DSC 672: Predictive Analytics Capstone

Milestone 2 - Preliminary analysis

Stanford Car Dataset: Vehicle Recognition Project

Alexandre Girault (Team Manager)
Chris Shaffer
Sean Sungil Kim
Ryoh Shinohara
Yanxi Cai

Spring 2019

Dataset Description

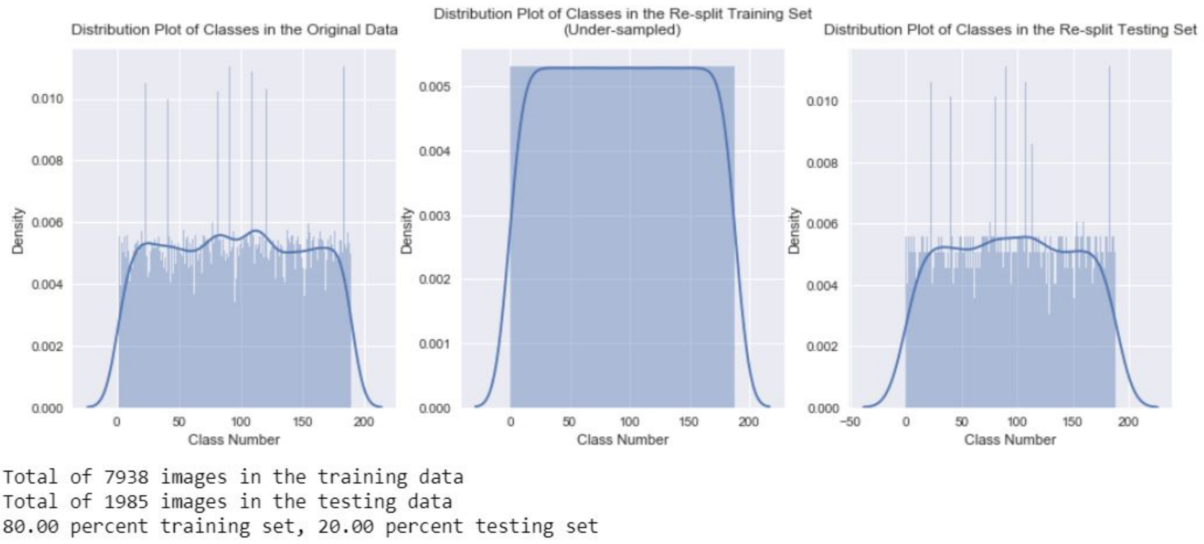
The Stanford Car Dataset is a large-scale, fine-grained dataset of cars that was generated by Dr. Jonathan Krause and his team from Stanford University. To generate an initial list of car labels, the authors crawled an unspecified popular car website to create a list of all cars from 1990 to 2012. Because many car models do not change their appearances across model years, the authors merged car classes with similar visual features using a technique called perceptual hashing. This technique compares two media objects such as images to see if they are different from each other. In this study, the authors used Hamming distance, the difference between two strings of numbers that represent the pictures, as a measure to determine the dissimilarities between car classes. After the initial round of perceptual hashing, the authors generated 197 classes of cars. To expand on the pool of car images, Dr. Krause and his team collected car images from Flickr, Google, and Bing. While these search engines allowed the authors to collect many images, they needed to verify that the collected car images were from the correct car classes. To verify the identities of these images, the authors utilized Amazon Mechanical Turk (AMT) workers to annotate the car images with the correct car labels. The car identification task contained an image of the car that needed to have its identity verified, an image of the actual car from the class of interest, and an image of a car from a different class that could easily be mixed up as an image from the target class. Based on the two images with confirmed classes, the workers must decide whether the unverified car image was from the class of interest. If not, the workers annotated the image with the correct class label. For the workers to qualify for this task, they needed to pass a series of tests that contained some of the most difficult cars to identify.

To determine the quality of annotations, the authors used a technique called Get Another Label (GAL), which is an algorithm using expectation-maximum, a type of maximum-likelihood algorithm that estimates values for model parameters for incomplete data, that estimates the probability that an image is from a certain class while also determining the quality of a worker based on their correct annotations. The criteria for GAL for the car annotation task were: 1) an agreement of workers on the correct car class of an image and 2) the ability of workers to identify “gold standard” images, which were images that the authors knew the correct labels. After the GAL probabilities of a candidate image exceeded a certain threshold, the image was put into the target class. GAL was also used to further weed out poor quality workers by assigning more and more images to users that have low scores, discouraging them to continue the task. After obtaining the set of images with assigned classes, the authors utilized a different group of AMT workers to assign bounding boxes, the section of an image that contains the target object, using a technique presented by Fei-Fei et. al. To further remove duplicate images, the authors used another round of perceptual hashing on the images based on the bounding boxes, yielding a total of 16,185 images with 196 classes of cars.

Data Preprocessing

The dataset contained no missing values, so no imputations or data removal was required due to the nature of image data. In terms of Exploratory Data Analysis, the class labels were split to explore the individual Make, Model, Type and Year levels of the labels. The string-formatted labels were split by a space, then the output of that were categorized into the Make, Model, Type and Year levels. This was tricky, since some the Make and Model levels had different lengths (for instance, Aston Martin vs. BMW in the Make level and Sonata vs. F-450 Super Duty Crew in the Model level). Due to the lack of domain knowledge, there may exist miscategorized class information; however, this extraction of class label levels were performed to the best of our abilities. There were 196 classes originally. Because of this high total class number, the levels of class labels were analyzed with the hopes of reducing the total class number. Initially the class labels were analyzed by human eyes. Our group believed that there were no multiple Make+Model yearly labels after the first examination. However, this was in fact not correct, proven by the Model vs Year scatterplot per Make graph demonstrated in this link: https://rshinoha.shinyapps.io/jitter_plot_of_car_makes_across_years_1990-2012/. Upon further inspection and research, the year level of the class labels were not accurate, since there were visual differences among the images of the same class. Analyzing how the dataset was produced, the creators of the Stanford Cars Dataset confirmed that a bunch of different yearly models were simply grouped together because of the minor differences in their published research, as mentioned in the dataset description section. The Year level was decided to be dropped due to these reasons. This Year removal only resulted in a slight decrease in total class number, from 196 to 189. Since 189 was still very high, a dataset that only had Make as the class label was produced as a back-up plan, in case the results from Make + Model classification were not significant.

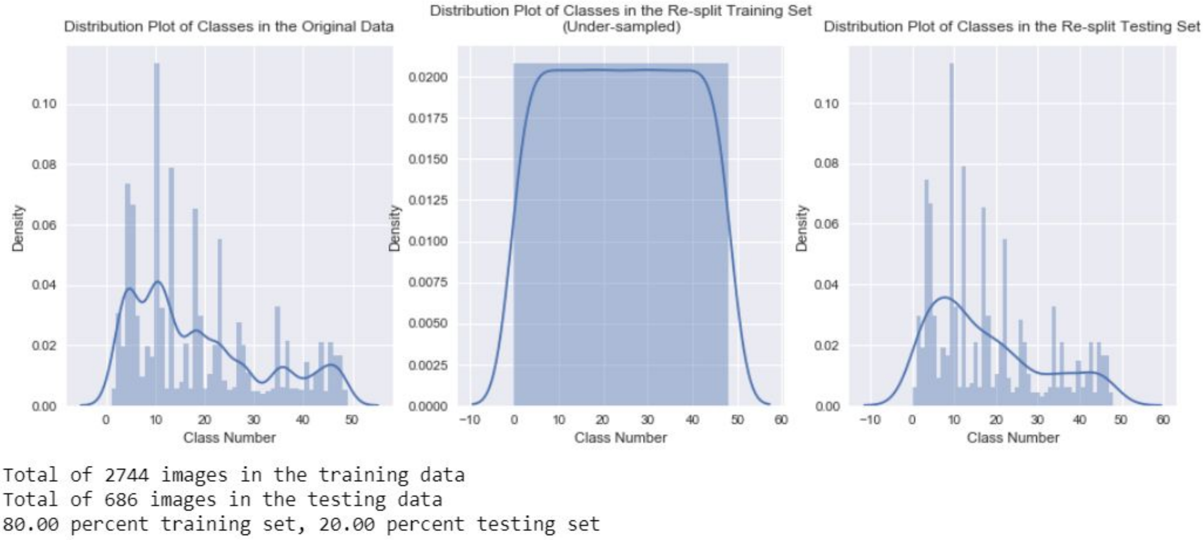
While the Stanford dataset contained pre-split training and testing data, the 50:50 split was not ideal for the purposes of this study. Therefore, the two sets of data were concatenated in order to re-generate a custom training-testing split. Due to the class imbalance issue shown in Figure 1 below, random undersampling without replacement was used to create the final training-testing split. A random 80:20 split was performed on the concatenated Make+Model data, where the 80 was the under-sampled training set and the 20 was the imbalanced testing set. The testing set was kept imbalanced in order to preserve the original data integrity. This would also provide accurate true testing performance due to the maintained data integrity. Undersampling rather than oversampling (or SMOTE) was utilized because of the restrictions in terms of computing power and time costs.



```
Total of 7938 images in the training data
Total of 1985 images in the testing data
80.00 percent training set, 20.00 percent testing set
```

Figure 1: Class Distribution Graphs Comparing Original vs. Under-sampled Training vs. Testing Using the Make+Model Class Label Data.

