Used Cars Recommendations Using AI Classification

Jadon Lai
CSC 466
California Polytechnic State University SLO
San Luis Obispo, USA
jlai43@calpoly.edu

Adrick Malekian
CSC 466
California Polytechnic State University SLO
San Luis Obispo, CA
amalekia@calpoly.edu

Andrew Cheung
CSC 466
California Polytechnic State University SLO
San Luis Obispo, USA
cheung2710@gmail.com

***Abstract*—With cars now being a part of everyday life, it's important to look at the options that people have for buying cars. One method that often works well and saves money is buying used cars. But with such a big market, it can be hard to tell what car make and model, horsepower, fuel efficiency, mileage, etc. a buyer should go for. We've used KMeans Clustering and Hierarchical Clustering algorithms to determine the commonly bought cars for certain price ranges, and their features so that the buyer knows what to expect within their price range.**

Keywords—KMeans clustering, hierarchical clustering, elbow method, dendrogram, complete linkage, single linkage

I. Background

In many American cities, cars are a near-mandatory method of transportation. Since we will be graduating in the next year or two, and we'll probably need to drive to work in the future, we were interested in finding groupings of cars that could point us in the right direction. For example, we hypothesized that certain car brands would tend to be expensive with high horsepower. We believe that our analysis of the data will provide useful insight into seeing what buyers should expect when going to buy a car within a certain price range, including mileage, horsepower, and fuel economy.

II. Previous Work

Previously, Jasmi Kevadia did an analysis [5] very similar to that of this project where she retrieved a used car dataset from Kaggle and analyzed some car qualities that might help dealerships determine a more accurate price for selling such a vehicle. Features such as mileage, price, transmission, fuel type, and body type were extracted and implemented in the k-means model to create 4 subsets of car classes.

Another study [4] conducted by Nour Chetouane, Lorenz Klampfl, and Franz Wotawa worked with car data but rather focused on the driving aspect of the data. They extracted information such as throttle control, steering inputs, and braking force from multiple cars driven on the road. So although they used clustering to classify different driving scenarios, they ultimately had the same goal to use information about cars to classify how certain vehicles in certain conditions react and handle.

III. Data

Initially, we struggled to find a dataset that had enough features and data points to form meaningful clusters. We chose a large dataset [7] from Kaggle, which contained 66 attributes of information about 3 million used cars in the United States. According to the description, the data was collected in September 2020 using a crawler on the website of CarGurus, an automotive shopping business. The dataset was highly relevant to our topic, but we needed to bring the file size down to something that we could process in Python. For the initial preparation, we removed features that didn't seem relevant, such as the car's exterior color, and features for which most rows had no data, such as cabin size for pickup trucks. We used the Pandas library to read the raw file in chunks of 10,000 rows, keeping the ten features that we were most likely to use and wrote the clean data to a new file. Although the removal of most unnecessary columns reduced the file size significantly, we encountered memory issues when we tried to cluster the 3 million data points. So, we took a random sample of 10,000 values and used that as our dataset instead.

IV. Methods

Since we had a large sample, we decided to use five features for clustering: price, horsepower, year, fuel economy, and mileage. These are some of the most important factors that a buyer would consider when shopping for a used car, and we hoped to point buyers towards a cluster based on what they want in a car. We chose to compare the results of k-means and hierarchical clustering because they are widely used and their outputs are easy to interpret. We standardized all of the features to ensure that the k-means distance function wouldn't be distorted by features with large numbers, such as price.

A. KMeans Clustering

To get the best results from k-means clustering, we considered how to handle outliers, the best way to choose initial centroids, and the optimal number of clusters. First, we made scatterplots with several combinations of features, and there were some clear outliers.

