# Cluster analysis of Sabre Insurance data

(2181 Words)

## Introduction: business problem overview

Cluster analysis is an unsupervised machine learning method which groups data objects into clusters. Similar data objects are grouped into one cluster, while dissimilar data objects are assigned to other clusters [1]. One of the most popular applications of clustering is in the insurance industry.

Sabre Insurance Company Ltd, one of the most outstanding insurers in the UK, specialise in selling motor insurance through brokers and its own brands: Insure 2 Drive, Go Girl and Drive Smart [2]. To aid their business operations, Sabre use supervised learning tasks. Classification, for example, is used to predict the probability that a new customer will make a claim, or the probability that an existing customer is a victim of fraud. Data collected by the company to be input into such supervised models often contain high cardinality categorical variables (hccv). Hccvs are categorical variables with many levels, for example vehicle type, which cannot be put into supervised models. To deal with this, further information is gathered on the hccv to form a new dataset and a lower dimension representation of the dataset is found which can be input into the models.

A dataset containing 65,340 data entries and 347 features describing a hccv has been provided by Sabre. The task set is to find a lower dimension representation of the data to be used in downstream tasks. Clustering is the chosen method to do this.

This report describes the cluster analyses carried out on the dataset supplied by Sabre to find a lower dimension representation of the dataset. It describes the data cleaning process and the construction of a distance matrix. The three algorithms used to cluster the data: hierarchical agglomerative clustering (HAC), Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) and k-medoids are compared, comparing the results of these algorithms. Limitations to the analyses are discussed, suggesting improvements. The report concludes that the analyses have been somewhat successful, but more work is required.

## Data Cleaning

Data cleaning is an important step in any analysis. The dataset provided by Sabre contains duplicate variables, data errors and missing data. First, variable duplicates and variables containing no data (by error) are omitted. Next missing data entries are dealt with. Data entries with missing values from the variable ‘a_v6’ are omitted because they are included by error and are not real objects. Missing values from histogram variables are replaced with ‘0’ as are not missing data and must sum to 100. Missing values from numerical variables are replaced with the mean value of the variable.

Next categorical variables are one hot encoded for better performance of the algorithms. In doing so, two variables a_v7 and a_v8 which are in a hierarchy, are concatenated and the most frequent categories that make up the first 95% cumulative frequency are one hot encoded while the last 5% (the least frequent categories) are grouped into one category.

For python code for data cleaning, see scripts “01_ReadData.ipynb’ and ‘02_DataManipulation.ipynb’.

## Data matrix computation

Continuing the analyses, a distance matrix is computed for the clean dataset. For python code for computation of the distance matrix described, see script ‘03a_DistanceMatrix.ipynb’. A distance matrix is a table of distances between pairs of data in a dataset. Distance matrices are necessary for clustering to improve performance of the clustering algorithm and there are many examples of distance matrices being computed before clustering is performed. Aiello and Pegoretti [3] demonstrated this by clustering newspaper pages to store old documents digitally. Agglomerative clustering was performed, but prior to clustering, a distance matrix was computed.

Before the distance matrix is constructed, the data is scaled. The numerical variables and histogram variables have larger values compared to categorical binary variables, so when constructing a distance matrix without data scaling, the categorical variables would be negligible. As a result, the numerical variables are normalised, and the histogram variables are divided by 100 (so instead sum to 1 instead of 100). The categorical binary variables are not scaled. Next, the dataset is down sampled and a random

sample of 20% of the dataset is selected to compute the distance matrix. This is done due to RAM constraints on the computer used to carry out the analysis as the code cannot be run with more than 20% of the dataset.

In the computation of the distance matrix, the variable ‘id’ is excluded so is excluded from the clustering as is not useful. The target variables ‘'tq_dt1_mean', 'tq_dt2_mean', 'tq_dt3_mean', 'tq_dt4_mean', 'tq_dt1_std', 'tq_dt2_std', 'tq_dt3_std' and 'tq_dt4_std' are also excluded as are later used to evaluate the cluster results. Then, three separate distance matrices are computed for the three different types of variables in the dataset, being numerical, categorical and histogram variables. Later, these separate distance matrices are combined into a final distance matrix ‘D_3’. This is done so that different metrics can be used for each variable type. In addition, different weights can be added to the separate distance matrices, so greater importance can be given to the more important variables. Initially, an equal weighting is given to each variable type due to limited knowledge of the dataset and unclear importance of each variable.

