**Capstone 1 – Milestone Report**

**Background**: The National Highway Traffic Safety Administration has an Office of Defects Investigation (ODI) with a public online form on which consumers can log complaints about their vehicles regarding safety or quality. These complaints are categorized (by the consumer) based on what system of the car they pertain to, e.g. Electrical or Engine and Engine Cooling. Additionally, fields of information logged in the database include: the make and model and Model Year of the vehicle, a date for when the incident was logged in the database, whether a fire/crash/injury or death occurred because of the incident, where the incident occurred, and a most importantly a full paragraph description of the event that took place (written by the consumer).

NHTSA employees review these data on a frequent basis in order to monitor the quality of vehicles on the road. If complaints are logged by consumers that describe dangerous safety issues or if a high volume of complaints is found pertaining to one issue, NHTSA could take further action and even recall a fleet of vehicles. Because of the large cost incurred by vehicle manufacturers when a recall takes place, it is in a car company’s best interest to anticipate these quality issues and take action before NHTSA forces them to. Thus, processing these complaints for vehicle OEMs (Original Equipment Manufacturers) is a valuable process. Manufacturers would like to understand on a detailed level what issues are occurring with their vehicles. With the correct understanding of the types of issues prevalent and their frequency, they can issue warranties or campaigns before they are forced to by NHTSA to address their customers’ concerns.

**Problem**: Because of the high volume of complaints in the NHTSA ODI database, manual review, while possible, is not feasible unless an entire team is tasked with the job. For one manufacturer, like BMW, there are about 2500 individual complaints that are submitted via ODI every year. BMW has relatively low production numbers compared to manufacturers like Ford or GM, so reading complaints one by one and applying a category or complaint type to them can quickly become tedious. Unsupervised learning, specifically clustering, will be used to reduce the amount of time needed to process these complaints for a specific OEM by grouping like complaints together. Furthermore, Latent Dirichlet Allocation will be used to reduce a complaint to a few intelligible words (topics) to reduce the dimensionality of the text complaints in our dataset. The aim of this process is to apply a defect/complaint type to every complaint in the database. The clustering will be used to group like complaints together and if LDA is successful, provide keywords to be used in the “defect type” assignment. Human inspection will verify the groupings and then the model will be adjusted as necessary. If the clustering model is able to group even 80% of the complaints into categories with the right resolution, it will be extremely useful in the overall process.

**Business User**: Vehicle manufacturers (in this case, BMW) and engineering quality departments at these respective OEMs would be specifically interested in this type of analysis because of the ability to save them millions of dollars with recalls. If it is possible to better process user complaints and predict when their volume will swell and when NHTSA will issue a recall, they can take measures before they are forced to by NHTSA. Usually, when manufacturers are mandated by NHTSA to address an issue, the fleet that they must recall is much wider than the fleet specified by an internal recall or service bulletin. Thus, keeping all action within the company usually results in lower costs.

**Data**: Because NHTSA is a government organization, all of the complaints logged into the ODI database are public and available for download via flat files. These flat files are uploaded daily and the new complaints are appended on to the end of the file.

The data can be found at: <https://www-odi.nhtsa.dot.gov/downloads/> as well as instructions for viewing. Please see Appendix A under “Import Instructions” for explanations of data fields used in this document.

A new flat file is uploaded daily to the NHTSA website and that file is downloaded via ETL process daily and saved in a PostgreSQL database. Thus, a table is created for each day’s file and retained. These files are used as a version check because with new complaints that come in on a daily basis are appended to the preceding day’s file and re-uploaded by NHTSA. For this project, only BMW complaints are being used initially, but the analysis will be expanded to other manufacturers as well once the initial model is created and tested.

A few methods of cleaning are necessary in order to format the data properly. The data does not have full VINs of the vehicle complained about by a specific customer. The VIN column in the data is supposed to be the first 11 digits of the full 17 digit VIN. However, while the majority of the VINs retained are the first 11 digits, a portion of the VINs are nulls, the last 11 digits of the VIN, or another assortment of characters. These are all retained and what is usable is employed in a later step for mapping extra fields of data used in the analysis.