The same random undersampling without replacement and random 80:20 split processes were performed on the concatenated Make-only data as illustrated in Figure 2 below.



```
Total of 2744 images in the training data
Total of 686 images in the testing data
80.00 percent training set, 20.00 percent testing set
```

Figure 2: Class Distribution Graphs Comparing Original vs. Under-sampled Training vs. Testing Using the Make-only Class Label Data.

While the initial expectations were to create models that classifies 196 classes of car models found in the dataset, this appeared to be not feasible in terms of computational power of most team members' devices. Moreover, the Stanford Cars Dataset had images that had no apparent order or structure. The angles the images were captured at were all different, as portrayed in Figure 3 below. It is apparent that the unstructured nature of the dataset is going to be a major hurdle in our classification task; there is high variability in how the photographs were taken.

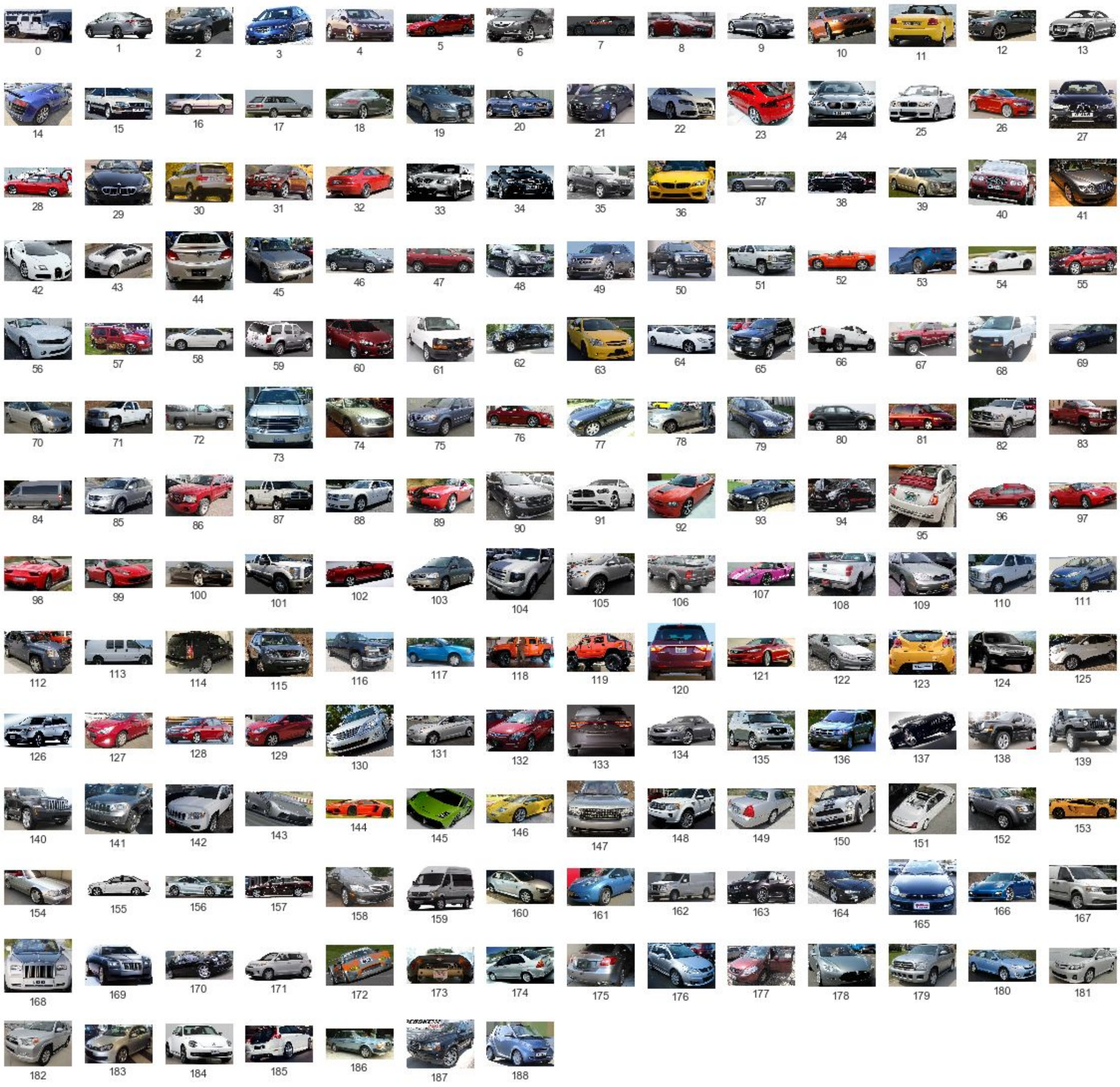


Figure 3: One Image from each of the Original 196 Classes

Descriptive Statistics & Further Explanations on Class Levels

The original dataset defined a ‘class’ as the combination of make, model, and year. This yields 196 individual and unique classes. An example of one of these classes is shown in Figure 4. The class levels were parsed into the components also shown in Figure 4. It may be possible to extract more useful information by separating these characteristics.

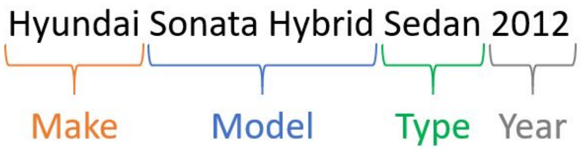


Figure 4: Example Class and Separable Characteristics

The following table provides specific descriptive statistics from the entire original dataset. The image dimensions (height, width, and channels) were added to support future modeling decisions. Due to the way that the image of this dataset was created, a thorough Exploratory Data Analysis of the original class distributions was highly desired.

	ClassNo	Class	Make	Model	Type	Year	Image Height	Image Width	Color Channels
Type	Integer	String	String	String	String	Integer	Integer	Integer	Integer
Uniques	196	196	49	177	13	16	Several	Several	3
Mean	-	-	-	-	-	2009.56	308	573	-
Std Dev	-	-	-	-	-	4.43	214	375	-

Table 1: Select Descriptive Statistics

Original distributions of the individual Make and Type are demonstrated in Figure 5, 6 and 7. The original distribution of the Model was not demonstrated, due to the big size of the graph.

