Cluster Analysis on Sabre Insurance Data Using HDBSCAN Method

1 Introduction

1.1 Problem Statement

Nowadays, most insurance companies use machine learning widely to help their business operations. It is common to see supervised learning applied to customer segmentation, claims and risks prediction and fraud detection [1]. However, unlike the usual tasks, the main object of this project is to perform an unsupervised learning on the data of Sabre Insurance Company, which represented high cardinality categorical variables (hccv), to get greater insight into the data and the lower level representation of the categories which can be used in downstream tasks.

Cluster analysis, one of unsupervised learning methods, is chosen to achieve this object. Through cluster analysis, data objects are grouped into several clusters. Data objects within the same cluster are more similar, and vice versa [2]. Although there are different methods for clustering, the three algorithms, hierarchical agglomerative clustering (HAC), hierarchical density-based-spatial clustering of applications with noise (HDBSCAN) and K-medoids, which are considered more appropriate for the dataset, are used in this project. This report will focus on HDBSCAN, and the two other clustering methods will be discussed thoroughly in the other partner's reports.

1.2 Data Pre-process

There are 65,340 data entries and 347 features, which contain a variety of information about car categories. Due to three different types of variables and the confidential nature of data, data preparation is a crucial process prior to performing the cluster model. There are two main steps: data cleaning and building the distance matrix. Firstly, raw data should be omitted errors and duplications, filled missing values and transformed the format. Secondly, most clustering algorithms which are based on the distances between points directly provide the distance calculation in the function. However, in this dataset, the choice of proper distance measure for numerical, categorical and histogram variables are Cityblock, Hamming and Jensen-Shannon, respectively. As a result, the distance matrix which presents the distance between pairs of data should be self-defined and pre-computed. Further detailed information on these processes is available in the group report.

This report will demonstrate the methodology of HDBSCAN carried out on the Sabre Insurance dataset. In the first section, it will introduce HDBSCAN with theory and practical knowledge. In the second section, it will present the process of HDBSCAN implementation in this dataset. Greater emphasis will be placed on the parameter experiment in this model. The following section will describe and evaluate the result of cluster in HDBSCAN model through visualisation. The last section will discuss the reflections and improvements suggested for the future.

2 Methodology of HDBSCAN

2.1 The Theory of HDBSCAN

Hierarchical density-based-spatial clustering of applications with noise (HDBSCAN) which is proposed by Campello, Moulavi, and Sander, is a hierarchical density-based clustering algorithm [3]. As the name indicates, firstly, it arranges data objects within clusters to construct a clustering hierarchy, detecting the structure of clusters [4]. Secondly, unlike distance-based algorithms such as HAC and K-means, it is a density-based algorithm which clusters the higher dense areas and separates the lower dense ones, allowing clusters can be any shape. Thirdly, it has the notion of noise which labels the density of area lower than any other cluster as outliers, finding the noise in real-world data set [5].

In fact, HDBSCAN can be regarded as an improvement of DBSCAN algorithm, which is the most representative method among density-based algorithms. They are similar in theoretical and practical aspects. One of the noteworthy differences is that HDBSCAN performs DBSCAN over all varying epsilon values, which stands for the local radius considered when forming a cluster [6]. For DBSCAN, there are two basic parameters which are minimum samples and epsilon [7]. The selection of these two parameters is always the classical issue in practice. In order to deal with this, HDBSCAN removes the requirement of epsilon. This means that HDBSCAN can find not only the stable clustering over epsilon values, but also the clusters with differing densities[6]. Hence, HDBSCAN is more robust for parameter selection. Also, HDBSCAN provides a hierarchy of clusters, while DBSCAN can show only the flat (non-hierarchical) of clusters [8].

According to McInnes et al. [8] and Chen et al. [9], the process of HDBSCAN can be divided into five steps, as follows.

1. Transform the space to make the clustering core more robust to outliers by defining the ‘mutual reachability distance’.
2. Construct the minimum spanning tree by adding the lowest weight edge which connects the tree.
3. Convert the spanning tree into the cluster hierarchical through sorting the edge of the tree by distance.
4. Condense the above cluster hierarchical into a smaller tree by taking the parameter ‘minimum cluster size’.
5. Extract the clusters by selecting the largest clusters in the condensed tree dendrogram.

2.2 Comparison With Other Clustering Algorithms

Besides being an improvement over DBSCAN, compared to other clustering paradigms, the characteristics of HDBSCAN make it outstanding. Given that there is more than one type of variable and that prior knowledge of the data is lacking, HDBSCAN seems to suit the dataset, due to the following reasons:

Although there are at least ten clustering methods which can be easily applied in the 'sklearn.cluster' module of Scikit-learn, some methods such as K-means, which cannot support the pre-computed and user-defined distance matrix as input, can be dismissed [10]. Another point is that many paradigms

necessitate the exact number of clusters, or other assumptions and unintuitive parameters.
In contrast, in this case, HDBSCAN, which can be viewed as an exploratory data analysis (EDA) tool, shows its natural advantage. Being a good EDA tool, HDBSCAN satisfies these requirements [11].

1. Be more conservative when trying to gain the result by detecting the outliers.
2. Parameters should be more initiative for choosing such as minimum cluster size.
3. Obtain a stable and predictable result when running the model with sampling data or experiment parameter.
4. Be able to run on the big dataset.

This makes HDBSCAN more appropriate for this dataset than other algorithms.

