אוניברסיטת אריאל בשומרון

INVESTIGATING VEHICLE RECALL TRENDS AND CHARACTERISTICS

Course: Data Mining
Lecturer: Chen Hajaj
Team Members: Yogev Ladani & Amit Fallach

https://github.com/YogevLD/Advanced-Machine-learning.git

Abstract

Despite advancements in automotive technology, vehicle recalls continue to occur regularly due to various factors such as manufacturing defects, design flaws, and safety regulation violations. The purpose of this study is to investigate the relationships between different characteristics of vehicles and circumstances related to their removal from service, with the aim of gaining insights into factors influencing vehicle usage patterns, maintenance needs, and safety issues. Additionally, the study seeks to explore anomalies or unusual events that may occur on specific dates, contributing to a deeper understanding of underlying data patterns and real-world implications. These insights have broad implications for consumers and the automotive industry, potentially informing smarter purchasing decisions and enabling manufacturers to refine their products and services accordingly. The research methodology involves data collection from data.gov.il via an API call, followed by preprocessing steps such as removal of irrelevant columns and data cleaning to ensure data quality. Subsequently, dimensionality reduction techniques including PCA and Kernel PCA are employed to explore the data's structure and identify underlying patterns. The study concludes by analyzing the results obtained from clustering algorithms such as K-means and hierarchical clustering, providing valuable insights into the characteristics and behavior of vehicle data clusters.

Introduction

Despite advancements in automotive technology, vehicle recalls continue to occur regularly due to various factors such as manufacturing defects, design flaws, and safety regulation violations. These recalls not only inconvenience consumers but also raise questions about the reliability and safety of vehicles on the market. Consequently, there is a growing interest in understanding the underlying factors contributing to vehicle recalls and their impact on vehicle usage patterns, maintenance requirements, and safety issues.

The purpose of the study is to investigate the relationships between different characteristics of vehicles and circumstances related to their removal from service. By uncovering these relationships, we seek to gain insight into factors that impact vehicle usage patterns, maintenance needs, and possible safety issues. Additionally, we aim to investigate anomalies or unusual events that may occur on specific dates, contributing to a deeper understanding of the underlying data patterns and potential real-world implications. These consequences can affect anyone who purchases a car or works in the automotive industry. If we can identify connections between different characteristics of vehicles and future breakdowns in advance, we will be able to make smarter purchasing decisions, while companies can refine the relationship between vehicle characteristics and breakdowns to improve their products and services.

Dataset and Features

We used a simple API call to read data from data.gov.il. Our data is updated once every 24 hours according to the new vehicles that receive recall calls in Israel.

The initial feature list consists of 25 columns and 30,000 rows. The contents of the data columns are described in figure 1.

columns summary:

1. `mispar_rechev` : Numeric column representing the vehicle's number.
2. `tozeret_cd` : Numeric column representing a code associated with the vehicle's manufacturer.
3. `tozeret_nm` : Text column representing the name of the vehicle's manufacturer.
4. `degem_cd` : Text column representing a code associated with the vehicle's model.
5. `degem_nm` : Text column representing the name of the vehicle's model.
6. `sug_rechev_cd` : Numeric column representing a code associated with the vehicle's type.
7. `sug_rechev_nm` : Text column representing the name of the vehicle's type.
8. `moed_aliya_lakvish` : Text column representing the date of vehicle's import.
9. `bitul_dt` : Text column representing the date of vehicle's cancellation.
10. `misgeret` : Text column representing the vehicle's classification.
11. `tozar_manoa` : Text column representing the manufacturer's code.
12. `degem_manoa` : Text column representing the model's code.
13. `mispar_manoa` : Text column representing the vehicle's number within a model.
14. `mishkal_kolel` : Numeric column representing the vehicle's weight.
15. `ramat_gimur` : Text column representing the vehicle's assembly status.
16. `ramat_eivzur_betihuty` : Numeric column representing the vehicle's insured value.
17. `kvutzat_zihum` : Numeric column representing the assembly group.
18. `shnat_yitzur` : Numeric column representing the production year.
19. `baalut` : Text column representing the vehicle's condition.
20. `tzeva_rechev` : Text column representing the vehicle's color.
21. `zmig_kidmi` : Text column representing the internal fuel system.
22. `zmig_ahori` : Text column representing the external fuel system.
23. `sug_delek_nm` : Text column representing the name of the fuel type.
24. `horaat_rishum` : Numeric column representing the registration status.
25. `kinuy_mishari` : Text column representing the vehicle's registration mark.