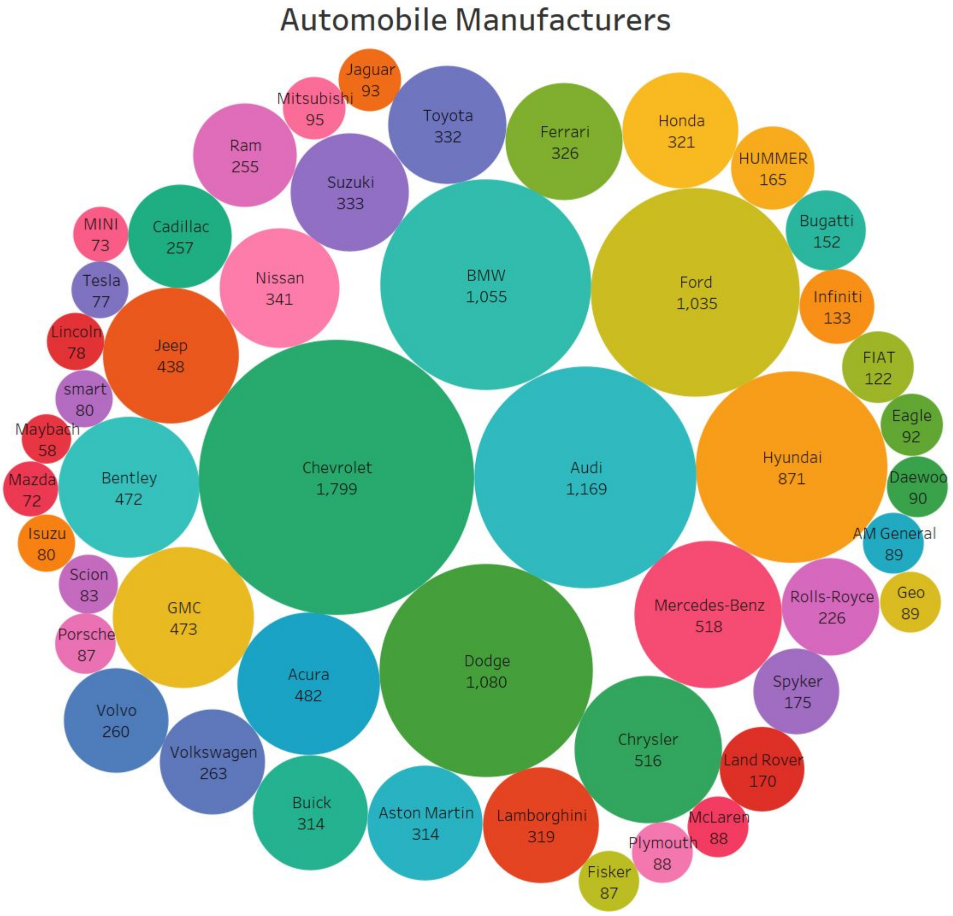


Figure 5: Bubble Chart of Make

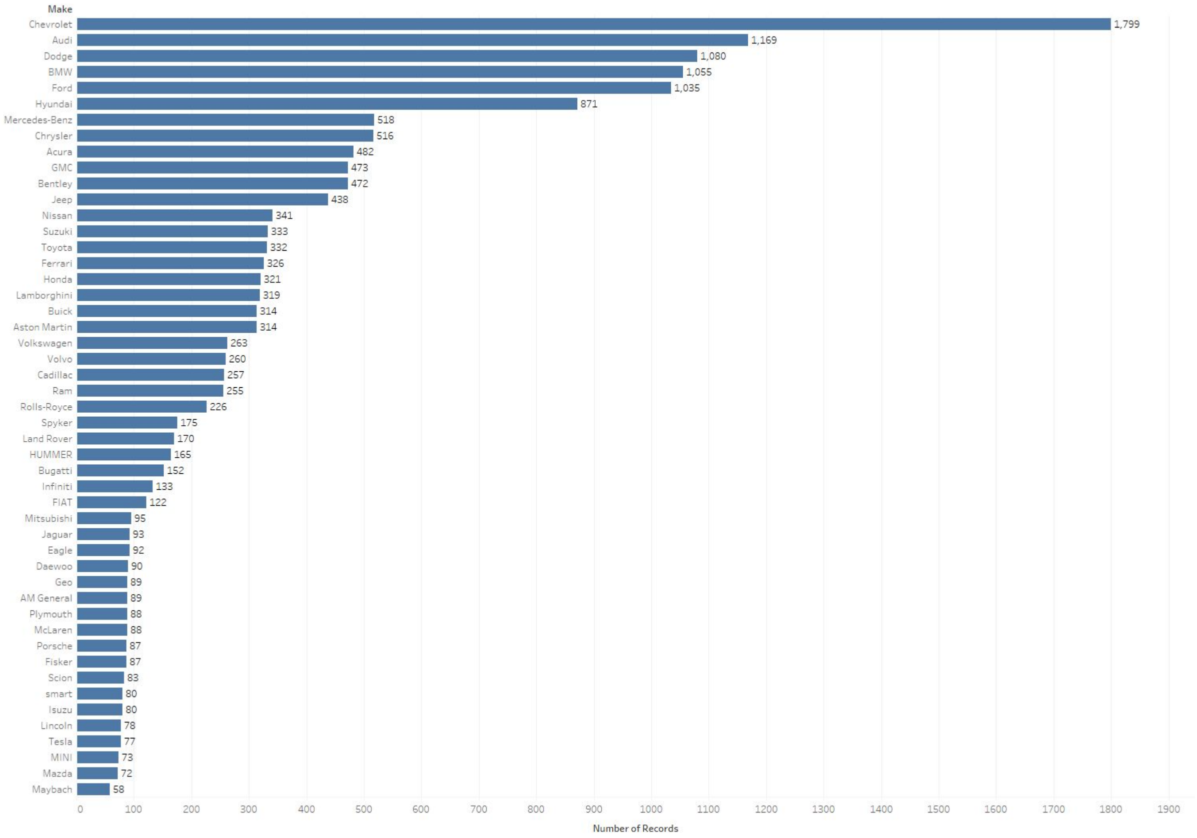


Figure 6: Original Make Distribution on the Concatenated Original Dataset.

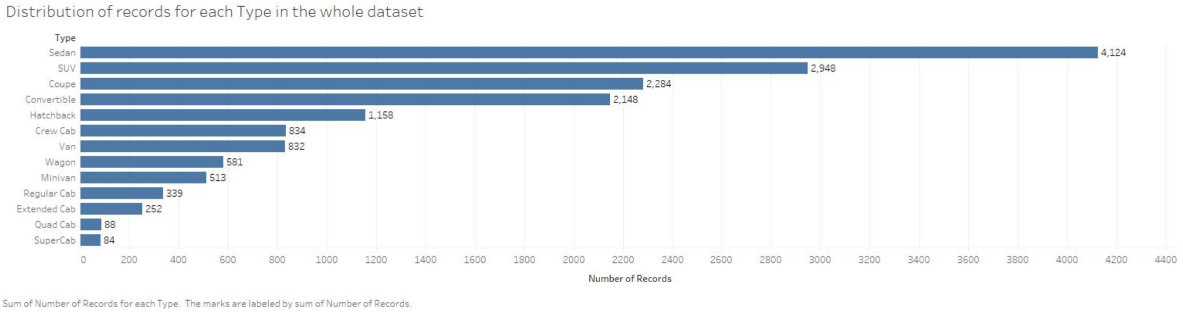
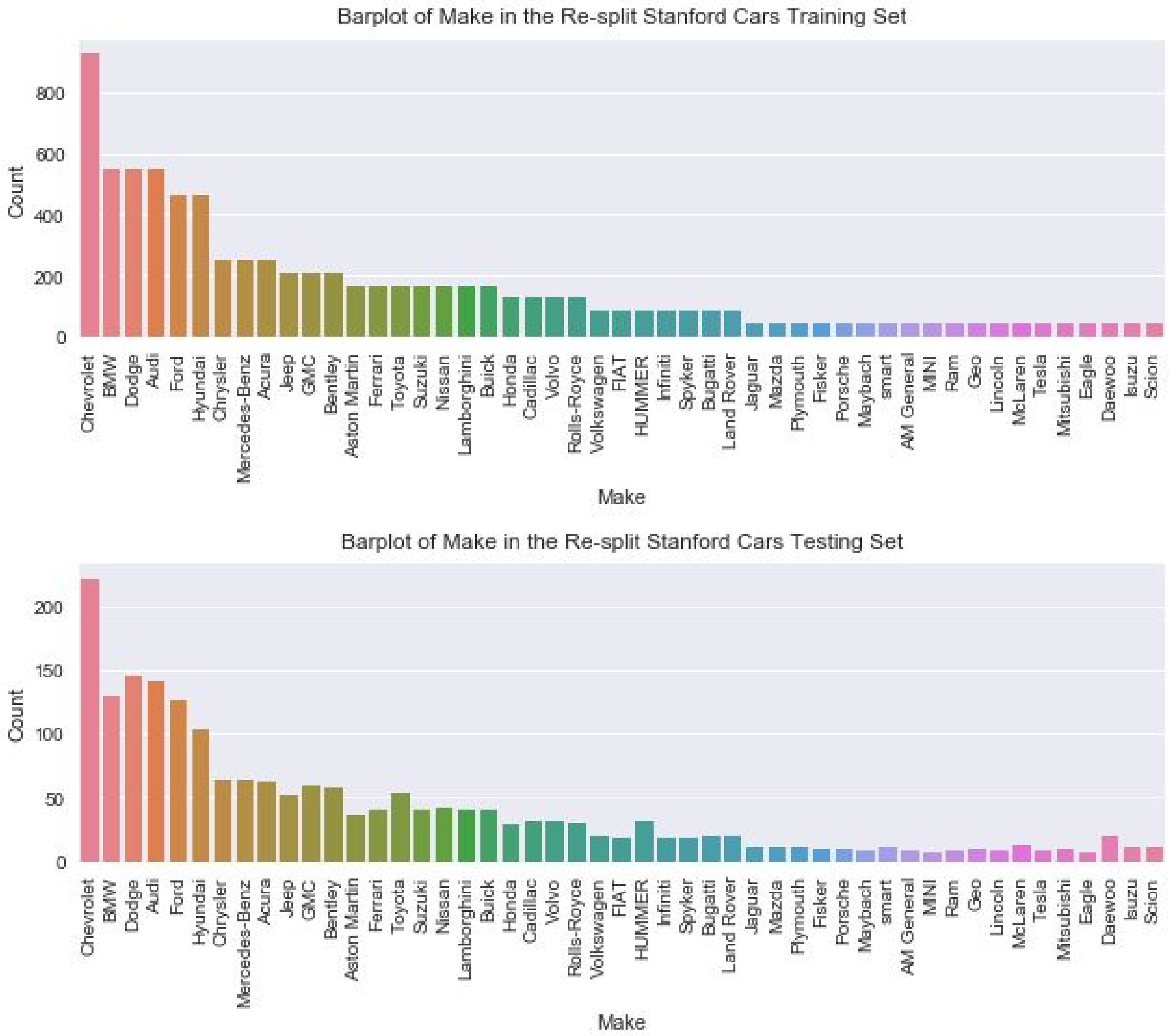


Figure 7: Original Type Distribution on the Concatenated Original Dataset.