Additionally, with text fields such as mfr\_name, maketxt, and modeltxt, the fields are not often standardized. Because the fields on the web form that receive this data used to be text fields, not drop down menus (like currently) that limit the selection to specific entries, a person in previous years could enter a VIN that maps to a specific make and model, or enter it in manually. This created many inconsistencies. For example, a person can choose to enter their car as a BMW 3-series, or be more specific by saying it is a BMW 335. They may also leave out other details; it could be a 335i or a 335xi (all-wheel drive) and they may choose to enter this option as either 335ix, 335xi, 335 x-drive or any other potential name. Using all these combinations to create subgroups of models becomes very messy. It was also possible for the customer to enter the VIN in incorrectly by skipping a character or replacing it with something else. Because of the possible inconsistencies, VIN decoding is used instead of the multiple fields in the flat file. The first 11 digits of the VIN provide enough information as to the make and chassis code of the vehicle. The model year is also gained from other fields. Using a script with multiple conditions for decoding the VIN gives us our cleaned model name and although it is not perfect (100% of the VINs are not usable), it is much easier than trying to standardize every model name. Model names can be used as an additional step, after the VIN decoding. Manufacturer names have a similar inconsistency. Sometimes, in the data, BMW is used, or its German name Bayerische Motoren Werke. Both have to be used to narrow down the group of complaints. A similar problem was noted with Model Year: if the customer does not know the model year of their vehicle, they have the option of entering it as “9999”. For this, we must also use VIN decoding (http://www.autocheck.com/vehiclehistory/autocheck/en/vinbasics), but when an incorrect VIN is provided, we have no option to apply these fields to the data. Because our main concern is the narrative text of the complaint (cdescr), with additional features such as the make, model, model year, fires, crashes, injuries, deaths, complaint type and component description, these fields were prioritized in cleaning.

It is also possible for a person to log a complaint with the same narrative text, but with multiple component descriptions. For example, if a person wants to complain about their air-bag seat detection mat because it is not working, they may log the complaint under the following fields: AIRBAGS, SEATS, SEAT BELTS, and ELECTRONICS. This creates four complaints in the data set that all have the same odino (ODI number), however, they are four different rows. Thus, the distinction has to be made between unique odino versus unque odino AND compdesc (component description). Also, this makes it necessary to apply our own index for each row and not rely on the CMPLID (NHTSA’s unique ID which as they say, could “potentially change from one output file to the next”). When training the clustering model this will be imperative to differentiate between. Additionally, it was observed that because the customer selects these fields, they are not always correct. A customer may select fields like in the example above because they are somewhat related to the problem even though from an engineering perspective, the occupant detection mat is only in the SEAT and thus only relevant to that component description. A customer may also select multiple fields because they think that by doing so, it will escalate the severity of their complaint and get more attention on their issue from NHTSA. This creates some text descriptions that are falsely matched to a component description. This may introduce some error if they are used as a feature to train on in the clustering model. If not, and only the narrative text is used, it may prevent possible mismatches.

Other datasets that could be used are also listed at the aforementioned link. There are respective files for recalls, technical service bulletins as well as defect investigations. These sets contain different data, but still describe issues prevalent with vehicles.

**Initial Findings:** The analysis was originally undertaken with the intention of employing K-means clustering for groupings. K-means was seen as a viable solution because it is a relatively easy algorithm to implement and K can be optimized based on the dataset used.

The preliminary analysis is documented in the notebook voq\_clustering.ipynb and employs nltk’s standard list of English stopwords as well as the Snowball stemmer for preprocessing. Afterwards, tokenization is used to split up the words in each document and after all is said and done, about 1.4 million words remain in the entire corpus. A TF-IDF matrix is then built eliminating words that appeared in more than 80% of the documents and less than 5% of the documents.

K-means was then run on the documents and the elbow method was used to determine an optimal number of clusters for the given dataset. From manual review, there are about 600 different complaint categories (defects) that are assigned to this dataset. We ideally would like to mimic that level of categorization to minimize manual review as much as possible. However, from the manual review process, we know that there are many complaints that fall into a few dozen categories, and the vast majority of categories only have a few complaints in them. The figure below describes this phenomenon.

