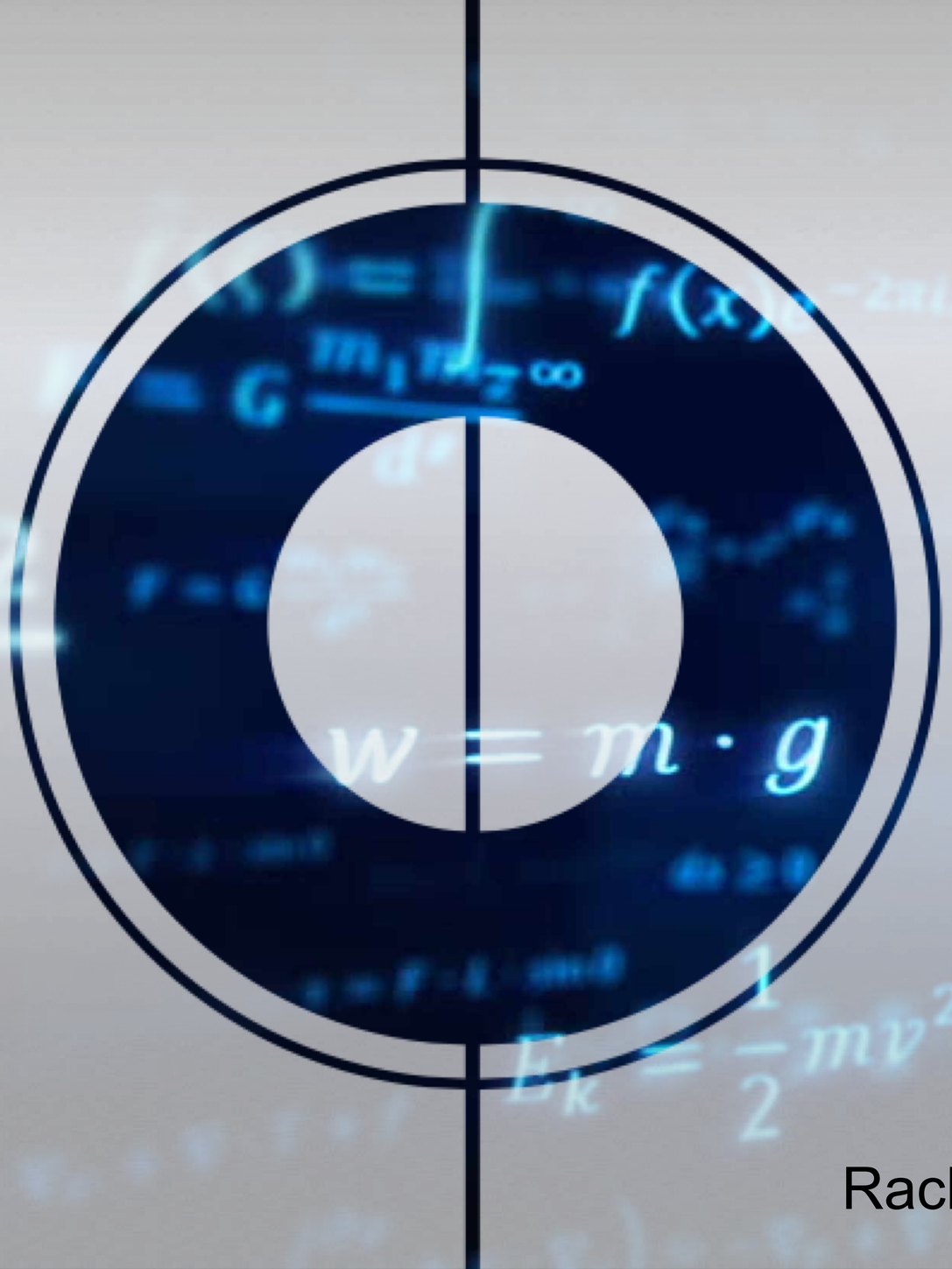


Sabre Insurance

Cluster Analysis

Rachel Dyer, Jui-Ting and Koonkarn Arthasarnprasit

- Vehicle type is **High Cardinality Categorical Variables** in supervised learning task.