Barplot of Make in the Re-split Stanford Cars Training Set

Barplot of Make in the Re-split Stanford Cars Testing Set

Figure 8: Training vs. Testing Make Distribution Graph

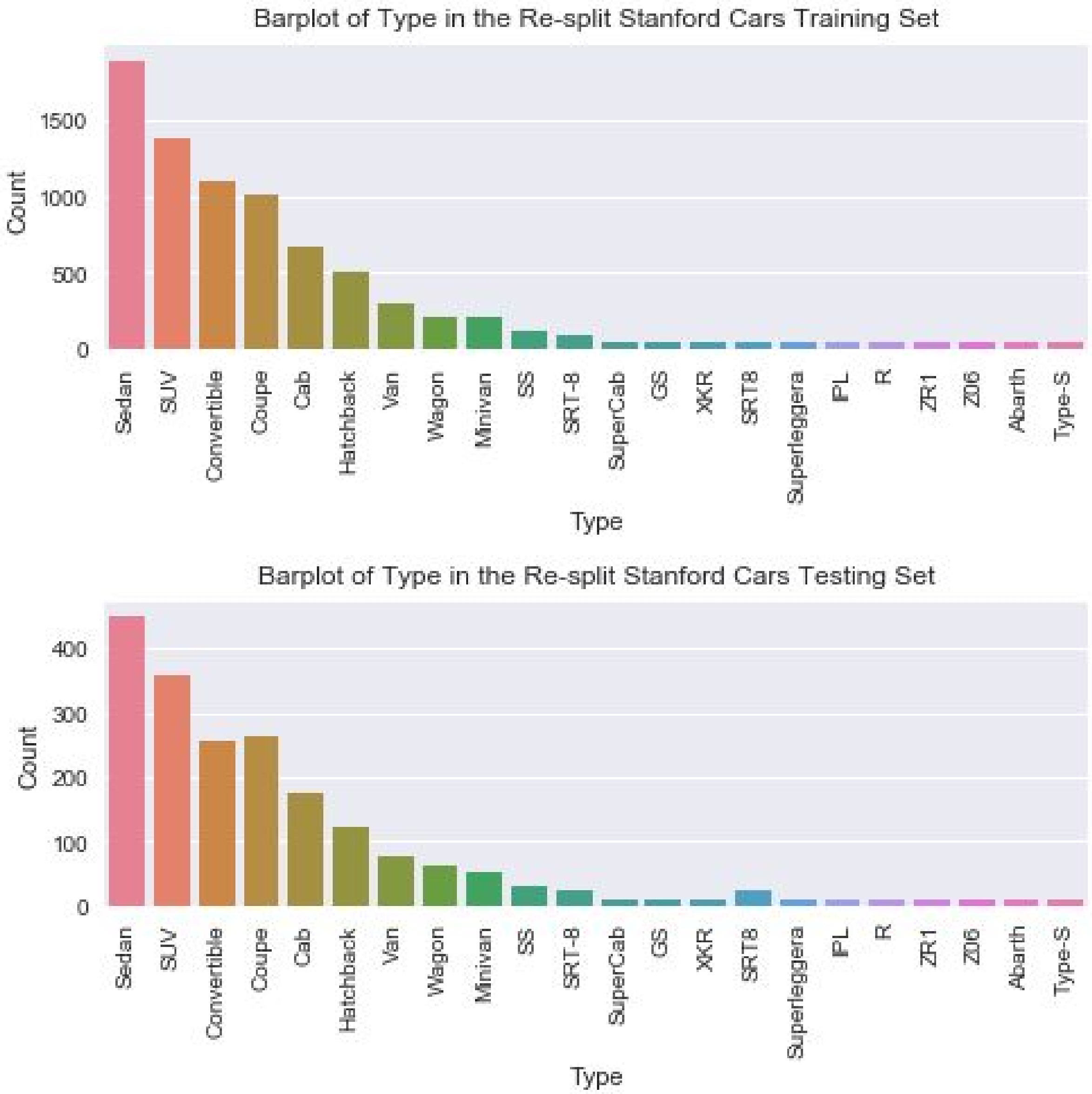


Figure 9: Training vs. Testing Type Distribution Graph

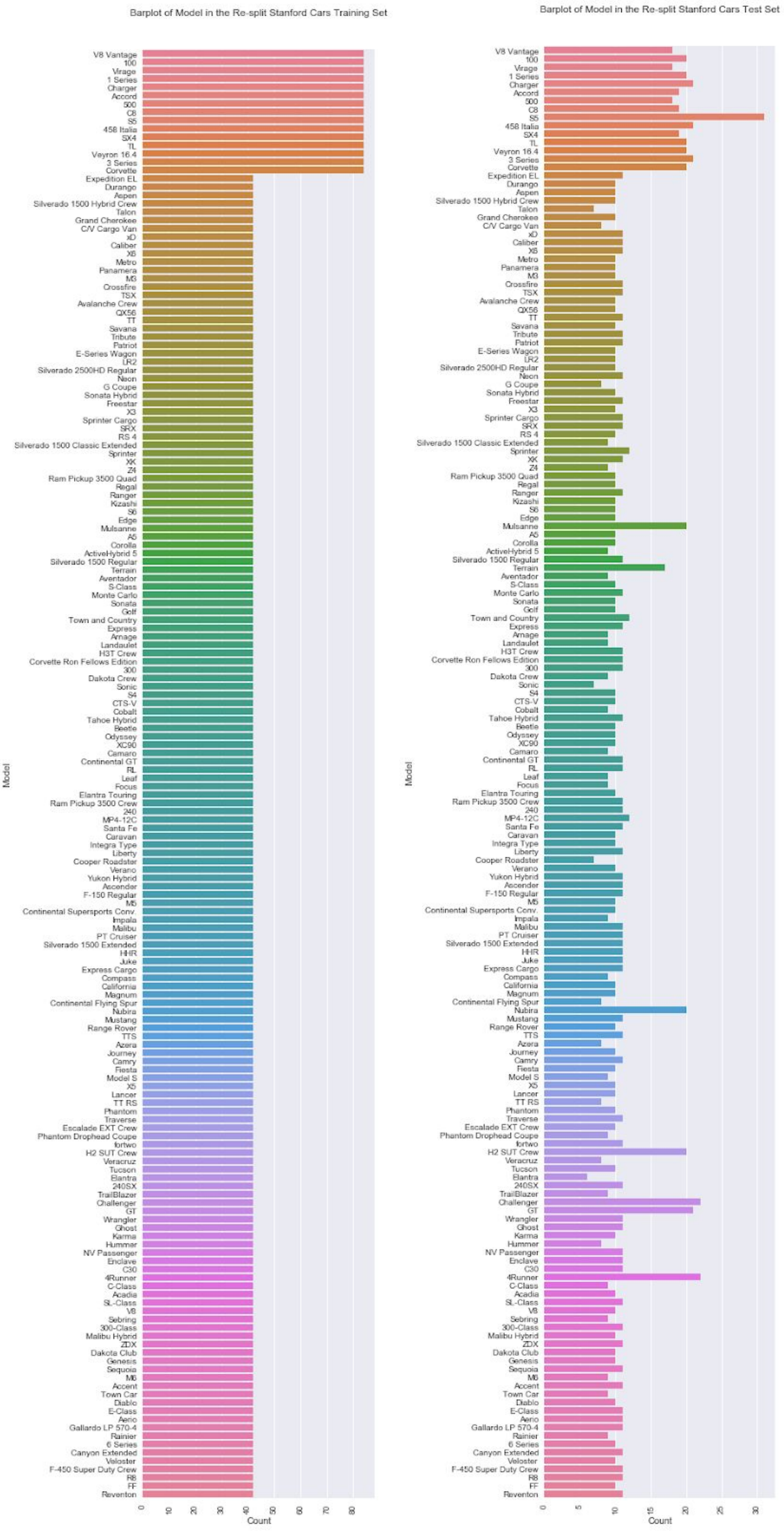
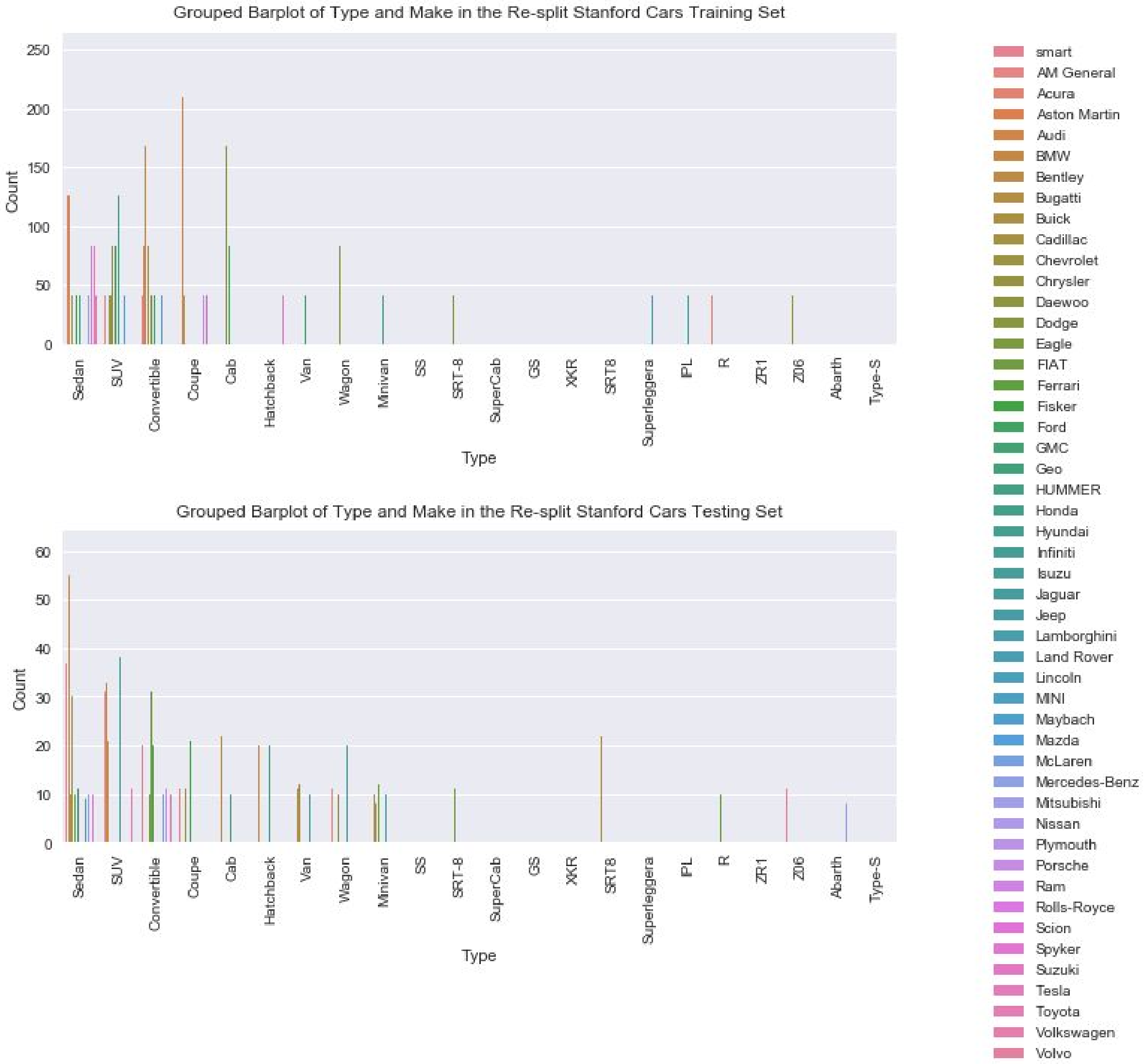