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

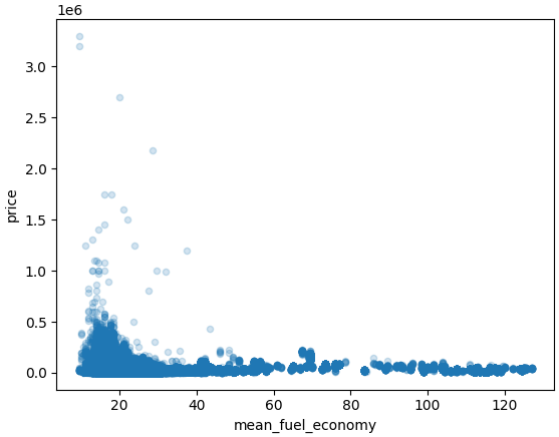

Fig. 1: Price (millions of $) vs. Fuel Economy (km/L)

Most of the cars were under $0.5 million, but some outliers cost over $3 million. Mean fuel economy was the average of the city and highway fuel economies from the dataset.

We found a research paper [6] showing that outliers can have a significant effect on the quality of k-means clusters. However, we saw that all the features had outliers, and we didn't want to remove so many extreme values, especially after having taken a random sample from the original dataset. So, we kept the outliers, keeping in mind that they could pull some data points away from more central clusters.

The standard k-means algorithm is known to sometimes produce poor clustering with randomly chosen centroids. We used the k-means++ initialization algorithm [3] to choose centroids more smartly because it was a convenient option in the SciKit KMeans function. The algorithm is as follows:

`input: k, the desired number of centroids; D, the dataset to cluster`

`output: C, a list of k centroids`

- `initialize the first centroid as a random point in D and add it to C`
- `while the length of C < k:`
  - `for each point in D, calculate its distance to the nearest centroid in C`
  - `choose a point in D such that points farther from their nearest centroid are more likely to be chosen, and add that point to C`

The algorithm favors centroids that are far apart from each other, which improves clustering quality and convergence time over random initialization.

To choose the optimal number of clusters, we used the Elbow Method, where the average distance of a point to its centroid is plotted against the number of clusters.

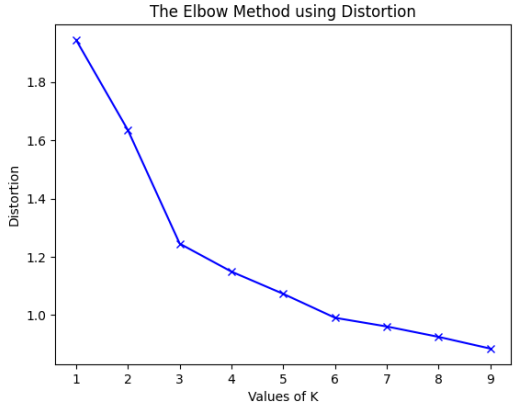

Fig. 2: Elbow Method

Distortion, the average of total squared Euclidean distances between the points and their centroids, was used as the error metric. The "elbow" is at k = 3.

The slopes of each segment in the Elbow graph show that the reduction in average distance falls off when the number of clusters increases from 3 to 4. So, we used k = 3 in our model.

After fitting our model to the data, we made multiple graphs to analyze our results. This includes scatter plots of the price vs. features, pie charts of the car make for each cluster, etc. The clusters appear to be well-balanced and provide useful insight, but we'll go over this in the results section.

B. Hierarchical Clustering

We created the dendrogram using the standard hierarchical clustering algorithm, where the closest two clusters are repeatedly merged until only one cluster remains. This dendrogram was made using a complete linkage, as the single linkage ran into depth errors, due to the size of our sample.

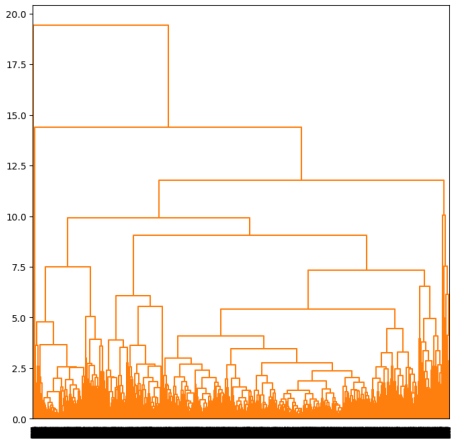

Fig. 3: Dendrogram

In the dendrogram of all the points, where we can visually see that if we 'cut' the dendrogram around y=12.5, we'll get our 3 clusters. However, we can also initially see that the clusters will not be even, with most of the points in one cluster.