figure 1

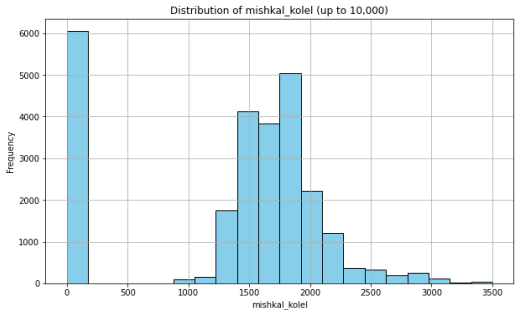

figure 2

During the preliminary research we discovered that the columns 'misgeret', '_id', 'mispar_rechev' and 'mispar_manoa' contain single-valued values without meaning. Since there is no way for us to use them in the process, we decided to remove them.

In the pre-processing, we started by checking whether there are duplicates in the data or the meaning of the features. After researching the background of the data on the Internet we discovered that there are some pairs of columns that a certain value in one column will always result in a fixed value in another column. For example, if in the degem_cd column the model value is "0365", the value in the degem_nm column will be "7EMH49". The pairs are:

('tozeret_cd', 'tozeret_nm')

('degem_cd', 'degem_nm')

('sug_rechev_cd', 'sug_rechev_nm')

('moed_aliya_lakvish','shnat_yitzur')

After verifying the match between each pair of columns according to the background behind the data (via gov.il) we wanted to make sure that there were no rows where incorrect information was entered and that there really is a perfect match if not very high (90%+).

We created functions whose task it is to determine whether every pair of columns actually matches completely. In each pair where we got a 95% match or higher, we selected one of the columns and removed the other. We knew that if we got less than a 95% match, there was probably an error in the data entry (something that turns out to be common in this database), so we removed the rows that did not match. Because we knew in advance that there was no unusual phenomenon behind the errors in the data, we chose not to investigate rows without matching (again, data entry errors are known in this database).

The degem_cd column had a lot of empty values. We chose not to remove the rows with the empty values in order not to dilute the data and to rely on the information that exists in the degem_nm column.

Another pair of columns that showed a match of 85% or higher were 'zmig_ahori', 'zmig_kidmi'. Because there is a high but not perfect match, we chose not to remove the lines where there is no match, but to investigate the lack of match as a motif for recall errors later on. It is very logical that different types of tires in the front and back can affect vehicle breakdowns. On the other hand, we had a lot of empty values in the 'zmig_ahori' column. We chose to fill them in accordance with the existing values in the 'zmig_kidmi' column and rely on the fact that 85% or more of the data is found with identical tires in general.

We split 'bitul_dt' column into the 'bitul_year', 'bitul_month' columns thinking that maybe later on we can identify trends by range of years or by periods during the year.

In 'kinuy_mishari' column we encountered a lot of missing values, this column was very important because it represents the known name of the vehicle in the market (for example Octavia, Corolla, A3, etc.). This column is critical for later cluster division and conclusion-making. We chose to fill in the common value for each make (Skoda, Toyota, etc..) according to the 'tozeret_nm' column. Following that, we deleted the lines that we were still unable to fill in due to the short number of lines and the fact that, in the absence of more examples, we were unable to identify the vehicle's commercial name.

After that We found many empty values in the 'ramat_gimur' column. We were aware that this column would be important because different finishes have an impact on the vehicle's initial cost and other features that are significant to both the manufacturer and the customer. To handle those empty values we decided to check what is the most common value for each year of production, manufacturer and trade name of the vehicle ('shnat_yitzur_and_aliya_lakvish', 'tozeret_nm', 'kinuy_mishari') and fill them in respectively. After we managed to fill in over two-thirds of the missing values, we removed the rows that we still couldn't fill in because we have no way of knowing what the corresponding finish level was for that row and we didn't want to create more bias in the data.