Figure 10: Training vs. Testing Model Distribution Graph



Grouped Barplot of Type and Make in the Re-split Stanford Cars Training Set

Grouped Barplot of Type and Make in the Re-split Stanford Cars Testing Set

Figure 11: Training vs. Testing Cross-tabulation Graph of Type and Make

The Figure 8, 9, 10 and 11 display the new data distributions after the 80:20 split with undersampling. To facilitate viewing the plots, the attributes (x-axis) in the training and testing plots are in the same order. As most apparent in the distribution of the horizontal bar plot of car models, the class imbalances are successfully evened out in the post-undersampled training set. As seen in the various plots of model and make counts, there is an uneven distribution of different car makes, models, and types. American car makers made a large proportion of the dataset with European car makers following. Several less-common car makers also exist in the dataset such as Spyker, Fisker, and Plymouth. Considering how the top selling cars in the US in the past few decades also contained a large portion of Japanese manufacturers in addition to US, German, and Korean manufacturers, this dataset is most likely not a representative sample of car makes commonly seen in the US. The car type count appears more representative of distribution found among US drivers. Sedans and SUV dominate the dataset, with approximately 40% of cars belonging to the two types. Like car models, there are obscure car types such as

QuadCabs and SuperCabs. The high presence of convertibles further indicates some bias in sampling methods of car classes.

Another odd characteristic of this dataset is the sparse nature of the data for car models made before 2007. While the authors of the Stanford Car dataset claimed to have used car models throughout the years 1990 through 2012, nearly half of the dataset is from 2012. This indicates that the Hamming distance threshold used in the data collection method was most likely insufficient in distinguishing subtle visual characteristics between different model years of cars. To back up on our suspicions of insufficiencies in the car labels, we created plots that displayed car models across car model years. The jitter plots shown in the Shiny app accentuates the sparse distribution of car models across the car model years. Most car models only have one or two different model years despite the authors obtaining various car models throughout 1990 to 2012. For an example, Chevrolet Impalas announce new models every year, but this is not reflected in the plots. Due to the inconsistencies in the year attribute, the attribute was removed from further analysis as mentioned in the preprocessing section.
https://rshinoha.shinyapps.io/jitter_plot_of_car_makes_across_years_1990-2012/