The single linkage can be defined as a distance measurement between two clusters, where the distance between the two closest points in each cluster is taken. On the other hand, a complete linkage is the distance between the two farthest points in each cluster, so it's the maximum distance. These linkages need to be computed so that we can determine which subgroups to form and when inside of the dendrogram. Further information on this topic can be found on GeeksForGeeks [8]. Then, we cut the dendrogram into 3 clusters to match the k-means model.

Again, after fitting the model, multiple graphs were created so that we could see the results of our clusters. These clusters appeared to be less fruitful than the clusters from the KMeans model, but are still helpful, nevertheless.

V. Results

A. KMeans Clustering

Our k-means classification model was able to define 3 distinctive groups which tells us lots of valuable information about the subset of cars. These 3 clusters each have a good proportion of the cars in them, so there are not too few data points in any certain cluster.

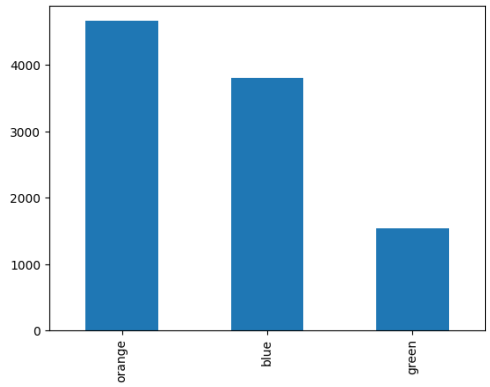

Fig. 4: Distribution of Cars in Each KMeans Cluster

Our model could classify cars with similar attributes (mileage, mpg, horsepower, year, and price) together appropriately despite having some outliers. Taking into consideration that this dataset consists of only used cars, we were able to see lots of interesting results when examining our main variable price.

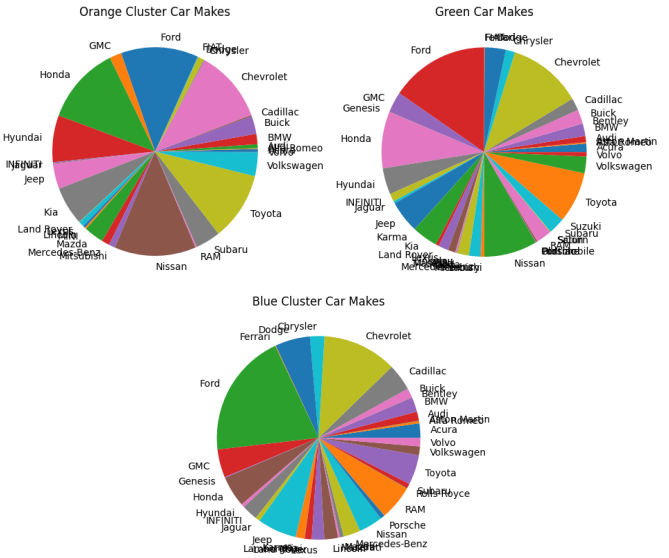

Fig. 5: Car Makes per KMeans Clusters

Looking at the k=3 different clusters, the model classified the blue cars as the more luxurious/expensive cars, and the orange/green clusters are a mix of sports cars and normal economical cars. The blue, orange, and green clusters car prices range from approximately $20,000-$300,000, $10,000-$80,000, and $5000-$50,000 respectively.

