 3.

For **research question 3**, We realized that the absolute value of our skewness of numerical features is quite large (Price:1.414260, Prod. year:-1.030792, Mileage:0.871295). This was not what we expected for creating a linear regression model, since we assumed our data were normally distributed and normalized the price into log(price) in previous questions. Like what we did to the price, we decided to normalize the distribution of our numerical features as well to ensure our data was suitable for the statistical procedure.

After creating the dummies and processing our data, we created our first machine learning model - Simple Linear Model. The Linear regression model's mean squared error is 1.4768088762969262e+18, which is quite large. Using the argsort(), we discovered that the first three most influential features are dummy variables from feature title_status, which is basically the condition of the car. Following that, model and manufacturer are important features that have a significant impact on the price, which makes sense. The second model is the Lasso Regression model, and the mean squared error from this model is 0.006699509756813408, which is significantly less than our simple linear regression model. The mean squared error (MSE) for the third model, Ridge Regression, is 0.0041717616550025725! The third model's MSE is significantly lower than that of our simple linear regression model, indicating a

significant improvement in price prediction by using a different model, which is consistent with our expectations based on our exploration and understanding of new models.

Impact and Limitations

Our findings suggest that, while there are numerous features of a car, a few more important ones, such as the manufactured year, mileage, and manufacturer, have a greater impact on the price of a used car. After we report this information, people who want to buy used cars can use our findings as a guide and choose their cars wisely based on the features they value under their budget. On the other hand, this can help them predict the price of the car that has the features they require.

However, there are still limitations of our analysis and results. Firstly, according to our results, the correlation coefficients of price and other numerical attributes are not large enough to be called “strong”, which all lie in the range of -0.3 and 0.3, meaning that price can hardly be affected by any single attribute to a great extent. In this way, our machine learning model cannot predict the car price only based on a single feature, but has to combine all features together to get a thorough analysis. Therefore, due to this limitation, people who only care about single attributes of cars, such as manufacturer or mileage, may not want to use our model.

Secondly, we identify "more significant categorical variables" by plotting all of the boxplots that show the relationship between each of them and the price which is not efficient. This is a time-consuming process; even though we only have 6000 rows in our train dataset for this project, drawing boxplots with twelve subplots takes significantly longer than others. As a result, if the database grows larger, the time required to run the code increases significantly, and the image generated may appear even more jumbled, making it difficult for people to observe the plots.

Challenge goals

1. Multiple Datasets: For this project, we found two datasets containing the information of second-handed cars in the US. One dataset contains around 2500 car samples, with features including manufacturer, model, produced year, mileage, etc.. We first found this dataset and thought this dataset may not be enough for our analysis. The reason is that this dataset only contains numerical variables like produced year and mileage, but doesn't contain enough categorical variables, which is also a very important kind of attribute, about cars for us to analyze and investigate. Therefore, we found another dataset that contains over 19000 car samples with many new categorical variables such as Leather interior, Engine volume, Cylinders, etc.. So we are able to analyze our second research question for both numerical and categorical variables with the help of these two datasets. Then we combined them together and cleaned them, just as we said in the method section before, and used the combined dataset to do more in-depth analysis with its more comprehensive information.
2. Machine Learning: Explore and deep dive into new models from scikit-learn library's Linear Regression family. We chose to dive deep into the linear models for our data and project purpose after carefully considering and comparing many models on the website. The linear model was chosen because it is a supervised Machine Learning model that seeks the best fit linear line between the independent and dependent variables, which matches our previous expectation. Furthermore, we wanted our model to perform some automatic feature selection in order to determine which features should and should not be included on their own. We discovered lasso as a perfect candidate, a modification of linear regression that penalizes the sum of absolute weight values. We're curious about what else machine learning can do and how intelligent it is. After extensive research in machine learning, particularly the linear regression family, we gained insights into the benefits of each model. We sought something that could try its best to use complex models while avoiding overfitting, and we discovered the Ridge model that estimated coefficients are pushed towards zero to make them work better on new data-sets. Finally, we chose these three models in the Linear Regression Family, and compared them to find the best model which has the lowest mean squared error for our data.