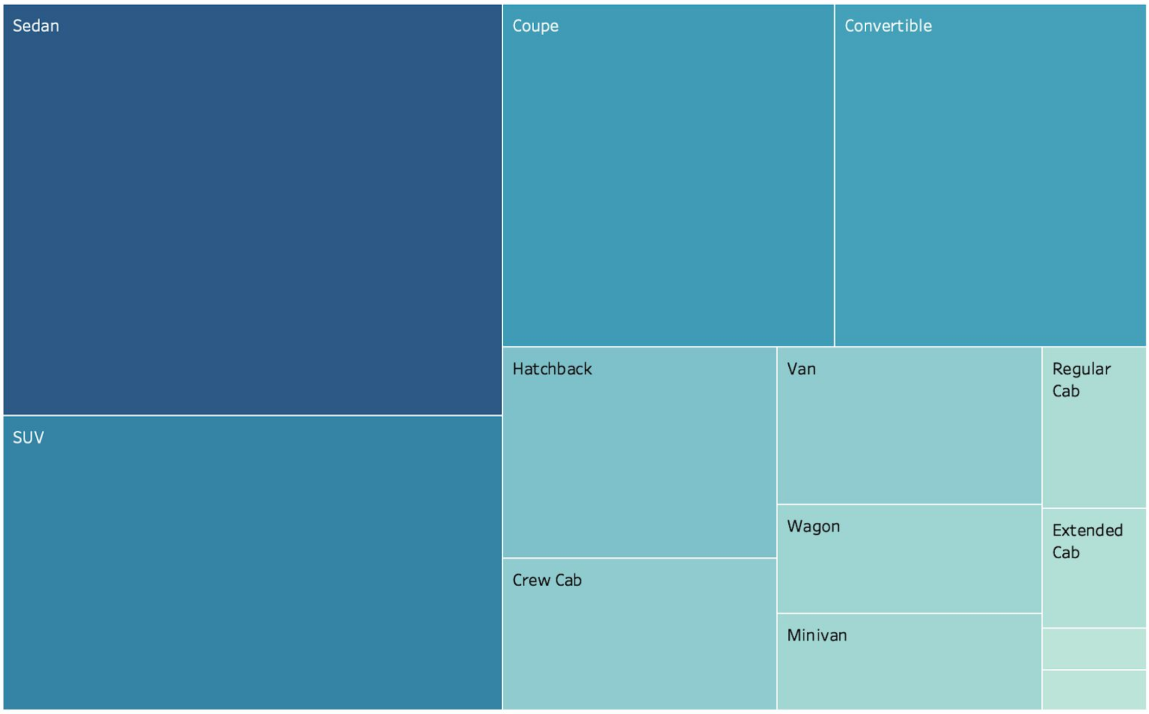


Figure 12: Treemap of All Vehicle Types

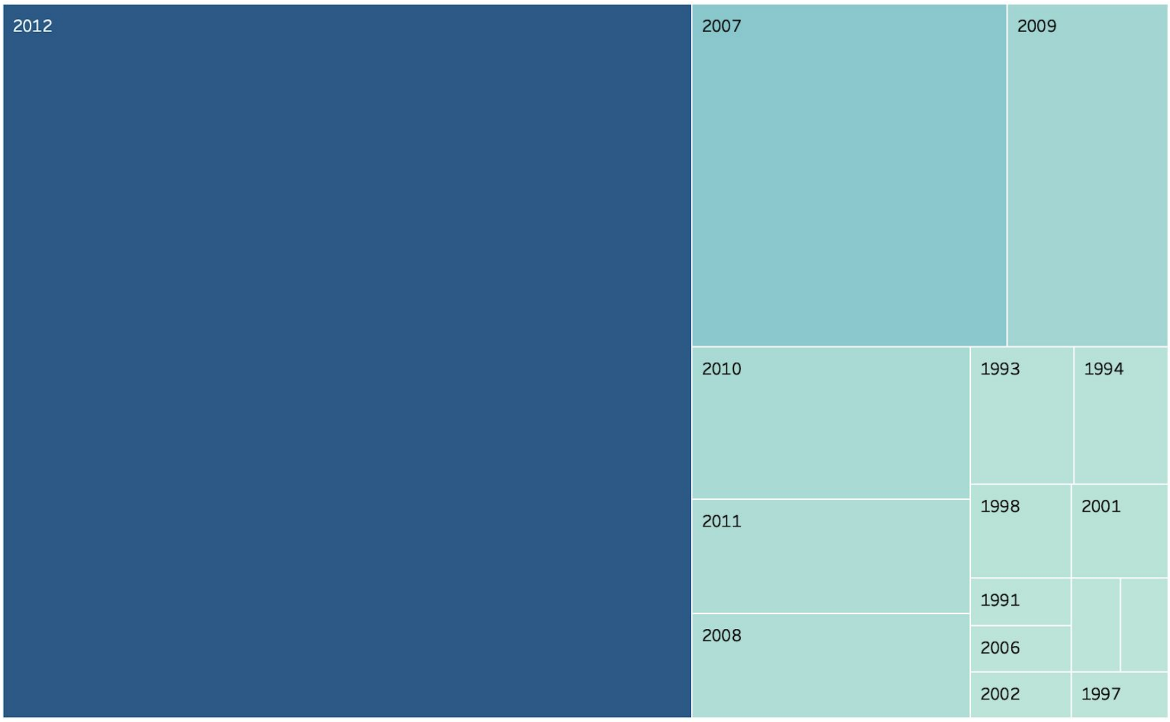


Figure 13: Treemap of All Vehicle Years

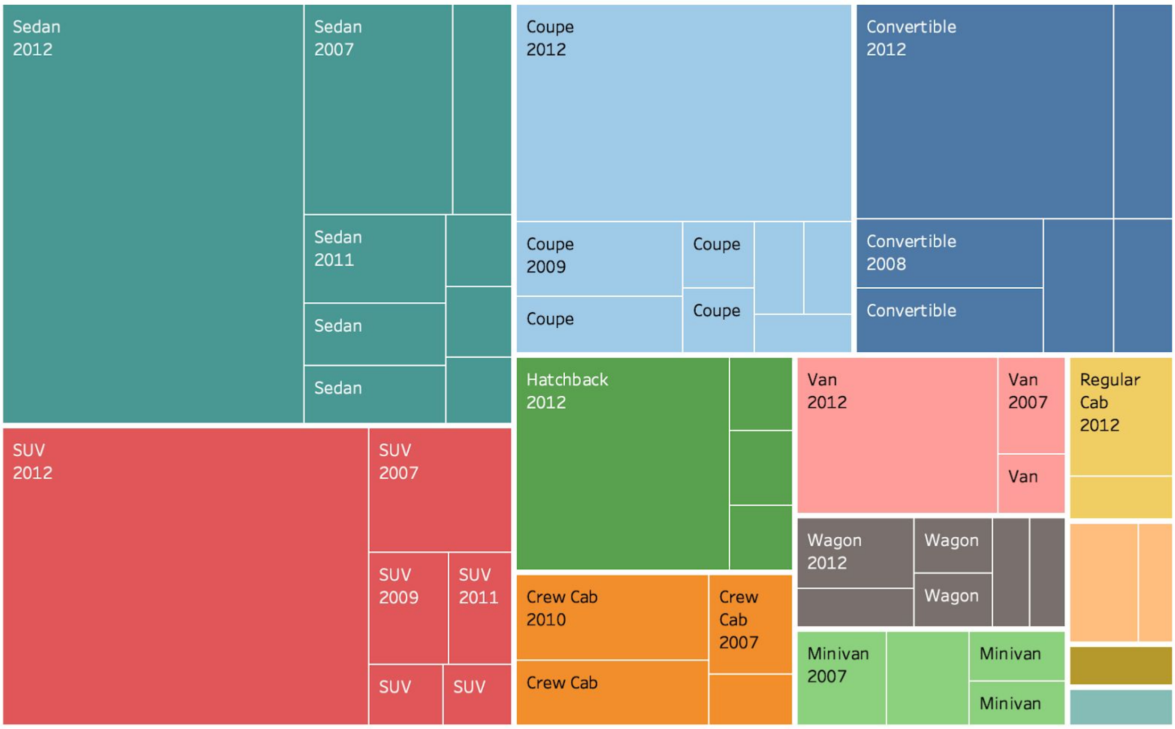


Figure 14: Hierarchical Treemap of Vehicle Types and Year

These tree maps provide additional visuals on attribute distributions. The car type and model year tree maps are coded with both area and color to emphasize the differences in distribution. With the tree map of the model years, it's especially apparent that there is an issue with model years from the class labels of the original dataset; 2012 makes up over half of the car model years in this dataset. The third tree map is hierarchical, combining both vehicle type and year. This further emphasizes the issue with the Year attribute, since 2012 occupies models occupy the greatest area in nearly all car types. Hatchbacks display an especially skewed distribution, with nearly 75% of the area occupied by models from 2012.