We then turned to handle the 'mishkal_kolel' column. We knew beforehand that we didn't have many numeric columns, so it was important for us to understand the values inside the column in order to use it later. We discovered that this column also has a lot of data that was entered incorrectly with incorrect and illogical values. For example, you can see that there are many rows whose total weight is zero (in figure 2).

First of all, to handle values equal to zero, we built a function that fills in the average weight (without counting the zeros) for all vehicles of the same manufacturer and of the same type.

After we finished handling the zeros in the column we noticed that there are many types of values in the 'sug_rechev_nm' column. Since the rest of our research process involved segmenting the 'mishkal_kolel' column according to each type of vehicle and we knew we didn't want too sparse data for each category, we chose to define more general categories in another column for vehicles ('vehicle_category'). We sunned regular expressions to divide the rows into new categories (eg 'Van \\ Jeep', 'Private Vehicle', 'Taxi', 'small Truck', 'Mini-Bus') according to the column 'sug_rechev_nm. During some of the times we called the data, additional categories such as motorcycles, large trucks and tractors came up. We adjusted the code that will divide each type of vehicle that arrives and define a general category and association according to regular expressions. It should be noted that due to the dynamism of the data, not all categories are always repeated. That's why it was important for us beforehand to adapt the solution to all the options.

After the pre-processing we moved to encoding. We coded the data according to a "binary encoder" because none of the categorical columns had an ordinal order. For the rest of the numerical columns we defined scaling according to StandardScaler() to avoid ranges that are not limited as in MinMaxScaler.

After that we debated whether to start with dimension reduction or with treatment of abnormal values. After several attempts, we saw that dropping the exceptions in advance brings us much more successful results in reducing dimensions.

We tried several methods of removing outliers, first of all we checked if we have a normal distribution for the numerical columns or for the categories from these columns (for example the weight of the vehicle according to the type of vehicle). The test yielded negative results for a normal distribution (for example the histogram in Figure 3) therefore we ruled out the use of Z score.

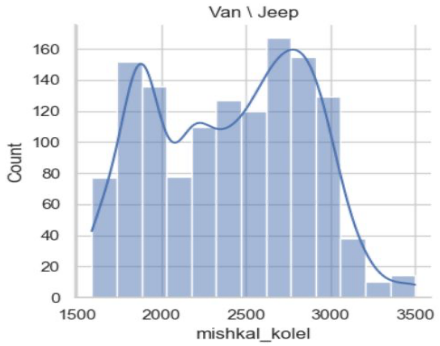

figure 3

Also, we tested LOF and decided not applying it for two reasons - the first is that it is very sensitive to noise. And our data is dynamic and includes different noises each time (some of the noises are justified because of unusual data), in addition the LOF gave us a lot outliers in relation to the data set itself (almost a third of the data).

Finally, we implemented an isolation forest algorithm that gave us the most logical result, so according to the result we removed outliers.

We proceeded to reduce the dimensions after completing the processing of the outliers. In order to compare various approaches, we also used clustering when lowering the dimensions; for this reason, we will discuss both the dimension reduction and the clustering in the methodology section.

Methodology

In this section we will describe the methods we tried to reduce dimensions and the logic behind it. After that we will connect the work process of reducing the dimensions together with division into clusters.

When we approached the downsizing phase we knew we had reached the bottleneck of our project. Our data is mostly categorical with lots of outliers. We wanted to try a variety of methods, both to learn and to examine situations that we would not have thought of beforehand.

In the next chapter of the discussion we will detail the selection of the parameters we will use in the current chapter (silhouette score, number of clusters, scaled inertia, etc.).

PCA

At first we tried PCA even though we knew it only captures linear relationships and that it has difficulty in high dimensions. In the output we got 50-52 new features that explain 90% of the variation. We performed clustering with Kmeans for the data (K=5) before and after PCA to see if the silhouette score changed and we saw that it did not deteriorate, which means we were able to maintain the same level of separation in the data despite reducing the dimensions. We chose the K in the division into clusters with scaled inertia (we will detail the method for choosing the parameter in the Discussion because of the project guidelines). Then we presented the result in T-sne to graphically illustrate the division into clusters. The clusters in the graph turned out to be mixed and undefined, as can be seen in Figure 4 (we also tried different numbers of clusters without success).