3. Other challenges (Do more work on analyzing the data with a combination of statistical thinking and methods): such as calculating the skewness and normalize those with large skewness using log transformation based on our expectation, calculating statistical features like correlation coefficient for the numerical data, drawing statistical plots such as boxplots to explore relation between all the categorical features and price. In addition to satisfying and realizing these statistical ideas in Python, we explore new method from our familiar library - calculating the correlation coefficient using corr() and log transformation using log() from numpy, and also exploring method from new library, such as calculating the skewness using skew() from SciPy.stat.

   In more detail, before directly applying the data after initial cleaning to machine learning, we decided to thoroughly analyze the data using various methods. We analyzed the data separately for numerical and categorical data, and we used statistical data visualization tools like scatterplots, boxplots, and bar graphs to help us and better express our findings. For the numerical features, because we wanted to follow the statistical features of normal distribution, we used "skew" from SciPy.stat to print out the skewness of our price and numerical variables, and we used log transformation to transform the variables with very skewed data (absolute value greater than 0.75) to approximate normality.

Work Plan Evaluation

We divided the project to at least 3 tasks - including time and workflow:

Date	Objective

Plan: By Thursday, February 23 **Reality**: Met with TA on Friday, February 24 Finished the research question 1 on Thursday, March 2	Meet with TA to discuss the feedback on the proposal. Complete the data-cleanup and complete the research question 1. Expected Time: 2-3 hours. Actual Time: **5** hours
Plan: By Sunday, February 26 **Reality**: Finished the research question 2 on Friday, March 3	Complete the research question 2, record our data/observations, and save/finalize any visualizations. Expected Time: 5-6 hours Actual Time: 5 hours
Plan: By Sunday, March 5 **Reality**: Finished the research question 3 on Sunday, March 4	Complete the research question 3. Come together on a conclusion for our project. Expected Time: 4-5 hours Actual Time: 7 hours
Plan: By Tuesday, March 7 **Reality**: Finished the draft on Wednesday, March 8	Complete the rough draft of our written report and begin to revise the report. Expected Time: 3-4 hours Actual Time: 8 hours
By Friday, March 10	Finalize our written report and submit all of our work.

We used nearly double the time we planned to complete data cleaning and research question 1. We anticipated that our research question 1 would be simpler and require fewer operations than the other two questions, so it would not take as long. However, we completely overlooked the messy tasks that must be completed when dealing

with new data. We spent a lot of time thinking about how to deal with different variable names and missing values, as well as how to clean, filter, and combine data before diving into question1. Question 1 is also more complicated than we anticipated. At first, we got an incredible jumbled graph from plotting the relationship between mileage and price directly. The most time is spent not on writing code, but on imagining a way that clearly expresses our purpose. Finally, we decided to select the top nine manufacturers based on average car mileage and display their scatterplots in an image.

All of the remaining tasks took longer than expected because, in addition to writing code, we tried our hardest to explore, learn, and discover methods that could be used for our project.

Fortunately, with our cooperation and hard work, even though it took a little longer than expected, we successfully completed the project we were looking forward to!

Testing

Before choosing the most relevant features for our model and exploring and determining which models to choose, we manually separate our entire dataset of 12000+ rows into two sub-datasets - "train data" vs "test data". The "train data" contains the first 6000 rows of the entire dataset, while the "test data" contains the remaining 6000+ rows. We only used "train data" in all of our previous analyses.

Now, after selecting the most relevant features and determining the models with increasing behavior (having smaller and smaller mean squared error), we can use the "test data" we saved earlier to see if we were correct in our machine learning model selection. We added a method in our python code called test_smaller() to pass the "test data" into our machine learning. After running the code, the mean squared error of each model we got has the same pattern from the previous "train data". The simple linear regression model has the largest mean squared error, and the ridge regression model has the smallest mean squared error!

Collaboration

We didn’t consult with any other people and resources during the project aside from the course staff and team members.