2.3 Other Application Examples

Before starting the work, research for other studies based on HDBSCAN algorithm that might provide support, is considered. However, perhaps since HDBSCAN is a fairly new algorithm of recent origins, most of the reports available are related to DBSCAN. And most reports use HDBSCAN on geographical data to handle the geographical distance, and also to get a high-quality result such as recognising construction patterns [12] and organising the trip [13]. There is one more relative insurance data report found using HDBSCAN and customers' previous purchase data to analyse customer behaviour, but it has only the one type of data which is numerical [14]. However, it indicates the feasibility of HDBSCAN for this project.

3 Implementation

3.1 Build the Model

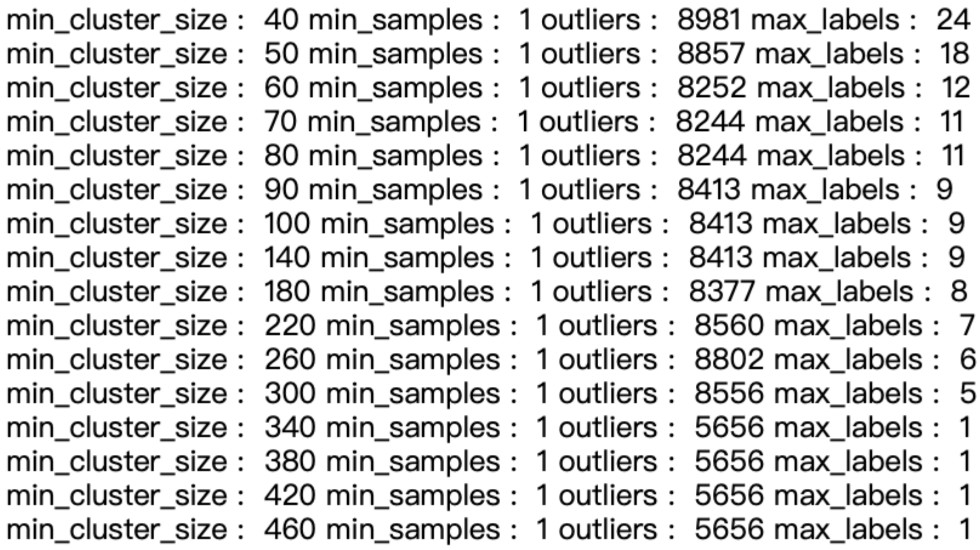
In order to perform HDBSCAN clustering, the 'hdbscan' package which inherits sklearn classes was installed and imported in Python [15]. Further, after pre-processing the data, the distance matrix which was already constructed with the clean data was the input data in the hdbscan.HDBSCAN(). Because of the computer processor and computational time, the dataset was reduced to 20 per cent. Thus, the distance matrix included only 13,039 of the total 65,196 data entries
. These data entries were selected at random with 'random_state = 123'. Moreover, it should be noted that the distance matrix was calculated without the eight variables that were labelled as possible target variables by Sabre Insurance.

Apart from the above-mentioned conditions, in order to obtain a better result of clustering, three slightly different distance matrixes were separately created and input to build three different HDBSCAN models, with the hope of observing and understanding data from these. Following the order of building the model, the first, standard version (Model 1) only excludes the target variables to evaluate the target analysis in the next step. Then the second one (Model 2) excludes the target variables and the other two variables. The last one (Model 3) excludes the target variables, but adds the weights on other variables. These will be presented further in the next three parts. For Python code for these three distance matrixes, please see scripts ‘03a_DistanceMatrix.ipynb’, ‘03b_DistanceMatrix_exclude_target_tqv3_ av78.ipynb’ and ‘03c_DistanceMatrix_exclude_target_weights.ipynb’.

The procedure for building the HDBSCAN model is simple and clear. To begin with, similar to the other methods need, HDBSCAN too is needed to choose the parameters. Except for the manual tweak, there is no other method which can help the choice. After deciding the parameter values, HDBSCAN model is completed as well. Fortunately, this ‘hdbscan’ package provides other functions to investigate deeper, such as outlier scores, probability scores and condensed trees [16].

3.2 HDBSCAN Parameter Selection in Model 1

This part would discuss the parameter selection for HDBSCAN in Model 1. For Python code for Model 1, please see script ‘05a_HDBSCAN_exclude_target(tqv3_av78).ipynb’. Thanks to the advantages of HDBSCAN, the parameter search is less complicated than for others. In the hdbscan.HDBSCAN(), although there is a large number of parameters available for change in the function, in practice, only four of them significantly affect the results of clustering: min_cluster_size, min_samples, cluster_selection_method, allow_single_cluster [17]. Of these, min_cluster_size and min_samples are the primary parameters. min_cluster_size is the minimum amount of data points that should be considered for a cluster. And min_samples indicates how conservative the clustering is. The larger the value, the more data objects are declared as noise. However, if the clustering result is still poor, the other two parameters, cluster_selection_method and allow_single_cluster could be adjusted. When the cluster result is only one or two large clusters, attempt to choose ‘leaf’ for cluster_selection_method. On the other hand, when the cluster result is numerous small clusters, attempt to choose ‘True’ for allow_single_cluster [17]. Based on these ideas, the parameter search of Model 1 is showed in Figure 1.



```
min_cluster_size :  40 min_samples :  1 outliers :  8981 max_labels :  24
min_cluster_size :  50 min_samples :  1 outliers :  8857 max_labels :  18
min_cluster_size :  60 min_samples :  1 outliers :  8252 max_labels :  12
min_cluster_size :  70 min_samples :  1 outliers :  8244 max_labels :  11
min_cluster_size :  80 min_samples :  1 outliers :  8244 max_labels :  11
min_cluster_size :  90 min_samples :  1 outliers :  8413 max_labels :  9
min_cluster_size :  100 min_samples :  1 outliers :  8413 max_labels :  9
min_cluster_size :  140 min_samples :  1 outliers :  8413 max_labels :  9
min_cluster_size :  180 min_samples :  1 outliers :  8377 max_labels :  8
min_cluster_size :  220 min_samples :  1 outliers :  8560 max_labels :  7
min_cluster_size :  260 min_samples :  1 outliers :  8802 max_labels :  6
min_cluster_size :  300 min_samples :  1 outliers :  8556 max_labels :  5
min_cluster_size :  340 min_samples :  1 outliers :  5656 max_labels :  1
min_cluster_size :  380 min_samples :  1 outliers :  5656 max_labels :  1
min_cluster_size :  420 min_samples :  1 outliers :  5656 max_labels :  1
min_cluster_size :  460 min_samples :  1 outliers :  5656 max_labels :  1
```