After analyzing the data, we can say that some more exotic car brands such as BMW or Mercedes fall in the same categories as those of more economical car brands such as Ford or Toyota. One reason for this is that certain factors such as mpg or horsepower might outweigh the price predictor variable more and thus cluster a similar car from both an exotic car company and an economical car company together even despite the extra price on the exotic car brand. However, the more predominant reason behind this phenomenon is the sharp decrease in certain exotic car values. This leads to a potential reason to not sell or purchase a used luxury car since you might lose a large amount of its initial value from that vehicle. Although the data is slightly biased towards Ford because there was more car data on that brand, it's hard not to mention that brands such as Honda, Nissan, Toyota, and Chevrolet are also among the most popular cars within the more affordable car price range of $10000-$50000. This says a lot about how these brands can hold value to their vehicles which can be an important factor to those investing in a vehicle.

According to CarEdge, the average depreciation of Toyota's best-selling car, the Toyota Corolla, is about 21% after 5 years. For Ford, their most sold vehicle is the Ford F-150 and that vehicle has a deprecation of 24% after 5 years. Cars that fall under the orange/green clusters might be cheaper than a brand-new exotic, however looking at long-term value, these vehicles tend to hold their value better than a luxury car brand.

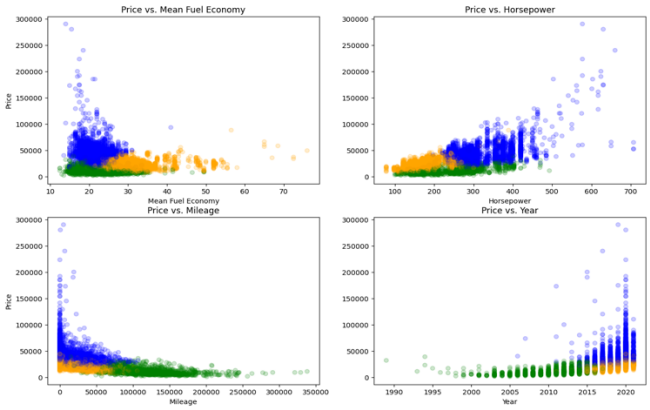

Fig. 6: Price vs. Features of KMeans Clusters

In the figure above, it is very notable that the expensive/exotic cars in the blue category tend to lose their value much quicker than those in the orange/green clusters whose values are at a plateau. Therefore, we can see luxury brands fall into the same cluster as economical cars due to their high depreciation rates even after considering things like mpg and horsepower. But examining the green cluster, we can see dependable brands such as Chevrolet, Honda, and Toyota that retain their value year after year despite having high mileage. As college students, we should be drawn towards the green cluster models of cars since we can foresee them retaining value so if we still want to sell the vehicle in the future, it won't be a detrimental loss, or the orange cluster if we have a slightly higher budget.