- One method is to gather further information about the hccv.
  ➡️ **(65430, 347)**

- Using **unsupervised learning** to find a **lower dimension representation** of the data which can be used directly in the original supervised task.
  ➡️ **Clustering**

```
df_all.shape # (65,340, 347)
```

```
(65340, 347)
```

```
df_all.head()
```

	tq_db11_mean	tq_db12_mean	tq_db13_mean	tq_db14_mean	tq_dt1_mean	tq_dt2_mean	tq_dt3_mean	tq_dt4_mean	t
0	14.6561	5.8498	1.4466	0.0079	0.1502	0.1146	0.0079	0.0040	0
1	16.4545	12.6364	1.7273	0.0000	0.0909	0.0909	0.0000	0.0000	0
2	15.6510	5.9688	1.4531	0.0052	0.1146	0.0938	0.0104	0.0104	0
3	15.3430	6.0233	1.3605	0.1628	0.0872	0.0814	0.0058	0.0233	0
4	13.7838	6.0270	1.5203	0.0000	0.0743	0.0676	0.0000	0.0000	0

- Omit duplicate, error variables and all 0 columns.

- Deal with **missing values.**

➡️ Histogram variables replaced with 0.

➡️ Numerical variables replaced with mean.

- One hot encoding categorical variables.

- Deal with a_v7, a_v8 (a_v8 is a lower hierarchy under a_v7).

  Concatenate a_v7 and a_v8, creating new variable a_v7_8.

```
df_all.shape
```

```
(65196, 643)
```

```
df_all.head()
```

_97_647	a_v7_8__97_770	a_v7_8__97_875	a_v7_8__97_909	a_v7_8__99_37	a_v7_8__99_995	a_v7_8__99_996	a_v7_8_other
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0

Data Cleaning

Distance Matrix Computation

- Random sample **20%** of data.

```
df_all_2 = df_all.sample(frac=0.2, random_state=123)
```

```
df_all_2.shape
```

```
(13039, 643)
```

- **Scaling data** :

➡️ Normalise numerical variables.

➡️ Histogram variables divided by 100.

➡️ Categorical variables not scaled.

- Exclude ID and target variables.

- Distance metrics components :

➡️ **City block** for numerical variables.

➡️ **Hamming** for categorical variables.

➡️ **Jensen-Shannon** for histogram variables.

- Add equal **weights.**

Cluster Algorithm Comparison

HAC **HDBSCAN** **K-Medoids**

- Main reason: **precomputed square distance matrix** can be used.

- Advantages & Disadvantages :

➡️ HAC & HDBSCAN can produce **a hierarchy of clusters**. ↔ K-medoids clusters are flat.

➡️ HAC & K-medoids can decide **the number of clusters**. ↔ HDBSCAN can not do this.

➡️ HDBSCAN detects **outliers** in the dataset . ↔ HAC & K-medoids do not do this.

- It’s difficult to identify which algorithm is best.