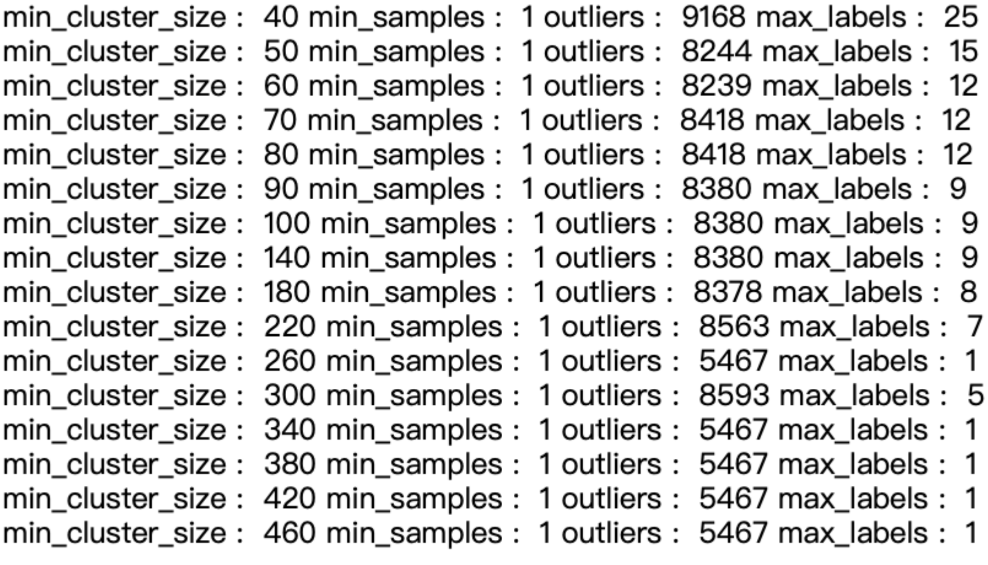
Figure 1

The sensible min_cluster_size values were tried from 40 to 500. The smallest value,1 chosen in min_samples, was due to too many outliers. The other two parameters were also tried, but did not appear to have improved the cluster results. After trials, it can be argued that the optimal parameters are 80 and 1 in min_cluster_size and min_samples. In these parameter values, the result of clustering has a reasonable number of clusters and fewer outliers with 12 and 8244, respectively.

Nevertheless, it is obvious that outliers are a serious issue. More than half of the points in 13,039 data objects are deemed as outliers. Hence, in a situation where the true cluster is unknown and the nature of data is unclear, the other two models were produced to experiment and explore how the cluster changes.

3.3 Experiment With Removing the Variables in Model 2

This part describes Model 2 which excludes the target variables, ‘tq_v3’ and ‘a_v7_8’ built the cluster. For Python code for Model 2, please see scripts ‘05a_HDBSCAN_exclude_target(tqv3_av78).ipynb’. The reason for removing ‘tq_v3’ is that according to the result of HCV methods, the clusters seem to vary dramatically by ‘tq_v3’, which means less common objects. It implies that the smaller cluster size had a lower ‘tq_v3’. Despite the result of HDBSCAN not having this problem, it is worthwhile to experiment whether it can reduce outliers. And ‘a_v7_8’ which are mega variables, are generated by unconfident method, and might mislead the result of HDBSCAN. Figure 2 displays the parameter search of Model 2. Since it was not much better than Model 1, it would not be discussed in the next paragraphs.



```
min_cluster_size :  40 min_samples :  1 outliers :  9168 max_labels :  25
min_cluster_size :  50 min_samples :  1 outliers :  8244 max_labels :  15
min_cluster_size :  60 min_samples :  1 outliers :  8239 max_labels :  12
min_cluster_size :  70 min_samples :  1 outliers :  8418 max_labels :  12
min_cluster_size :  80 min_samples :  1 outliers :  8418 max_labels :  12
min_cluster_size :  90 min_samples :  1 outliers :  8380 max_labels :  9
min_cluster_size :  100 min_samples :  1 outliers :  8380 max_labels :  9
min_cluster_size :  140 min_samples :  1 outliers :  8380 max_labels :  9
min_cluster_size :  180 min_samples :  1 outliers :  8378 max_labels :  8
min_cluster_size :  220 min_samples :  1 outliers :  8563 max_labels :  7
min_cluster_size :  260 min_samples :  1 outliers :  5467 max_labels :  1
min_cluster_size :  300 min_samples :  1 outliers :  8593 max_labels :  5
min_cluster_size :  340 min_samples :  1 outliers :  5467 max_labels :  1
min_cluster_size :  380 min_samples :  1 outliers :  5467 max_labels :  1
min_cluster_size :  420 min_samples :  1 outliers :  5467 max_labels :  1
min_cluster_size :  460 min_samples :  1 outliers :  5467 max_labels :  1
```