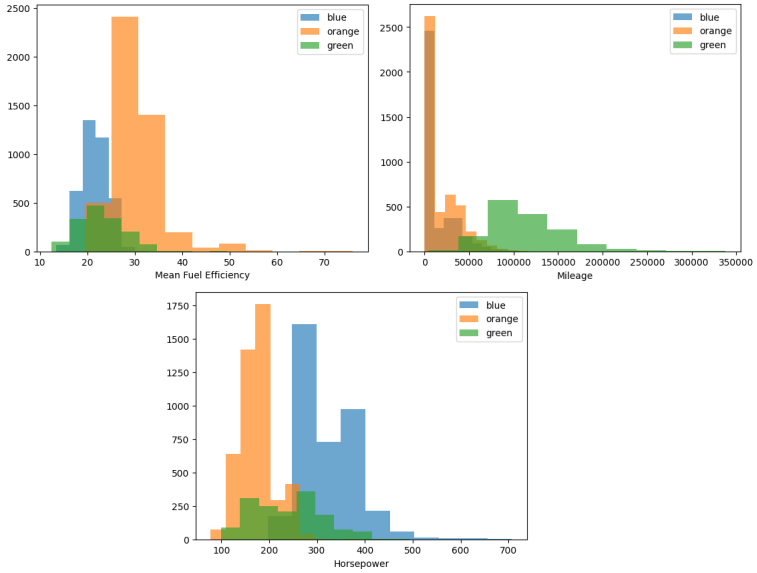

Fig. 7: Histogram of Clusters for Different Variables

The histograms for mean fuel efficiency, horsepower, and mileage only support our findings from the car make pie charts and the scatterplots of features vs. price. We see that the horsepower of orange and green cars tends to be lower than the blue cars (more expensive/exotic cars), along our suggestion of college students purchasing cars from the green category makes sense. The green cluster has a low mean fuel efficiency, medium horsepower, and high mileage, which are all "negative" features of a car that lead to a lower-priced car. The orange cluster is in between, with good mean fuel efficiencies, low horsepower, and low mileage, resulting in a good, reliable everyday car.

B. Hierarchical Clustering

Our hierarchical clustering created different clusters from the ones found in the KMeans clustering. These new clusters seem to have less variance since Hierarchical Clustering uses a more strict split whereas KMeans is a bit more flexible when it comes to other similar values.

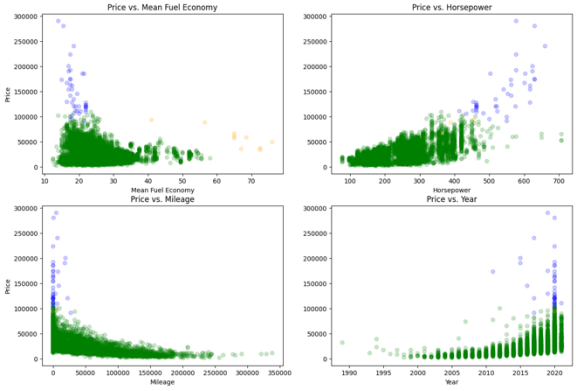

Fig. 8: Price vs. Features of Hierarchical Clusters

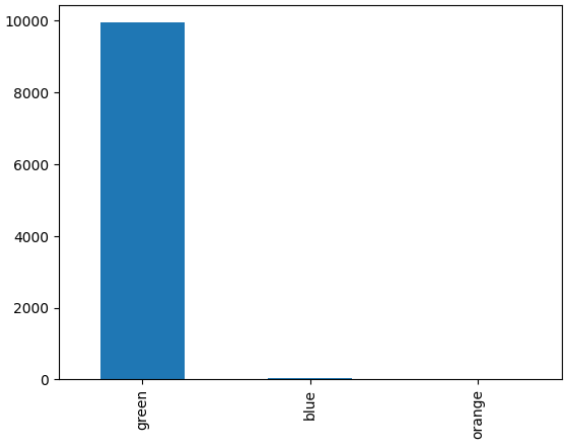

Fig. 9: Distribution of Cars in Each Hierarchical Cluster

The size of the green cluster makes the data a bit harder to read and shows where the cars are "consolidated," as is shown in Fig. 9. On the other hand, the small blue and orange clusters display the outliers that most people don't buy. These are the highly-priced sports cars, cars with extremely good fuel economy, etc. Overall the results of the Hierarchical Clustering model give similar information to that of the KMeans, however, KMeans is probably more useful for our analysis since the Hierarchical seems to mainly focus on the outliers of the dataset.