Cluster Size Comparison

HAC

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

HDBSCAN

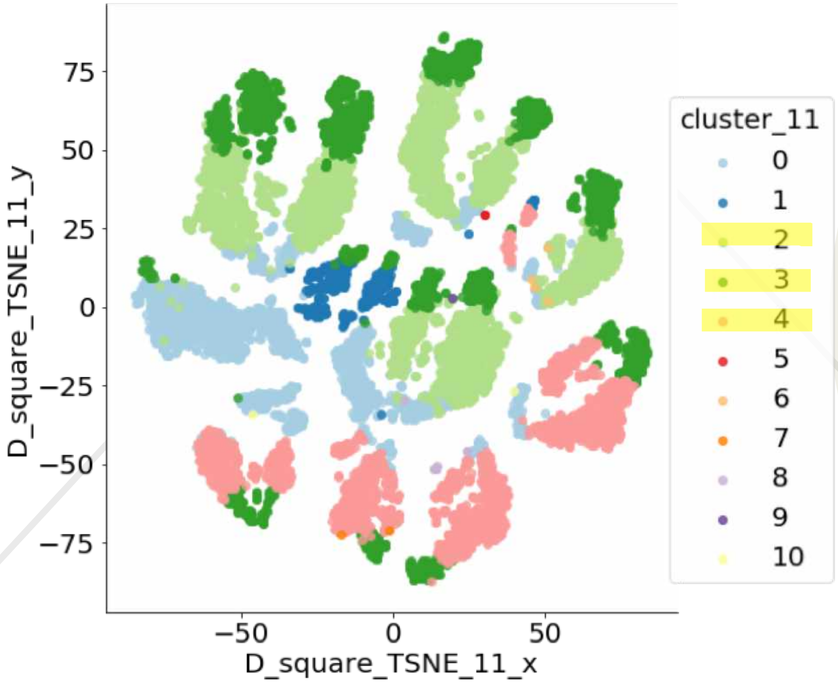
Cluster Label	Cluster Size
-1 (outliers)	8422
0	1269
7	328
10	290
9	742
5	487
4	328
11	526
2	183
6	177
1	85
8	242
3	84

K-Medoids

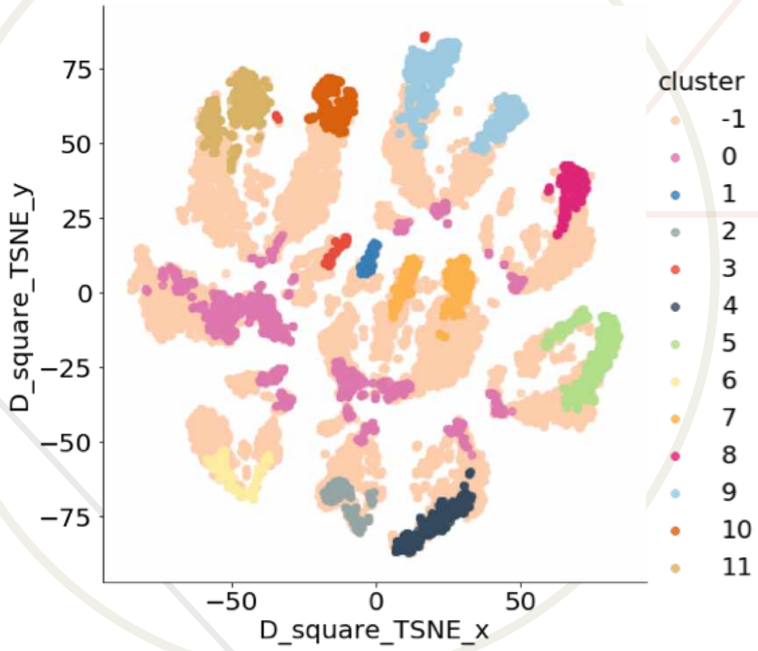
Cluster Label	Cluster Size
11433	3952
8947	2571
4872	2801
3366	3715