Figure 2

3.4 Experiment With Adding the Weights in Model 3

This part describes how the weights of the distance matrix change the clustering results in Model 3. For Python code for Model 3, please see scripts ‘05b_HDBSCAN_exclude_target_weights.ipynb’. After adjusting the parameter in HDBSCAN or ignoring the variables which might cause bias, there was no improvement in the serious outlier problem. However, apart from the HDBSCAN parameters in hdbscan.HDBSCAN(), the weights in the distance matrix, which are also the parameter values, could be adjusted. Although there is no knowing the importance of each variable, the clustering results might provide useful information when the weights were changed. For example, more insight into the nature of data should be possible by observing what happens when the weights are systematically changed. Table 1 summarises the results that include weights, min_cluster_size, outliers and the number of clusters. Surprisingly, it is clear that when the categorical variables were added to the weight, the clustering result was far better than other results. Compared to the other results which have an unreasonable number of clusters and more than half noise points, it has 16 clusters and only 411 outliers. This might suggest that there are some particular reasons and facts meriting further study. However, for the moment, it cannot be easily realised without background knowledge of the data.

Weight the variables					
Numerical	Categorical	Histogram	min_cluster_size	Outliers	Clusters
1	1	1	80	8244	12
5	1	1	80 - 460	6850	2
1	5	1	260	411	16
1	1	5	90-460	5435	3

Table 1

3.5 Limitation

During the process of building HDBSACN, it appears that there are two main limitations. Although parameter selection in HDBSCAN is easier than in other algorithms, it still does not have the decision criteria to determine the best values. Another difficulty is that, due to the feature of HDBSCAN, it might be too conservative for this data set. This may make outliers cannot be reduced.

4 Results and Evaluations

4.1 The Result of Model 1

Although three distance matrixes are used to build HDBSCAN models, only the result of Model 1 will be clearly illustrated as below. The result of Model 3 will be mentioned in the t-SNE visualisation part to compare with Model 1.

In Model 1, having tried a range of different parameters, the final result is 12 clusters (with labels 0 to 11) and a cluster of outliers (with labels -1). This gives the fewest outliers (8,244 data entries out of a total 13,039). The cluster size in each label is recorded in Table 2.

Label	-1	0	1	2	3	4
Cluster Size	8244	1269	85	183	84	328
5	6	7	8	9	10	11
487	177	382	242	742	290	526

Table 2

As can be seen in Table 2, two clusters (label 1 and 3) has smaller number of data compared to the other cluster sizes. Nonetheless, compared to HCV, which has more severely unbalanced cluster size (more details in HCV report), the result of HDBSCAN is more nicely distributed in each cluster.

Next, using the ‘hdbscan’ library support, the outlier scores and probability scores of HDBSCAN model can be accessed via ‘outlier_scores_’ and ‘probabilities_’. These two functions indicate how each data object fits into its cluster from 0.0 to 1.0. The higher the outlier scores data points are, the more is the likelihood of their being outliers [18]. The higher the probability scores data points have, the more likely it is that they are at the centre of clusters [19]. In order to show the overall performance of the HDBSCAN model, these two values were calculated by mean as statistical values, getting 0.119 in outlier scores mean and 0.32 in probability scores mean. This result shows the persistence of the outlier issue in this model too.

4.2 Target Variables Analysis

As mentioned above, because there are eight target variables defined by Sabre Insurance, one of the evaluation methods for the model is to analyse them and see if they vary by the cluster. Therefore, the values of each target variable were plotted in the box plots in each cluster. For instance, Figure 3 furnishes the box plot showing the distribution of 'tq_dt1_mean' in each of clusters.