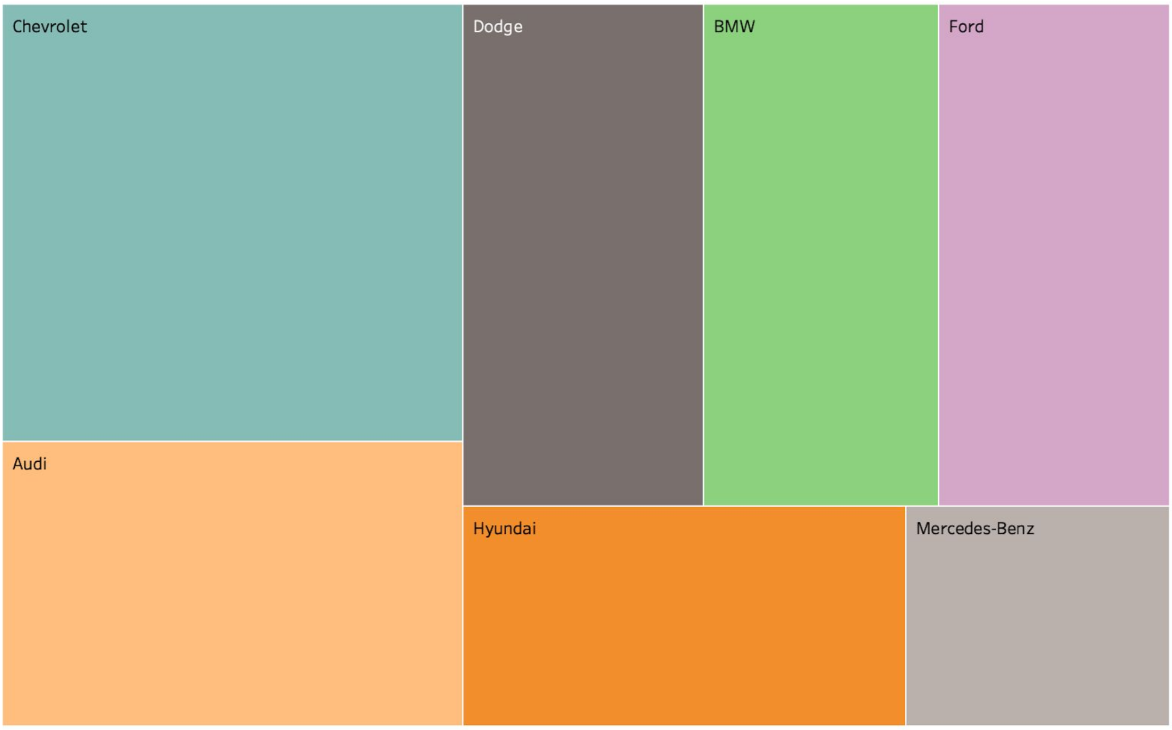


Figure 15: Treemap of Top-7 Vehicle Manufacturers

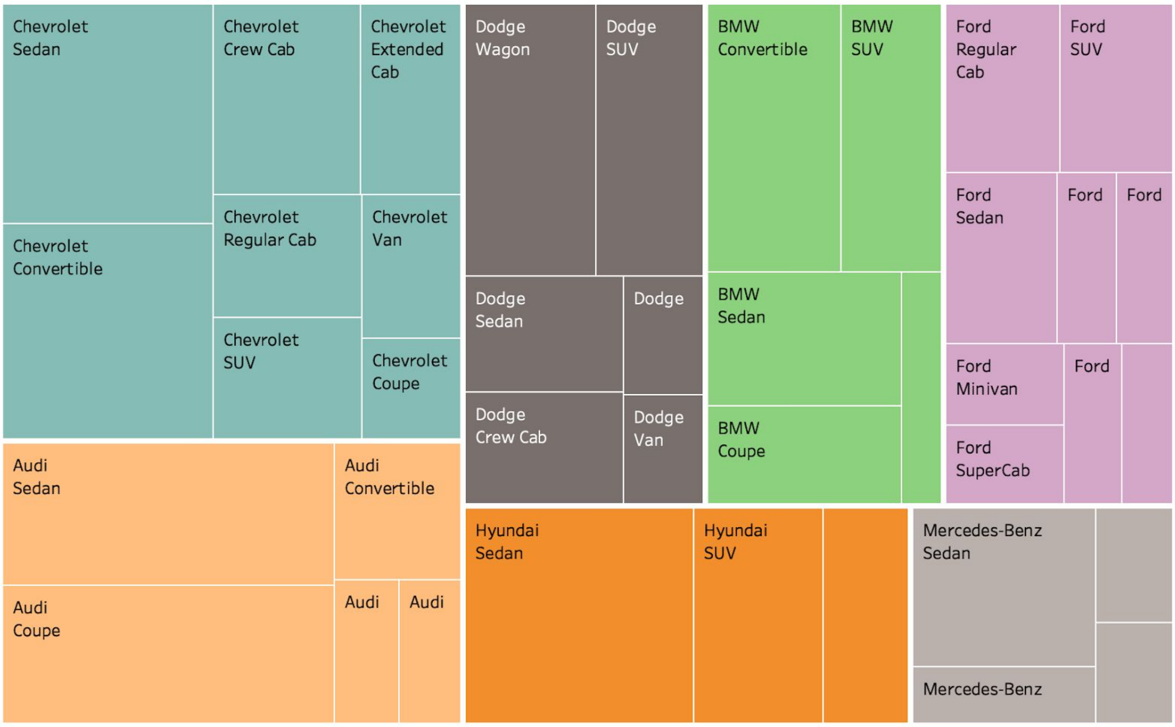


Figure 16: Hierarchical Treemap of Top-7 Vehicle Manufacturers and Vehicle Type

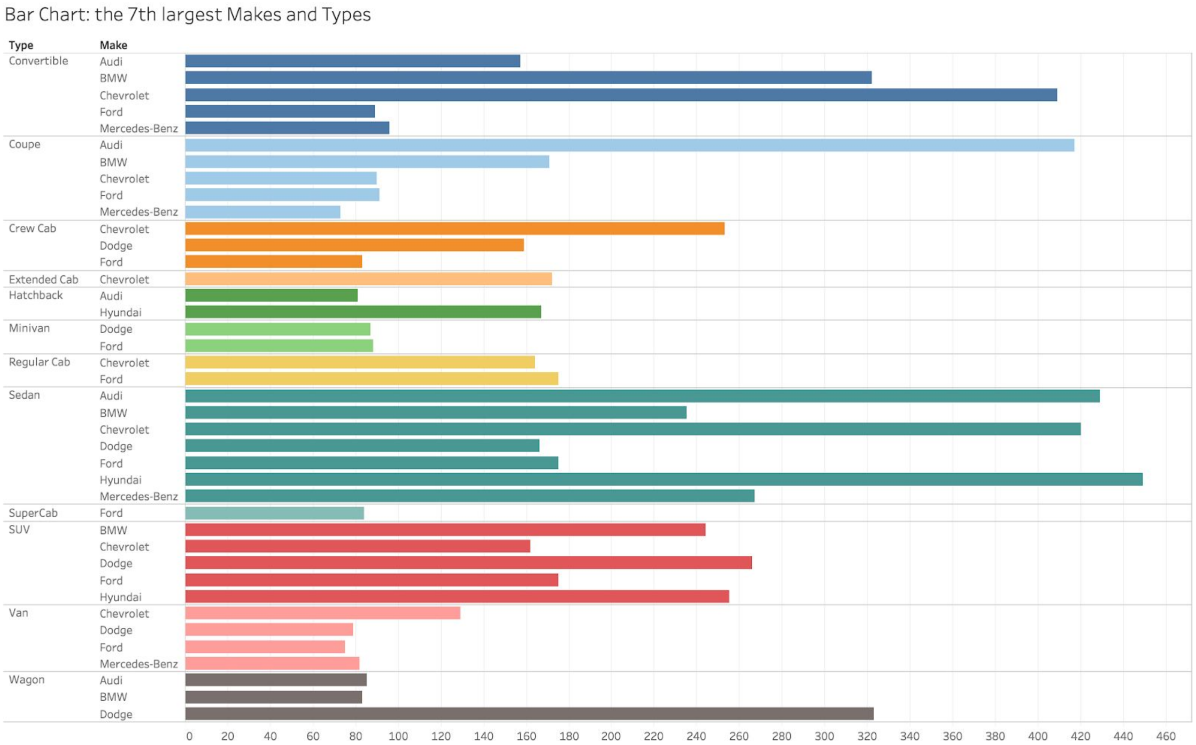


Figure 17: Hierarchical Bar Chart showing Top-7 Manufacturers for each Vehicle Type

The Figures above focus on the seven major car makes in this dataset. As mentioned in previous sections, absence of Japanese car makers, relatively low counts of Ford vehicles, and high counts of luxury vehicles such as Audi and Mercedes-Benz being in the top seven car makes indicate issues with image sampling. While the tree plot of makes and car type display car type distributions approximately representative of what we see on US roads for Hyundai and German car makes, the American car makes have relatively strong presence of various types of cabs. Since Ford Focus and Fusions are very common sedans found on the road owned by average US families and F-150s are regularly ranked as one of the top-selling cars annually, it would be expected that sedans and trucks make up a large portion of the tree maps for Fords. However, the different types of cabs occupy a greater area of the Ford section, indicating sampling methods of this image collection. The bar chart of the different car types and their respective composition of manufacturers also shows issues in the distribution of car images. Certain manufacturers are over or under-represented among vehicle types. This is an additional consideration for future modeling approaches, since it may be more feasible to train models that can identify vehicle type and make separately.

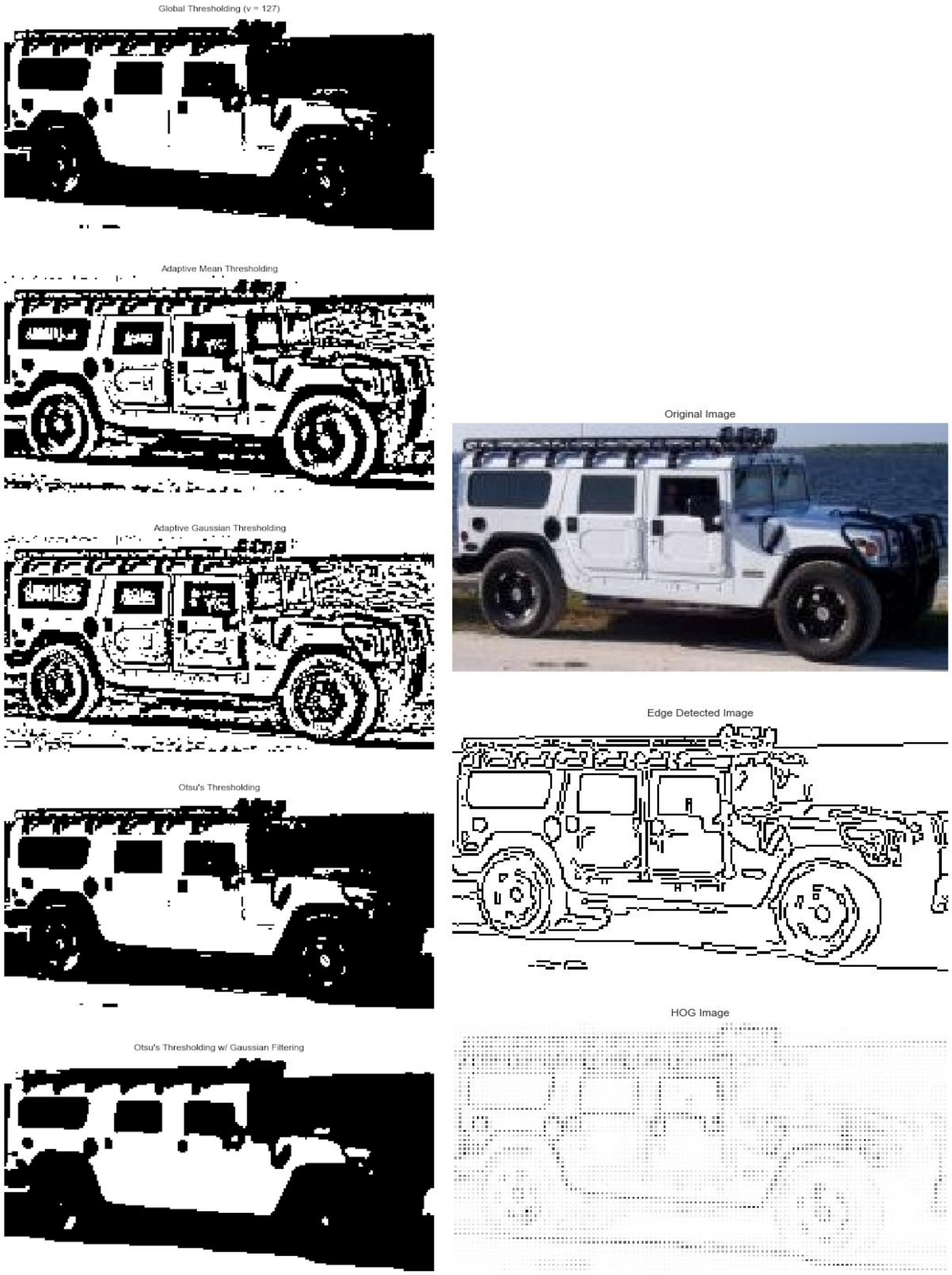
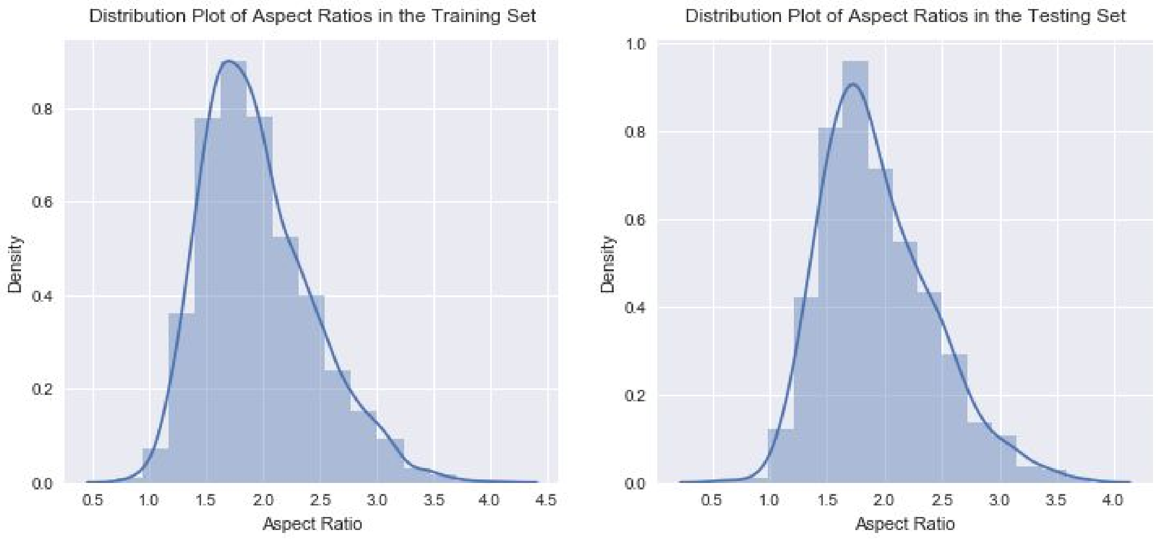