The metric used for the numerical variables is the City Block distance. For two vectors x and y, the City Block distance is the sum of the difference between two vectors (xi - yi). The similar metric, Euclidean distance, is also considered. However, with this metric, vectors with larger differences contribute more to the distance matrix than those with smaller distances, making them negligible. Therefore, the city block distance is chosen.

For the categorical (binary) variables, the chosen metric is the Hamming distance. For two rows, the Hamming distance measures the number of positions with mismatching characters. This metric is used as it is popular for categorical variables and easy to implement in python using the function scipy.spatial.distance.pdist() [4], having a fast computational time. Furthermore, the Hamming distance has error correction capability, which enables reliable delivery of digital data and detection of errors [5].

The metric used for the histogram variables is the Jensen-Shannon distance. The Jensen-Shannon distance between two probability vectors p and q is defined as:

$$\sqrt{\frac{D(p \parallel m) + D(q \parallel m)}{2}}$$

Where m is the pointwise mean of p and q and D is the Kullback-Leibler divergence [6]. It is possible that this is not the optimal metric for the histogram data and others are considered, such as the Wasserstein distance. However, this is not appropriate as the metric does not account for all histogram bins [7]. An alternative metric is the Quadratic-Chi which calculates the relationship between cross-bins [8]. However, this metric is not available in the function scipy.spatial.distance.pdist [4] hence Jensen-Shannon distance is chosen.

**Algorithm comparisons**

Three algorithms are used in this cluster analysis. These being hierarchical agglomerative clustering (HAC), hierarchical density-based-spatial clustering of applications with noise (HDBSCAN) and k-medoids. For details on clustering using these three algorithms, see individual reports. One reason for the choice of these three algorithms is that all algorithms take a precomputed square distance matrix as the data input, so the distance matrix computed for 20% of the dataset can be clustered.

There are many advantages and disadvantages to these algorithms. HAC and HDBSCAN produce a hierarchy of clusters, a preference to Sabre. K-medoids does not do this and clusters are flat. Moreover, HDBSCAN does not have an assumption for a particular for a number of clusters, whereas HAC and K-medoids do and deciding on the number of clusters is a difficult task. Despite this, parameter selection in HDBSCAN is difficult and subjective. Another advantage to HDBSCAN is that it detects outliers in the dataset. The other two algorithms do not do this and will cluster outliers.

In all, it is difficult to identify which algorithm is best for finding a lower dimension representation of the dataset provided by Sabre, as all have advantages and disadvantages. Therefore, all three algorithms are used.

**Result comparisons**

The distance matrix computed for 20% of the dataset provided by Sabre is clustered using the three algorithms HAC, HDBSCAN and k-medoids. The results of these cluster analyses are compared. For python code for results described, see scripts “04_HierarchicalAgglomerativeClustering.ipynb’, ‘05a_HDBSCAN_exclude_target(tqv3_av78).ipynb’ and ‘06a_KMedoids.ipynb’. For further details on results from the three models, see individual reports.

First, the configuration of the clusters is compared. HAC produced 11 clusters, HDBSCAN produced 12 clusters and a cluster of outliers, and k-medoids produced 4 clusters. For cluster sizes, see Tables 1,2 and 3.

| Cluster Label | Cluster size |
|---|---|
| 2 | 4094 |
| 0 | 2798 |
| 3 | 2791 |
| 4 | 2849 |
| 1 | 484 |
| 6 | 4 |
| 9 | 1 |
| 8 | 12 |
| 5 | 2 |
| 10 | 2 |
| 7 | 2 |

Table 1. Cluster sizes of clusters produced by HAC

| Cluster Label | Cluster size |
|---|---|
| -1 (outliers) | 8244 |
| 0 | 1269 |
| 7 | 328 |
| 10 | 290 |
| 9 | 742 |
| 5 | 487 |
| 4 | 328 |
| 11 | 526 |
| 2 | 183 |
| 6 | 177 |
| 1 | 85 |
| 8 | 242 |
| 3 | 84 |

Table 2. Cluster sizes of clusters produced by HDBSCAN

| Cluster Label | Cluster size |
|---|---|
| 8208 | 3689 |
| 3366 | 3508 |
| 8947 | 2642 |
| 5637 | 3200 |

Table 3. Cluster sizes of clusters produced by k-medoids

In Tables 1 and 2, it is noted that the clusters formed by HAC and HDBSCAN vary in size significantly. The discrepancy in cluster size is higher in the clusters formed by HAC compared to HDBSCAN, given that there is a difference in size of 4092 objects compared to 1185. The clusters produced by k-medoids do not vary in size. It is thought that very small clusters provide a poor lower dimension representation of data. Hence k-medoids provides the best representation in this sense. It is also noted that HAC and HDBSCAN produce a similar number of clusters and k-medoids does not.