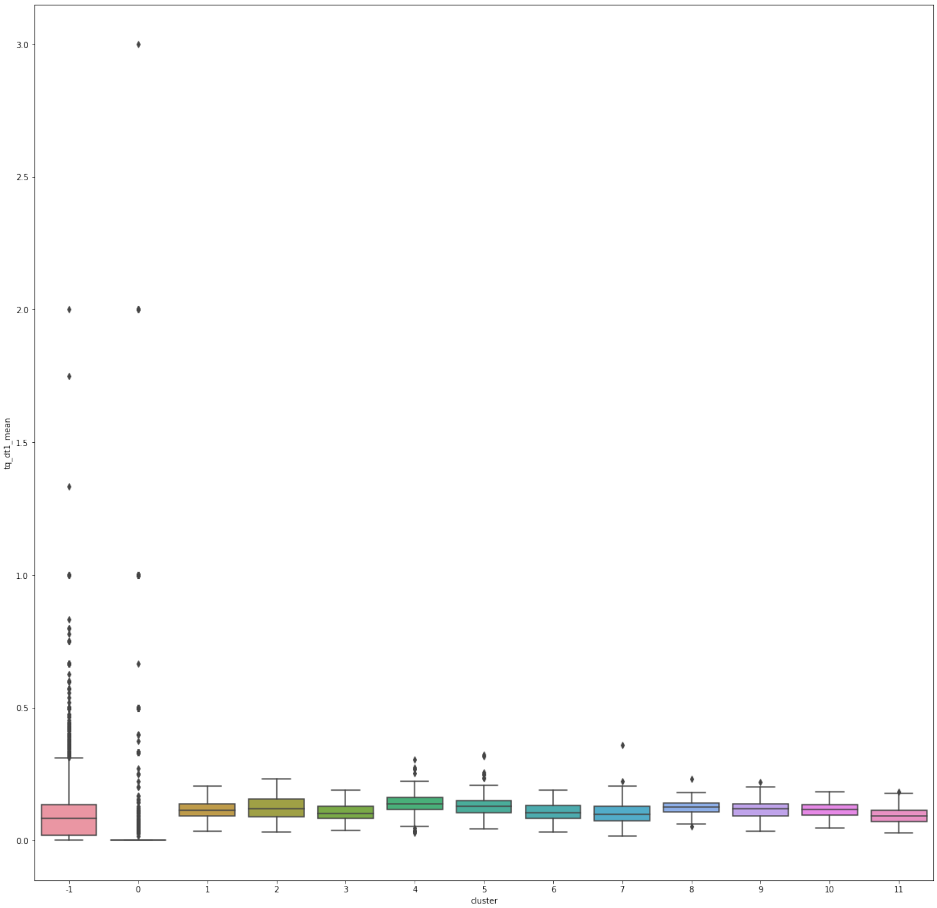


Figure 3 (Model 1)

Looking at the box plot, it is clear that the quartiles of ‘tq_dt1_mean’ and all the other target variables are overlapping between clusters. Owing to this, it is believed that the clusters cannot predict the target variables. However, it does not absolutely mean that the clusters are poor.

4.3 t-SNE Visualisation

At the present stage, although there is no perfect understanding of the result of Model 3, visualisation is the easiest way to compare the result of Model 1 and Model 3 to try to obtain more information. In order to achieve the visualisation, t-Distributed Stochastic Neighbour Embedding (t-SNE) was used for reducing the high-dimension to two dimension dataset by using the function ‘sklearn.manifold.TSNE()’ with ‘random_state = 123’ [20]. Figures 4 and 5 show the distribution of clusters by different colours in the two-dimension dataset. But after reducing the dimensions, the data points arrange themselves quite differently, as between the two models. Thus, the distribution of clusters cannot be compared directly. However, it appears that the clusters of Model 3 are more concentrated, clearer and bigger than those of Model 1. For example, except for cluster -1 which is outlier, cluster 0 of Model 1 is more dispersed than any cluster of Model 3.

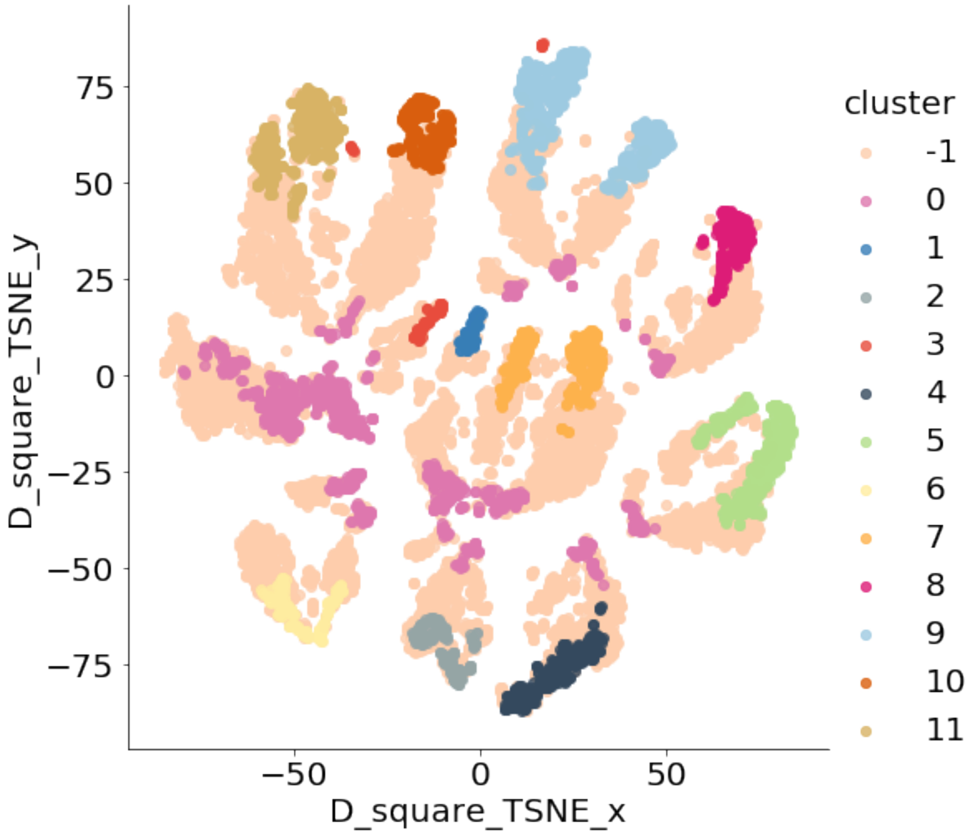


Figure 4 (Model 1)