Figure 1: A large categories with the majority of categories only including a handful of complaints

That said, it was necessary to understand what K-means would determine from the data. Perhaps a more ideal way of grouping is inherent in the data that is not currently employed by manual coders. The elbow method was used to determine this and in order to find the ideal K, an inflection point would have to be determined. A first run at the elbow method used a K between 50 and 450 (very wide range). The plot is shown below.

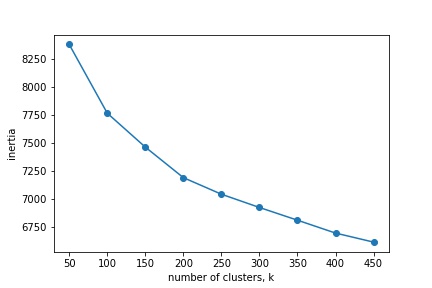


Figure 2: Elbow method run 1

A pronounced elbow wasn’t observed from this first run, but it seemed it could be in the area of K = 200 to 300. Thus, the elbow method was run again in order to determine in a more clear inflection point was in that range. Unfortunately, Narrowing in on a K range didn’t provide a better illustration of an inflection point.

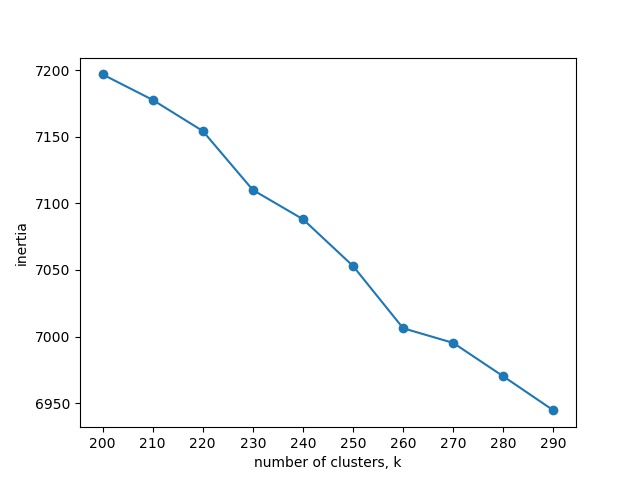


Figure 3: K = 200 to 300, no real inflection point observed.

Upon discussion with my mentor, it was decided to step away from K means at this point. K means was better suited for a lower number of clusters, but with so many possible defect labels, there would need to be reduction in dimensionality before anything of use could be gleaned from the analysis. LDA (Latent Dirichlet Allocation) was decided on as a new option for the analysis. LDA is based on the assumption that the entire document corpus is made up of a number of topics. Each document in the corpus is a mixture of different topics, and the topics are assigned a probability. This greatly reduces the dimensionality of the problem while still allowing inspection of the topics and the complaints that are assigned those topics.

A first run at LDA was done using gensim. The same stopword removal and tokenization was done and 10 topics were chosen to start.

**Steps Forward:** In looking to improve upon the model, there are a number of steps that must be followed. Primarily, more preprocessing on the text must be done to clean out unnecessary terms that are currently included and biasing the model. From the output of topics, words like “bmw” and “vehicle” are completely unnecessary because all the complaints we are currently processing are for BMW vehicles. Additionally, there are terms like “tl\*” that are nonsensical annotations that don’t improve our groupings. Also, stemming needs to be revisited. Currently, a very aggressive stemmer is used and lemmatization may be a better way to reduce the amount of terms in the corpus without trimming out valuable information. Finally, if possible in a similar method to K-means, the number of topics used should be optimized. For a first attempt, ten was arbitrarily chosen, however, this may not be ideal. It should be noted, that for our use case, even properly categorizing 10 or 15 of the largest complaint categories would be a huge help. About a dozen categories make up a significant portion of the complaints received on a yearly basis as figure 1 shows, and that could produce a savings of 30% to 50% of the time spent in manual review. This will be the goal for the next part of the analysis.