The clusters were plotted on T-distributed Stochastic Neighbour Embedding (t-SNE) plots to visualise the clusters (see Figures 1,2&3)

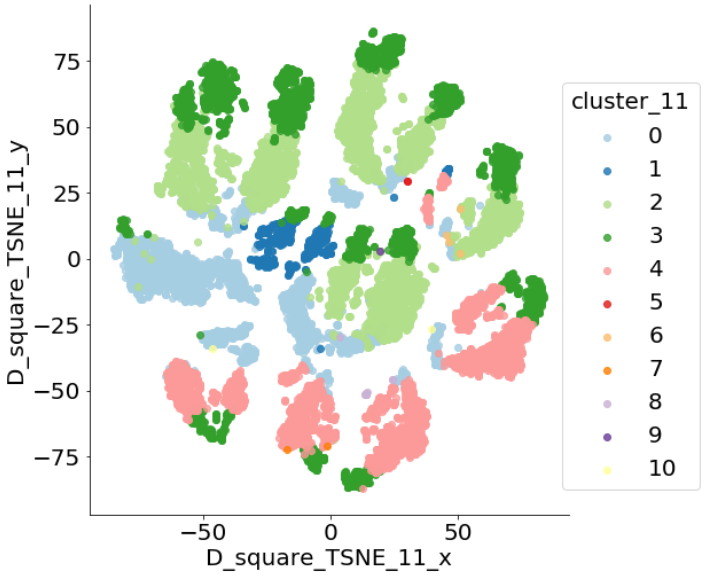


Figure 1. t-SNE plot of clusters formed by HAC

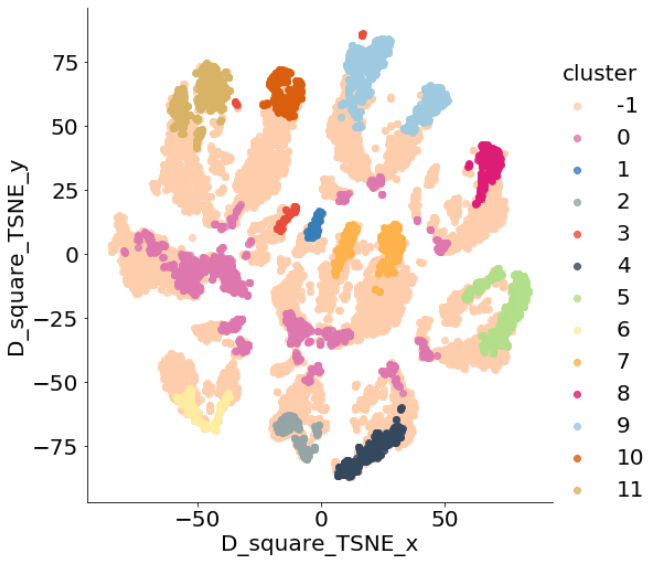


Figure 2. t-SNE plot of clusters formed by HDBSCAN

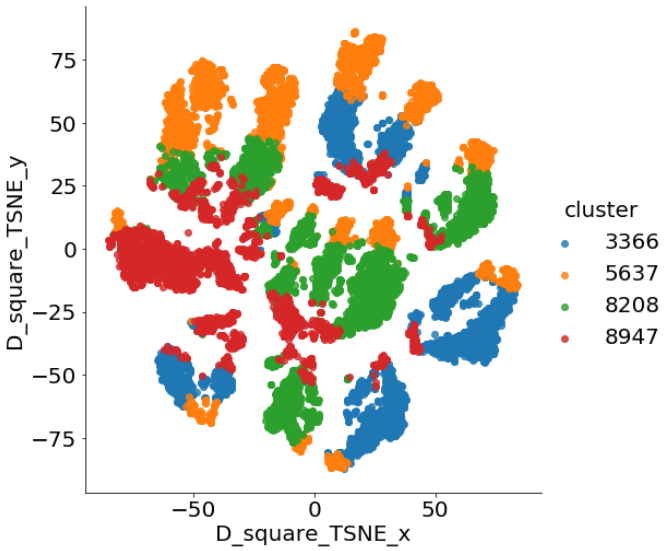


Figure 3. t-SNE plot of clusters formed by k-medoids

In Figures 1,2&3, the overlapping of clusters produced by all algorithms is noted. The configuration of cluster 3 from HAC and 5637 from k-medoids and cluster 2 from HAC and 8208 from k-medoids have similar patterns. This infers that the two algorithms are producing similar results. Background knowledge of the objects in these clusters and feedback on downstream tasks is required to identify if