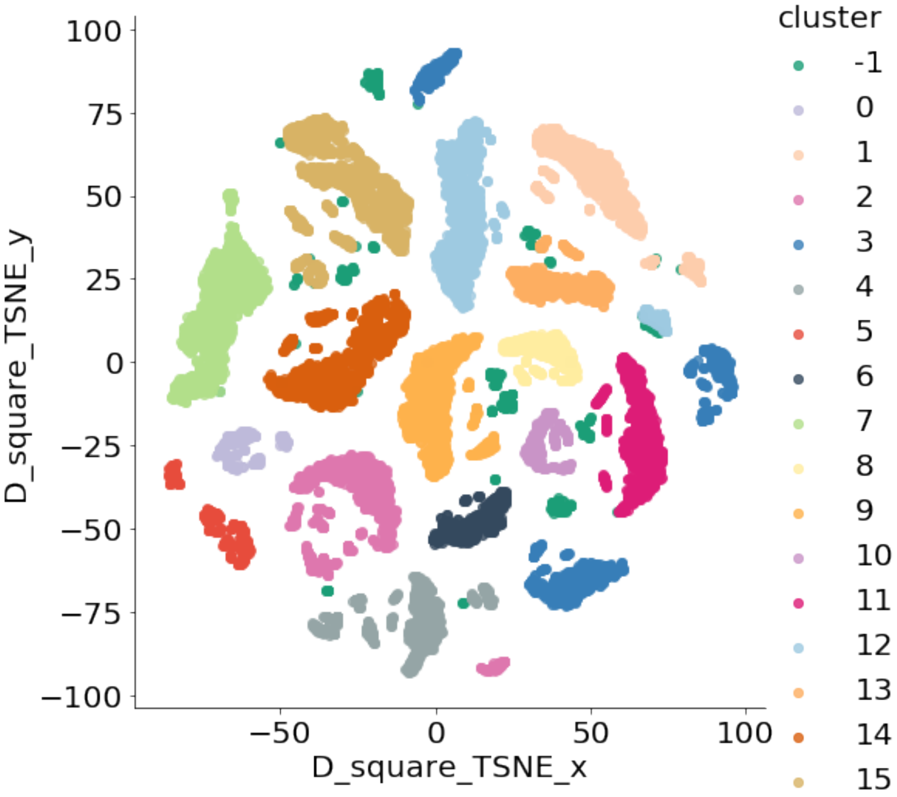


Figure 5 (Model 3)

In addition, the ‘hdbscan’ package also allows plotting the outliers. Figures 6 and 7 show the distribution of outliers in two models. Interestingly, although there are lots of outliers in Model 1, they look close to each other or even the existing cluster. The reason why they cannot be grouped should be investigated.

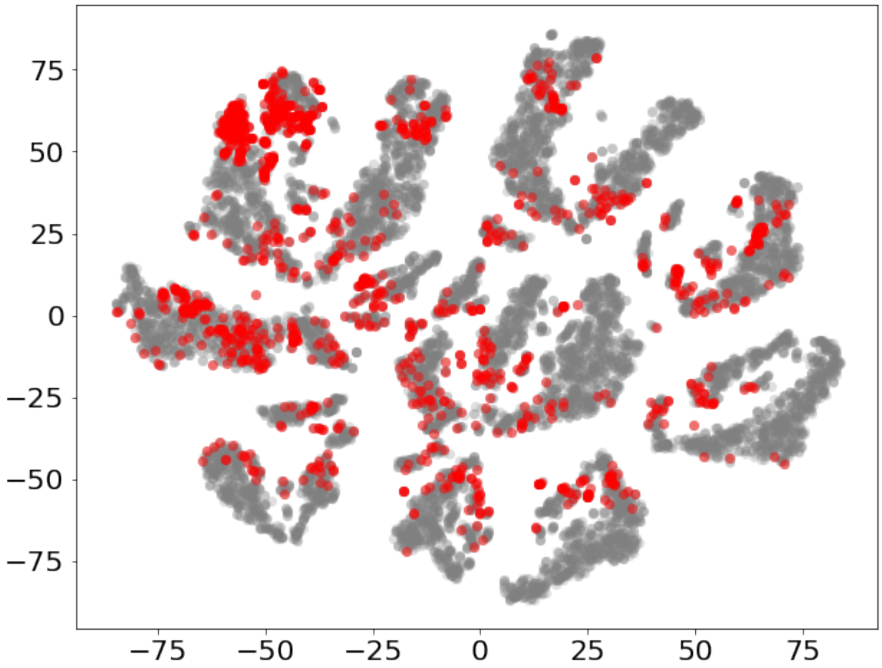


Figure 6 (Model 1)

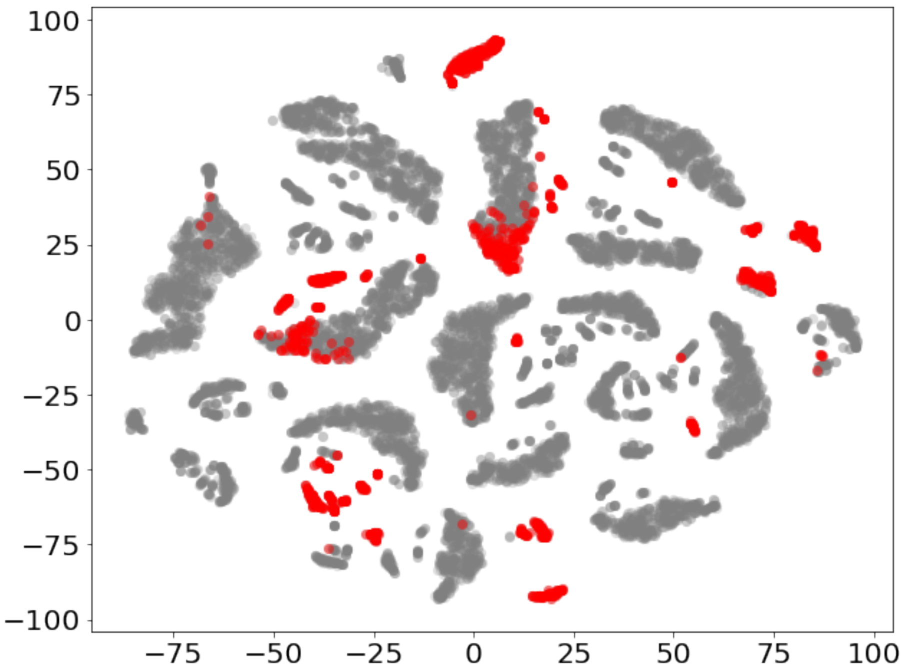


Figure 7 (Model 3)

5 Reflections and Improvements

Due to the lack of unsupervised learning experience and the confidential nature of data, this project presents a great challenge. This causes confusion as to whether the direction of research is correct. A lesson from this project is that before starting to work, no one can know the certainly accurate way to analyse for real-word data which is usually complicated. Thus, different paths of analysis are deserved to try when studying or coding. In addition, thanks to the guidance from the mentor, Sabre Insurance Company and partners, three basic models could be constructed.Overall, after interaction with the mentor and Sabre Insurance, several aspects could be improved in this project.