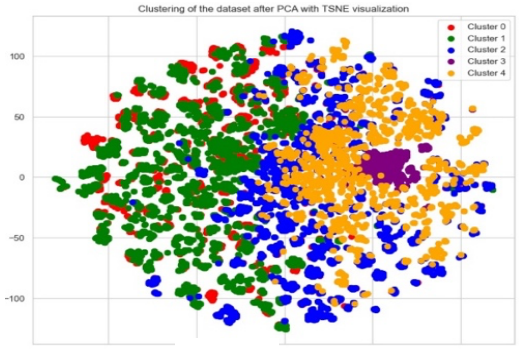

figure 4

It can be seen that graphically there are almost no defined clusters at all (silhouette score came out 0.074343). The PCA results did not satisfy us even though we reached a relatively high percentage of explained variance. We knew that PCA could mislead us because it fails to truly describe the relationships behind it. We left the process in the notebook for 2 reasons, the first is because we wanted to show the direction of thought and the second is that if we had an unusually good result we might be able to divide into clusters after using the algorithm

Karnel PCA

Because of the low values in PCA we decided to try methods that capture non-linear relationships. We started with Karnel PCA. When we graphically presented the output we were unable to draw a final conclusion. For several days we ran the test each time on new data and saw that there are large vectors in the same direction regularly - meaning there are main and constant components that capture the variation (Figure 5).

To check in more depth the nature of the dimension reduction, we also checked here the data after dimension reduction with Kmeans and then presented the clusters with TSNE. We used scaled inertia again to choose the optimal K. Despite the advantages of the scaled inertia, we did not see a convergence to an absolute minimum point, the silhouette score values of all divisions in the range 2-6 were relatively high and about the same (0.75-0.78) .therefore we decided to try K=2 and K=4 and visually choose the best of them

When we presented the TSNE of K=2 we reached a relatively clear result. It can be seen that there are 2 relatively separate clusters but not of the same size (Figure 6) therefore we decided to continue with K=2 and with the hypothesis that the data is divided into 2 clusters. Later we will try to strengthen the hypothesis using hierarchical trees. Because the results were to our satisfaction, we chose to continue with the data after reducing dimensions with PCA kernel.

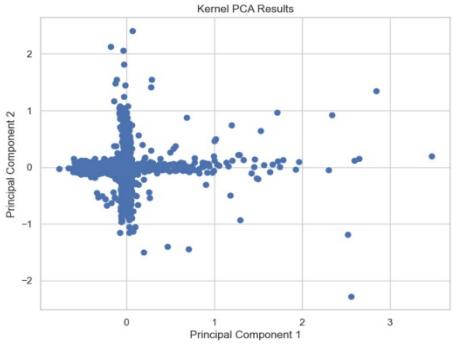

figure 5

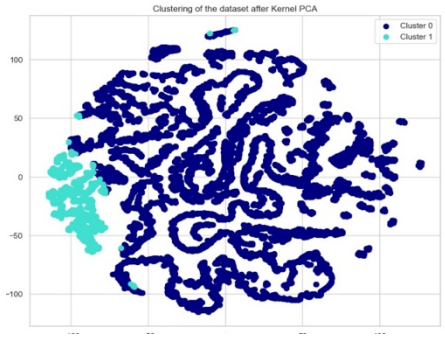

figure 6

To strengthen our claim that the data is divided into 2 clusters we used hierarchical trees. We wanted to examine the data without creating a bias and in addition to minimize the variation therefore we used Wards Linkage. In the division itself (Figure 7) it can be seen that there is a large vertical distance between the division and 2 clusters in contrast to the divisions that come after that show a less distinct separation (smaller vertical distance). Therefore, it can be said that the hierarchical trees also contribute to our claim that there are 2 distinct clusters in the data (the right side is divided in an unbalanced way and did not always repeat itself in all iterations, so we chose to refer to the division into 2 clusters and not for more).