this is a true finding. Feedback from Sabre is required for this. Moreover, the larger number of outliers from HDBSCAN is emphasised.

Next, the distribution of the 8 target variables in the clusters produced by the three algorithms is investigated to identify if the clusters are predicting these variables. Boxplots are produced, showing the distribution of the values of the target variables in the clusters. One box plot shows the distribution of one target variable in the clusters produced by one algorithm. Three of these boxplots are shown in Figures 4,5&6.

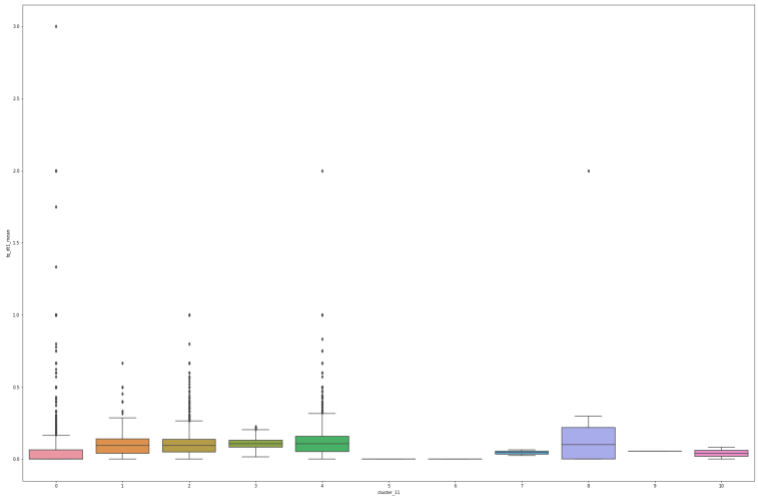

Figure 4. Boxplot showing 'tq_dt1_mean' distribution in clusters formed by HAC

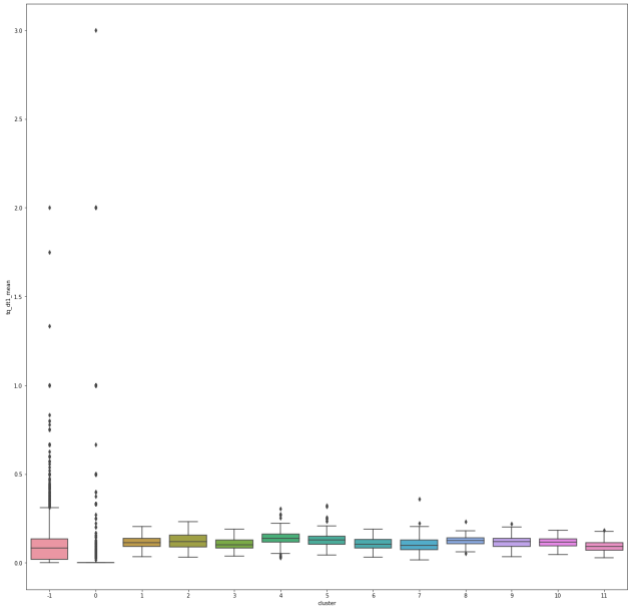

Figure 5. Boxplot showing 'tq_dt1_mean' distribution in clusters formed by HDBSCAN