Firstly, after the codes are improved, a higher percentage data set will be able to run in the models. However, the report did not use more than 20 per cent to present the results due to the slower processor and computational time limitations. Secondly, the weighted distance matrix gives an interesting result which might include useful information. It should be systematically tested with more and different weights, to inspect how the result changes. Thirdly, in terms of interpreting the cluster, the condensed trees function 'condensed_tree_' in the 'hdbscan' package seems helpful. This function provides information about the cluster hierarchy in different forms. It not only can plot the hierarchy tree to visualise the order of cluster construct but also can convert the hierarchy tree into the pandas DataFrame to analyse the nodes of clusters [16]. However, this is a proposed idea which needs more knowledge. Fourthly, using other evaluation methods for HDBSCAN are necessarily required. In this model, only target variables analysis is used to measure the performance of the model. Silhouette scores and confusion matrix should be attached. Last, and most importantly, the outlier issue in HDBSCAN model should be paid more emphasis. Although it is possible that the dataset truly has many outliers, other reasons also need to be taken into account. The understanding of why these objects cannot be clustered will be useful to eliminate the problem, but there is no clue for this question at the moment.

6 Conclusion

In this project, clustering was chosen to be used in the analysis of the high cardinality categorical variables dataset provided by Sabre Insurance Company. Cluster analysis is a technique in unsupervised learning used to group similar data objects. The similarity of data objects is defined using a distance measure. The general process of implementation of this project as following: data cleaning, constructing the distance matrix, building the clustering model. In terms of clustering model, three methods which are used are hierarchical agglomerative clustering (HAC), hierarchical density-based-spatial clustering of applications with noise (HDBSCAN) and K-medoids.

This report is demonstrated HDBSCAN algorithm in theory and practice. From the theoretical perspective, HDBSCAN is similar to DBSCAN, but instead forms hierarchical and varied density cluster. This makes HDBSCAN more robust, intuitive and easier to select parameters than other clustering paradigms. From the practical perspective, three different distance matrixes are tried to solve the critical outlier issue in the models. This leads to the process of the experiment parameter is put much effort, not as easy as expected. These parameters include not only min_cluster_size, min_samples, cluster_selection_method and allow_single_cluster in HDBSCAN function but also weights of numerical, categorical and histogram in the distance matrix.

As mentioned previously, the result of Model 1 which excludes the target variables and the result of Model 3 which excludes the target variables and adds the different weights are worthy of being discussed. It can be argued that Model 1 which used the same distance matrix as other algorithms models can be compared the results of other models, but it includes serious outliers issue. Then, although Model 2 reduces lots of outliers, it cannot be compared with other algorithms models and also cannot be made more explanation on this result.

However, after further testing on HDBSCAN model by Sabre Insurance Company, the model has not produced any major breakthrough yet but seems to be possible information in downstream tasks which sounds exciting. At the same time, this model is suggested many aspects needed to be improved. This implies that continuous researching and working is required for the object of this project.

Reference:

[1] Bobriakov, I. (2018). *Top 10 Data Science Use Cases in Insurance*. [online] Medium. Available at: https://medium.com/activewizards-machine-learning-company/top-10-data-science-use-cases-in-insurance-8cade8a13ee1 [Accessed 14 Aug. 2019].
[2] Stat.columbia.edu. (n.d.). [online] Available at: http://www.stat.columbia.edu/~madigan/W2025/notes/clustering.pdf[Accessed 14 Aug. 2019].
[3] McInnes, L., Healy, J. and Astels, S. (2016). *How HDBSCAN Works — hdbscan 0.8.1 documentation*. [online] Hdbscan.readthedocs.io. Available at: https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html [Accessed 14 Aug. 2019].
[4] Zhou, R., Zhang, Y., Feng, S. and Luktarhan, N. (2018). *A Novel Hierarchical Clustering Algorithm Based on Density Peaks for Complex Datasets*. [online] Available at: http://awi.com/journals/complexity/2018/2032461/ [Accessed 14 Aug. 2019].
[5] Qiao, A. and Jackson, J. (2018). *Scalable Clustering for Exploratory Data Analysis*. [online] Medium. Available at: https://medium.com/@Petuum/scalable-clustering-for-exploratory-data-analysis-60b27ea0fb06 [Accessed 14 Aug. 2019].
[6] GitHub. (n.d.). *scikit-learn-contrib/hdbscan*. [online] Available at: https://github.com/scikit-learn-contrib/hdbscan [Accessed 14 Aug. 2019].
[7] Salton do Prado, K. (2017). *How DBSCAN works and why should we use it?*. [online] Towards Data Science. Available at: https://towardsdatascience.com/how-dbscan-works-and-why-should-i-use-it-443b4a191c80 [Accessed 14 Aug. 2019].
[8] McInnes L, Healy J. *Accelerated Hierarchical Density Based Clustering* In: 2017 IEEE International Conference on Data Mining Workshops (ICDMW), IEEE, pp 33-42. 2017
[9] Chen, M., Arribas-Bel, D. and Singleton, A. (2018). Understanding the dynamics of urban areas of interest through volunteered geographic information. *Journal of Geographical Systems*, 21(1), pp.89-109.
[10] Scikit-learn.org. (2019). *2.3. Clustering — scikit-learn 0.21.3 documentation*. [online] Available at: https://scikit-learn.org/stable/modules/clustering.html [Accessed 14 Aug. 2019].
[11] Hdbscan.readthedocs.io. (2016). *Comparing Python Clustering Algorithms — hdbscan 0.8.1 documentation*. [online] Available at: https://hdbscan.readthedocs.io/en/latest/comparing_clustering_algorithms.html#some-rules-for-eda-clustering [Accessed 14 Aug. 2019].
[12] Şahbaz, Kadir & Basaraner, Melih. (2017). *Recognising Building Patterns in Topographic Maps with HDBSCAN Clustering Algorithm*.
[13] Bendemra, H. (2018). *Using Unsupervised Learning to plan a vacation to Paris: Geo-location clustering*. [online] Medium. Available at: https://towardsdatascience.com/using-unsupervised-learning-to-plan-a-paris-vacation-geo-location-clustering-d0337b4210de [Accessed 14 Aug. 2019].
[14] POSTIGO SMURA, M. (2019). *Cluster analysis on sparse customer data on purchase of insurance products*. [online] Math.kth.se. Available at: https://www.math.kth.se/matstat/seminarier/reports/M-exjobb19/190417.pdf [Accessed 15 Aug. 2019].