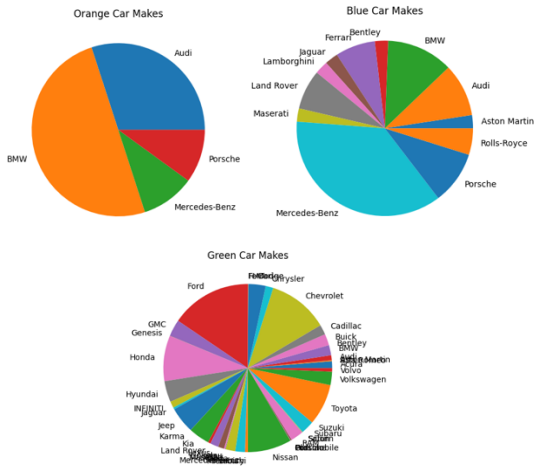

Fig. 10: Car Makes per Hierarchical Cluster

Notably one distinct difference between KMeans and Hierarchical clustering is the proportion of car makes for each cluster. Although the orange and blue clusters have much fewer data points than the green cluster, it's evident to see that the most common cars for the blue cluster are Mercedes-Benz and the most common for the orange cluster are BMW. This implies that Mercedes-Benz tends to make higher priced sports cars which retain their value better, while if you're looking for a car with a higher fuel efficiency and cheaper used car prices, it'd be a good idea to look at BMW. The reason that the green cluster has such a diverse distribution of car makes is because it has so many data points, but we can still see that Ford, Chevrolet, Honda, and Nissan make the most "average" cars.

VI. Conclusion

In conclusion, we can say that our KMeans model effectively gives a consumer the best 3 ranges of cars they can pick from when purchasing a new car in the US. One can make many judgments when deciding what vehicle to purchase, however, this model gives us a more logical insight into how important factors when purchasing a vehicle can essentially categorize it based on other vehicles with certain aspects outweighing others. While producing this model in the eyes of a college student, we tried to take factors such as budgets and usability when selecting our most desired variables. By viewing the clusters among the different graphs of features, we can tell the types of cars that a college student should look at and what to expect when buying a car. Of course preferences and personal opinions are not taken into consideration when choosing a vehicle from one of these subsets of vehicles, however, this model serves its purpose of educating someone who might not have much knowledge of what kind of vehicle could be particularly valuable or usable for their needs.

This research could be furthered by looking into new cars, using multiple datasets, and/or using different classification methods. There is also the option of fitting regression lines onto the data to try to predict what the price (or other feature) of a car will be, given certain parameters. This paper is meant to be a simple analysis of clustering cars from one dataset to see what college students should look for and expect when buying a car.

References

[1] "Ford F-150 Depreciation," *caredge.com*. https://caredge.com/ford/f-150/depreciation (accessed Mar. 03, 2024).

[2] "Toyota Corolla Depreciation," *caredge.com*. https://caredge.com/toyota/corolla/depreciation

[3] "ML | K-means++ Algorithm," *GeeksforGeeks*, Aug. 19, 2019. https://www.geeksforgeeks.org/ml-k-means-algorithm/

[4] N. Chetouane, "Extracting information from driving data using k-means clustering (S)," *Proceedings of the 33rd International Conference on Software Engineering and Knowledge Engineering*, Jul. 2021, doi: https://doi.org/10.18293/seke2021-118.

[5] J. Kevadia, "Cluster Analysis of a Used Vehicles Dataset," *INST414: Data Science Techniques*, May 12, 2023. https://medium.com/inst414-data-science-tech/cluster-analysis-of-a-used-vehicles-dataset-cdbe6988ff82 (accessed Mar. 03, 2024).

[6] A. Nowak-Brzezińska and I. Gaibei, "How the Outliers Influence the Quality of Clustering?," *Entropy*, vol. 24, no. 7, p. 917, Jun. 2022, doi: https://doi.org/10.3390/e24070917. A. Nowak-Brzezińska and I. Gaibei, "How the Outliers Influence the Quality of Clustering?," *Entropy*, vol. 24, no. 7, p. 917, Jun. 2022, doi: https://doi.org/10.3390/e24070917.

[7] "US Used cars dataset," *www.kaggle.com*. https://www.kaggle.com/datasets/ananaymital/us-used-cars-dataset

[8] "ML | Types of Linkages in Clustering," *GeeksforGeeks*, Jul. 19, 2019. https://www.geeksforgeeks.org/ml-types-of-linkages-in-clustering/