t-SNE Plot Comparison

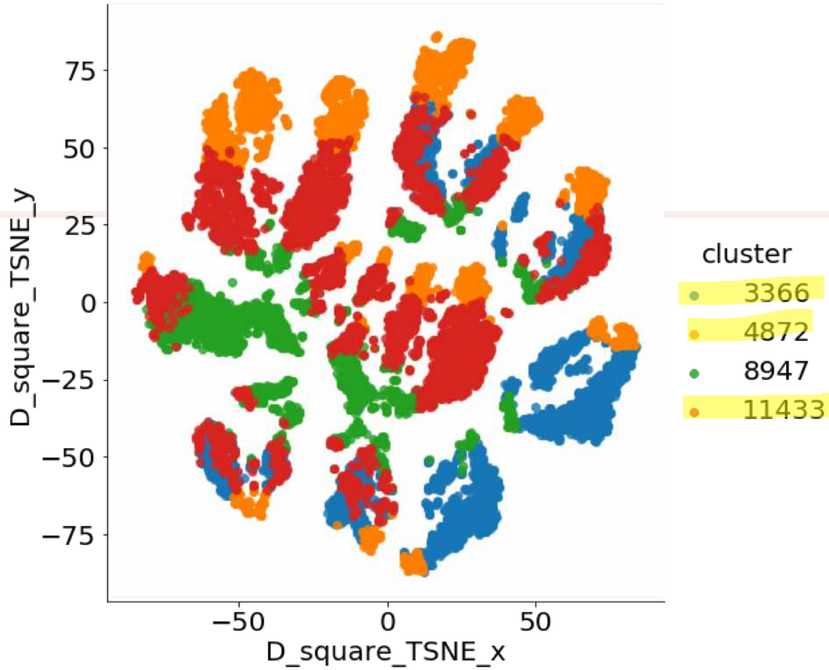
HAC



HDBSCAN

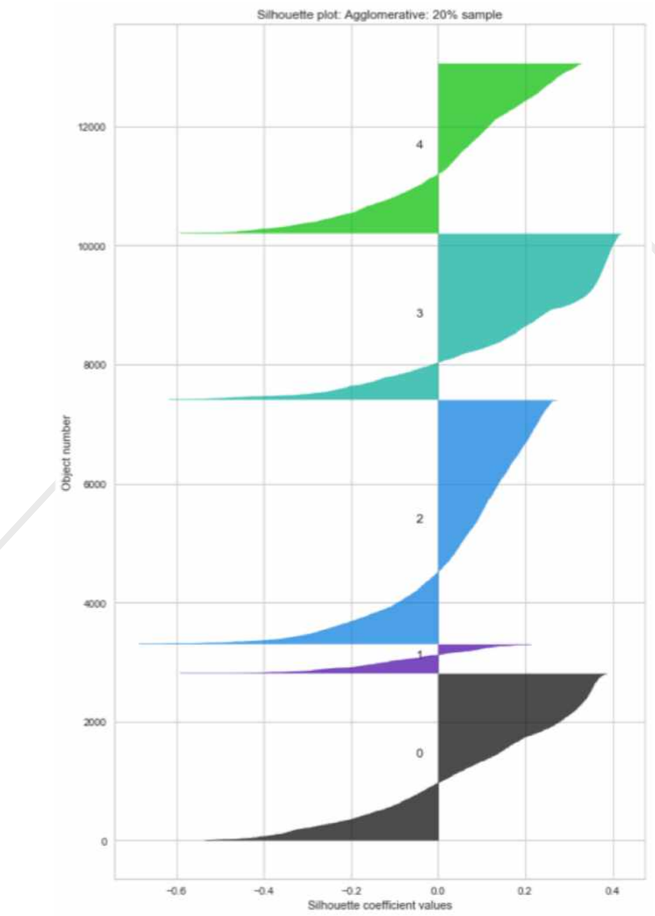


K-Medoids

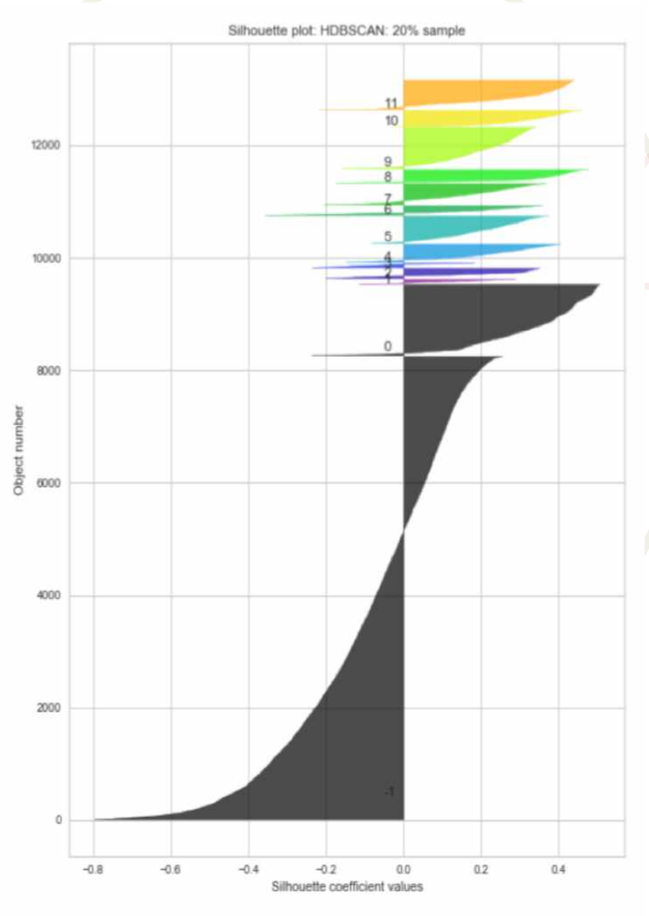


Silhouette Score Comparison

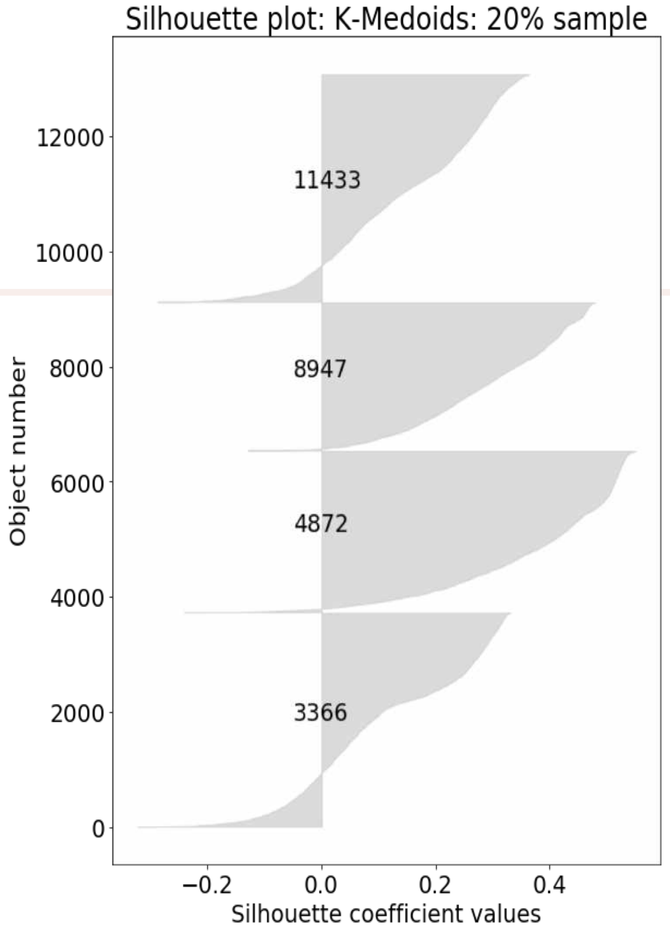
HAC



HDBSCAN

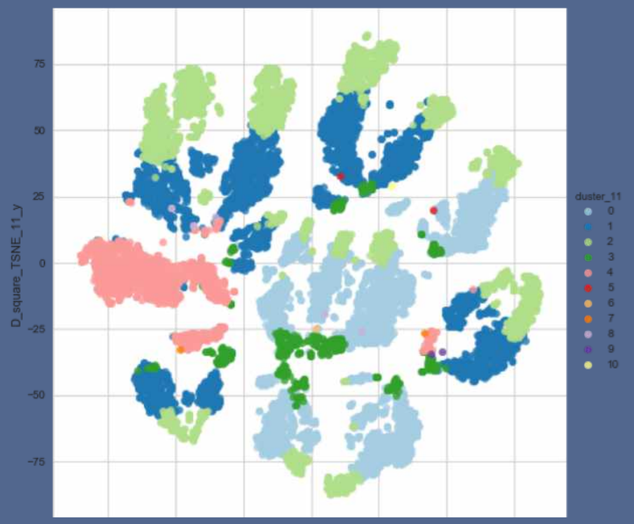


K-Medoids

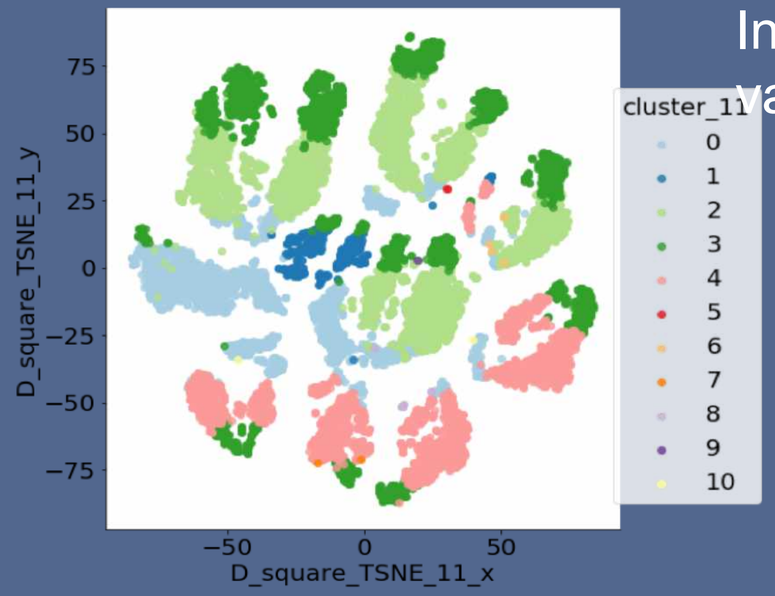


HAC Limitations & Further Work

- Varying cluster sizes.
- Different percentage of data produces different results.
- Including different variables produces different results.