[15] PyPI. (n.d.). *hdbscan*. [online] Available at: https://pypi.org/project/hdbscan/ [Accessed 15 Aug. 2019]
[16] Hdbscan.readthedocs.io. (2016). *Getting More Information About a Clustering — hdbscan 0.8.1 documentation*. [online] Available at: https://hdbscan.readthedocs.io/en/latest/advanced_hdbscan.html [Accessed 15 Aug. 2019].
[17] Hdbscan.readthedocs.io. (2016). *Parameter Selection for HDBSCAN* — hdbscan 0.8.1 documentation*. [online] Available at: https://hdbscan.readthedocs.io/en/latest/parameter_selection.html [Accessed 15 Aug. 2019].
[18] Hdbscan.readthedocs.io. (2016). *Outlier Detection — hdbscan 0.8.1 documentation*. [online] Available at: https://hdbscan.readthedocs.io/en/latest/outlier_detection.html [Accessed 15 Aug. 2019].
[19] Hdbscan.readthedocs.io. (2016). *Basic Usage of HDBSCAN* for Clustering — hdbscan 0.8.1 documentation*. [online] Available at: https://hdbscan.readthedocs.io/en/latest/basic_hdbscan.html [Accessed 15 Aug. 2019].
[20] Scikit-learn.org. (n.d.). *sklearn.manifold.TSNE — scikit-learn 0.21.3 documentation*. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html [Accessed 15 Aug. 2019].

Peer Assessment Form for the Masterclass in Business Analytics (BUS131)

Team Number/Name(s) Jui-Ting Hu, Rachel Dyer, Koonkarn Arthasarnprasit

As a team you should keep a record of your meetings, attendance, and performance/contribution of each of your team members based on the statements below. The team work requires you to be pro-active, i.e. if you notice issues in your team work, you should first discuss them internally and try to work them out as a team. If you cannot resolve them after having made all possible efforts, please notify the module organisers. In such cases the module organisers will have a meeting with the team to resolve any issues.

If you cannot reach one of your team members, please copy the module organisers in your emails.

Please fill out and submit below peer assessment form as a team on the end of the year on QMplus. Please append the form to your final essay. As a default all team members receive the same grade. However, in cases in which team members cannot come to an agreement regarding their contributions, each individual within the group should submit a separate peer assessment form via email to the module organisers. In this case individual grades may be altered. The module organisers reserve the right to alter the marks of individuals within a team in such a way that they reflect the effort of individual members. Your presentation and essay will not be marked until peer assessment forms have been received.

Please list all your team members, including yourself, by name in the tables below (left column) and then tick (x) the appropriate boxes for each group member.

1. **Meeting attendance & punctuality**

Team Member	Attended all meetings & arrived on time	Attended less than half of the meetings	Never attended any meetings
Jui-Ting Hu	x		
Rachel Dyer	x		
Koonkarn Arthasarnprasit	x		

2. **Contribution to meetings/group discussions**

Team Member	Contributed meaningfully to all meetings	Contributed something to less than half of meetings	Never contributed to meetings
Jui-Ting Hu	x		
Rachel Dyer	x		
Koonkarn Arthasarnprasit	x		

3. **Quality of work**

Team Member	Consistently produced high quality work	Work tended to be of average quality	Work tended to be of poor quality	No work produced
Jui-Ting Hu	x			
Rachel Dyer	x			
Koonkarn Arthasarnprasit	x			

4. Overall positive contribution to the team assignment, including discussion, code, analysis and presentation

Team Member	Has contributed significantly	Has at times made contribution	Has only made a small contribution	Has failed to make any contribution
Jui-Ting Hu	x			
Rachel Dyer	x			
Koonkarn Arthasarnprasit	x			

We confirm that all members of our group have reached the above consensus:

Name: Jui-Ting Hu Signature: ___________________________ Date: 16. Aug. 2019

Name: Rachel Dyer Signature: ___________________________ Date: 16. Aug. 2019

Name: Koonkarn Arthasarnprasit Signature: ___________________________ Date: 16. Aug. 2019