Global Thresholding (v = 127)
Adaptive Mean Thresholding
Original Image
Adaptive Gaussian Thresholding
Edge Detected Image
Otsu's Thresholding
HOG Image
Otsu's Thresholding w/ Gaussian Filtering

Figure 18: A Visual Exploration of Feature Extractions on a Sample Image

To start exploring feature extraction techniques to potentially reduce computation load and time costs for sklearn classifiers, Thresholding, Edge Detection and HOG features were analyzed. It is not apparent which method will work the best post downsizing, so only a visual comparison was performed. The downsizing phase can ultimately help reduce time costs that it takes to train different models. This can affect information demonstrated after these feature extraction approaches. Therefore, deciding on what image dimensions to use is a difficult task.

Train/Test Split Visualizations



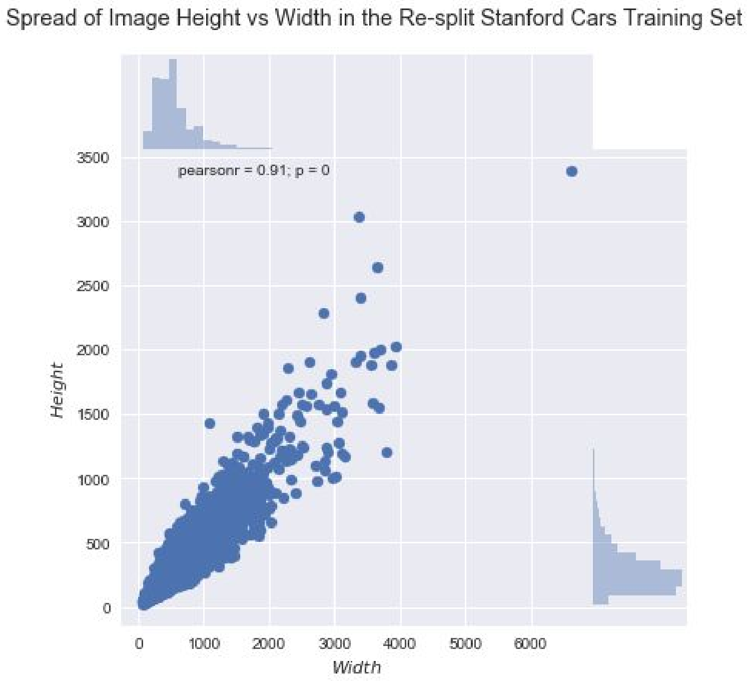
Distribution Plot of Aspect Ratios in the Training Set

Distribution Plot of Aspect Ratios in the Testing Set

```
The average image size in the entire original Stanford Cars Dataset is (573, 308)
The standard deviation of the image size in the entire original Stanford Cars Dataset is (375,  214)

The average image size in the Re-split Stanford Cars Training Set is (572, 308)
The average image size in the Re-split Stanford Cars Testing Set is (563, 303)
```

Figure 19: Train/Test Splits Show Similar Distributions of Image Aspect Ratios



Spread of Image Height vs Width in the Re-split Stanford Cars Training Set

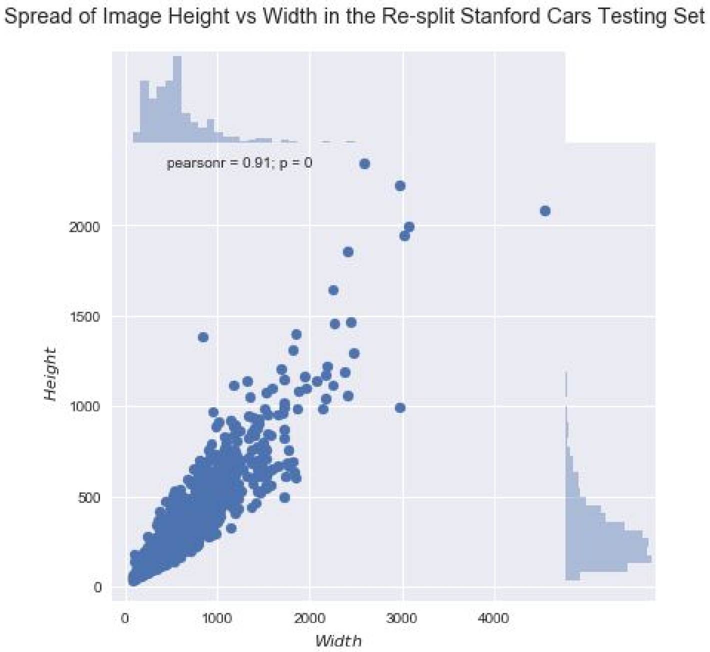


Figure 20: Train/Test Splits Show Similar Distributions of Image Heights and Widths

While the majority of the images have image dimensions less than or near 1000 by 500, there are many images with high image quality. There are also variations in aspect ratios consistent with the histogram of the distribution of aspect ratios. Further analysis on image resizing is required in order to alter the image shape to be suitable for sklearn/keras/tensorflow, which will be analyzed more in Milestone 3.

Summary of EDA Findings

After analyzing all car class attributes of the entire dataset, it became apparent that there were class imbalances and issues with the Year attribute. While class imbalances were expected due to the popularity of certain car makes and models, the distribution of car models in this dataset appeared to be not representative of what is commonly seen on the roads in the US. An example of this is how the images of Bugatti Veyrons outnumbering Ford Focuses. Since the business objective of this project is to create an image classification algorithm for users to be able to identify cars models found on US roads, these class issues may negatively affect model performance. Class aggregation issues from the data collection methods by the authors of the Stanford Car dataset also became apparent when comparing the distribution of car models across model years. Despite many models producing new car models every year, images of almost all car models in this dataset reside in one model year.
Based on the EDA of the entire dataset, the dataset was split into an 80:20 training-testing sets using undersampling. Thanks to undersampling, class imbalances across models were reduced significantly. Using the newly sampled data, image dimensions were analyzed. While most of the images had similar dimensions, there were many images that were significantly more high-quality, while others had very different aspect ratios.

Future Plans Outline

1. Process image data:
   a. Determine optimal image size
   b. Perform other necessary image manipulation to finalize image data to be used in all models
   c. Explore which feature extraction methods to use and do background research on how they work.
2. Determine algorithms and evaluation measures to be used for image classification:
   a. Find and discuss relevant literature pertaining to image classification
   b. Finalize approach to creating CNNs
   c. Decide which non-NN classifications to perform
   d. Decide who is doing which non-NN classification
3. Attempt to create initial models
   a. Non-NN and baseline CNNs: create first iteration of models
   b. Complex CNNs: read more literature and start coding
4. Find/create images to add to data set