  - ➡️ With/without target variables.
  - ➡️ With/without a_v7, a_v8 & tq_v3.
- Use different algorithms.
  - ➡️ HAC with greater selection of linkage.
  - ➡️ Hierarchical Divisive Clustering.
  - ➡️ Limitations of HAC on large data sets.



Excluding variables



Including variables

k-medoids Limitations & Further Work

- K- Medoids compute the compactness but not relationship.
- The algorithm select medoid randomly and every time it is rerun the optimal number changes.
- Use Elbow curve and Gap statistic to verify optimal number of clusters in comparison to Silhouette score.
- Calculate medoids manually and use the algorithm to find nearest points.
- K-Medoids is not the best when dealing with large dataset.
- Use PAM and CLARA to complement the existing K-medoids.

20%

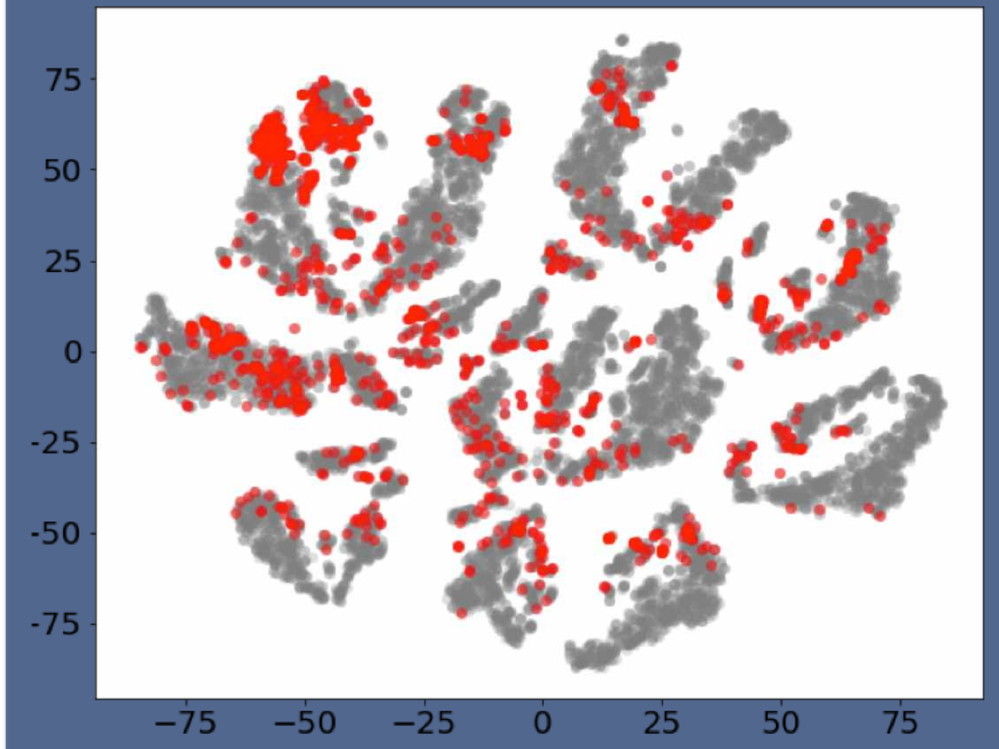
```
2 cluster 0.22399877482621114
3 cluster 0.20011354320561223
4 cluster 0.167287981299643
5 cluster 0.16533998789409415
6 cluster 0.15123478721775538
7 cluster 0.13771608311600655
8 cluster 0.13266757206370003
9 cluster 0.12381737638180808
10 cluster 0.1302421379930517
```