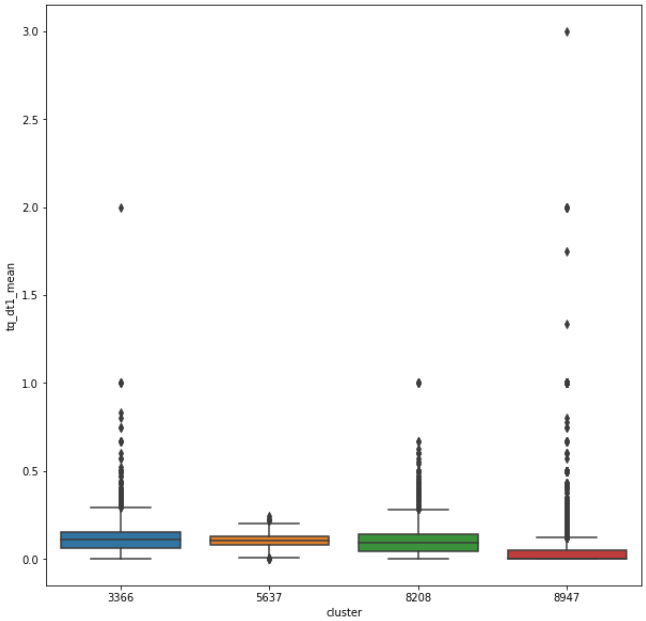


Figure 6. Boxplot showing 'tq_dt1_mean' distribution in clusters formed by k-medoids

From Figures 4,5&6 it is noted that the mean and quartile values of the target variable 'tq_dt1_mean' are overlapping in the clusters produced by all three algorithms. This infers that none of the clusters produced by the three algorithms are predicting the target variables. However, this does not infer that the cluster results are poor.

**Limitations and Improvements**

Limitations to the analyses described in this report are identified, and improvements should be made.

Dataset size

First, there are limitations to the large dataset size. Only 20% of the dataset is used to compute the distance matrix due to memory constraints of the computer, having only 8GB RAM. It is recommended that the code is recoded so has better scalability and that the analysis is carried out on a computer with higher RAM. It is important that the analyses are carried out on the whole dataset due to the nature of the task. A lower dimension representation of a dataset is poorer if only consists of a proportion of the data. Moreover, the cluster results are not stable when using different percentages of the data. This was noted by Sabre Insurance when testing the results of HAC and comparing cluster results from 60% and 100% of the

dataset. Therefore, it is important that the analyses are carried out on the whole dataset to ensure stability of the methods and results.

Results visualisation

Additionally, there are limitations to the visualisations of the results. Visualisation is important as aids human evaluation of the results at Sabre. One such limitation is using t-SNE for cluster visualisation for the large dataset. It is usually recommended to first use principle component analysis or TruncatedSVD to supress noise for an improved t-SNE plot [9].

Moreover, to better assess the distribution of the 8 target variables in the clusters to identify if clusters are predicting these variables, other visualisation techniques should be used. For example, a dendrogram of the clusters should be plotted, colour-coding the nodes of the dendrogram with the average values of the target variables. Blotches of the same colour would indicate that the clusters are predicting the target variables.

## Conclusion

Using clustering to find a lower dimension representation of the dataset provided by Sabre has been somewhat a success. The analyses have not produced any exciting results, nor led to improvements in downstream supervised learning tasks. However, the analyses have produced unanswered questions, which aids Sabre with their progress. Lots more work is required on these analyses.

## References


1. Columbia University. What is Cluster Analysis? [Internet]. Available from: http://www.stat.columbia.edu/~madigan/W2025/notes/clustering.pdf
2. Sabre Insurance Company Limited. About Us [Internet]. 2017. Available from: https://sabre.co.uk/about/
3. Aiello M, Pegoretti A. Textual Article Clustering in Newspaper Pages. Dep Inf Commun Technol [Internet]. Available from: https://pdfs.semanticscholar.org/91e0/e9bdac5eb67a4b1ed3bf52f2f68fe34775f8.pdf
4. SciPy.org. scipy.spatial.distance.pdist [Internet]. 2019. Available from: https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.pdist.html
5. Tokareva N. Hamming Distance. ScienceDirect [Internet]. 2015; Available from: https://www.sciencedirect.com/topics/computer-science/hamming-distance
6. SciPy.org. scipy.spatial.distance.jensenshannon [Internet]. 2019. Available from: https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.jensenshannon.html#scipy.spatial.distance.jensenshannon
7. Stanford University. The Earth Mover's Distance (EMD) [Internet]. Available from: http://infolab.stanford.edu/pub/cstr/reports/cs/tr/99/1620/CS-TR-99-1620.ch4.pdf
8. Pele O, Werman M. The Quadratic-Chi Histogram Distance Family [Internet]. Available from: http://leibniz.cs.huji.ac.il/tr/1218.pdf
9. Scikit-learn. sklearn.manifold.TSNE [Internet]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html