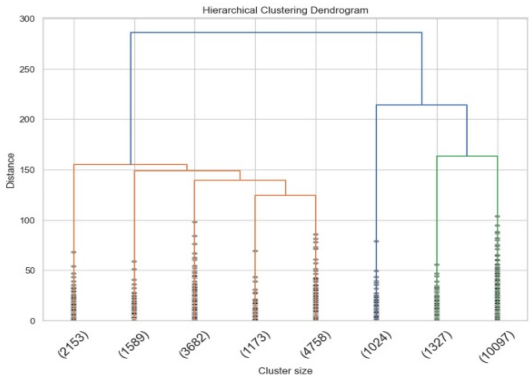

figure 7

Results & Discussion

In this section we will describe the final results and the choice of parameters that led to these results throughout the process. The project mainly deals with the division into clusters, therefore the main parameters we had to choose represent the quality of the division and the density of the clusters.

During the clustering we had to choose the number of clusters K each time we implemented Kmeans. To choose the optimal K we used the Scaled Inertia meter. The estimator calculates the inertia value for each possible K and weights it according to a penalty parameter (Figure 8). This parameter represents the trade-off between the inertia and the number of clusters.

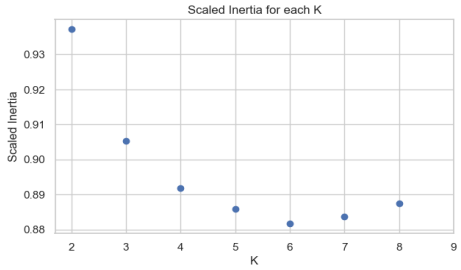

figure 8

The silhouette score - serves as a valuable measure if the main goal is to assess the separation and uniqueness of clusters. Higher silhouette scores indicate well-defined and separated clusters. During the project we tried to implement several types of dimensionality reduction algorithms and tested their result using Kmeans. In PCA we got a very small silhouette score (0.073) despite a high percentage of explained variance. Using the low silhouette score we decided that PCA was not enough for us. The result of the silhouette score helped us understand that we might be missing non-linear relationships in the data and thanks to it we decided to try PCA Karnel. After we tried to use clustering in PCA Karnel we got a relatively high silhouette score (0.76). This result helped us to reach the decision that the use of PCA Karnel is the optimal of the 2. During the procedure, we looked at two approaches to dimensional reduction: the linear approach (PCA) and the non-linear approach (PCA Karnel). We tested the PCA Karnel after noticing that the results of the PCA silhouette score were really low. When we developed it, we debated between a few different kernel functions. We could see that a non-linear function would be required. We looked at the results using the sigmoid function, a polynomial kernel, and RBF. Based on the quality of the results, we chose to stick with the sigmoid function. We believe that this option was logical as Tel presents a multitude of features that might be useful as explanatory factors in the future when determining whether or not to engage in the manufacturing or purchase of a vehicle (binary decision).

After realizing that our data is divided into 2 clusters that are not equal in size, we began to investigate the values in terms of descriptive and numerical data. We took into account that the small cluster could be a cluster of anomalous .data that would logically remain in the data (this logic continued with us from the anomaly detection stage)

We built a function that shows the quantity of each value from each categorical column in each cluster. We saw that the 2 clusters have relatively the same amount in relation to the initial amount in the data from each category in all the columns. Therefore, the first conclusion we reached is that there is no difference in terms of categorical data

between the clusters. After that we moved on to the examination of the numerical data. In the numerical columns we reached some important conclusions that helped us decide on the more accurate definitions for each cluster.

In the large cluster (cluster 0) there were vehicles with medium-high weight both in terms of average values and in terms of the end of the range of values, unlike the small cluster (cluster 1) where there were vehicles with low-medium weight (Figure 9).

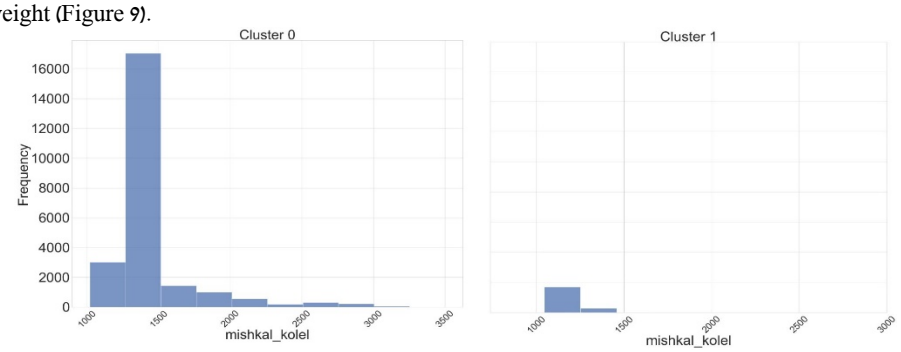

figure 9

In the large cluster, the time on the road was approximately 12 years, whereas in the small cluster, the years were 17 years. The years of production and going on the road in the small cluster were in the range of 1996-2010 as opposed to the large cluster where they were in the range of 1996-2024 (figure 10).

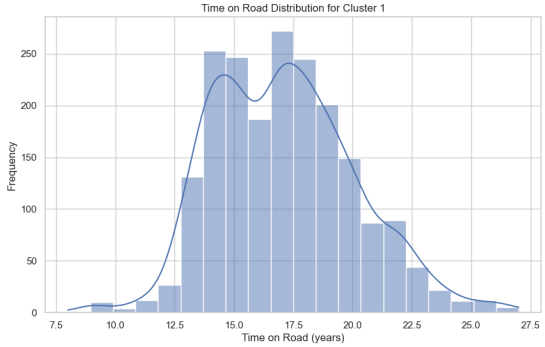

figure 10

From this information, we inferred the following details for the little cluster (cluster 1):

The cluster includes little part from the data (about 2000 in each iteration) so it may also be abnormal data. The information relates to cars that were driven up until 2010. Despite this, it can be seen that vehicles in this cluster survived a long time on the road until the malfunction that disabled them. The vehicles in this cluster are vehicles that weigh less, when we looked at their types we saw private vehicles on the border of mini vehicles (we did not show a summary figure because it changes all the time).

This leads us to believe that small- to medium-sized cars that are put on the road these days are most likely less prone to malfunctions and repair needs. The consumer may find this conclusion useful in determining whether to buy this kind of car. The manufacturer can use this finding to determine whether to continue producing cars that are comparable to those in this cluster.

Conclusion and Future Work

In this project we investigated a very critical area for anyone living in today's western world. In the end, almost every adult drives a car and we all want to drive a car with the maximum lifespan. In the research we were able to reach conclusions and divide the data into 2 clusters. One large cluster that includes vehicles of a wide variety of types and a small cluster that includes a type of vehicle that did not appear in the large cluster, these vehicles were more durable and reached a higher range of years on the road. If we had more data or more features, we could continue to investigate the reasons for taking it off the road and maybe even build a recommendation system that would define for everyone who uses it what the vehicle is or what the optimal vehicle category is according to their needs in relation to the amount of money they are willing to invest. This recommendation system could also help the manufacturer decide which type of vehicle to produce in accordance with its business goals.

Contributions

We invested a lot in the project, we worked on the project for two weeks and along the way we consulted each other to learn. We made sure that there would be an orderly division in the work method because of the project guidelines, but we are both well versed in all the steps. The division of labor was as follow:

Amit did the cleaning and preparation phase of the data including building the functions to check compatibility between the columns, in-depth research on the meaning of the data and the way they are entered into data.gov.il, adjustment of the code all the way to a general and non-specific solution due to the dynamism of the information and encoding of the data.

Yogev also contributed at this stage some of the functions that arranged the 'mishkal_kolel' column and helped to think of solutions to correct the problematic data in this column.

It is worth noting that we both invested a lot in the 'mishkal_kolel' column because we saw its future significance in the clustering process.

In the dimension reduction phase, Yogev built the functions of the PCA and the PCA karnel while doing an in-depth study of the meaning of each parameter. A colleague contributed at this stage the elbow method according to the scaled inertia and the use of Kmeans to compare the results.

We decided the work method at that stage together after several different attempts, in order to organize the data we decided that for the 2 methods of reducing the dimensions we would do exactly the same steps to illustrate the work methodology. At this point Yogev also added TSNE.

Then, in the clustering phase, Yogev implemented Hierarchical Clustering to compare his results with the results of Kmeans. We discussed the results together and decided together on all the conclusions.

Amit wrote the report, and Yogev made the presentation. We had really good teamwork.

Appendices

https://towardsdatascience.com/an-approach-for-choosing-number-of-clusters-for-k-means-c28e614ecb2c