40%

```
2 cluster 0.22753689851970096
3 cluster 0.19622297084338347
4 cluster 0.1755156793969386
5 cluster 0.1686271993618024
6 cluster 0.14104502741957992
7 cluster 0.13241135634283946
8 cluster 0.13112956435832096
9 cluster 0.1311048569849475
10 cluster 0.12433485177322712
```

HDBSCAN Limitations

- Not easy to determine the best values for **parameters**.
- More than half of the data objects are deemed as **outliers**.
- The dimensionality of data affect the performance.



The distribution of outliers

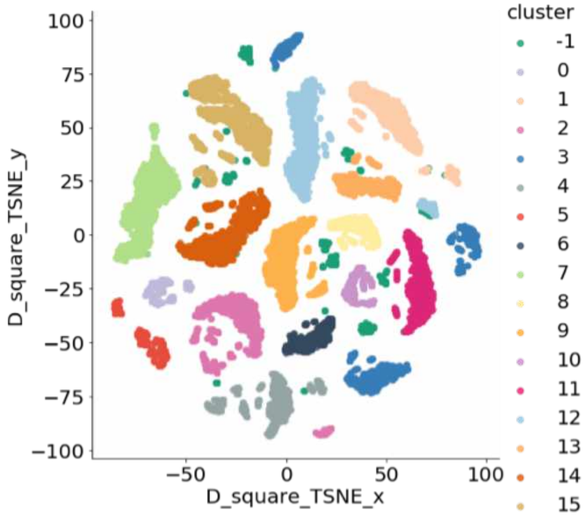
HDBSCAN Further Work

Experiment parameter with adding more different weights

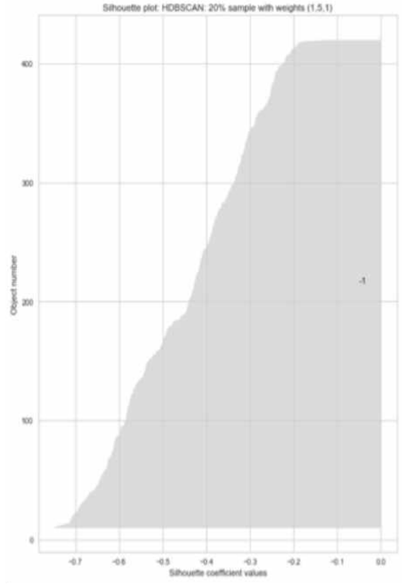
Numerical	Categorical	Histogram	min_cluster_size	Outliers	Clusters
1	1	1	80	8244	12
5	1	1	80 - 460	6850	2
1	5	1	260	411	16
1	1	5	90 -460	4629	2

Numerical	Categorical	Histogram	min_cluster_size	Outliers	Clusters
10	5	1	260 - 340	8750	7
1	10	5	460	1320	10
5	1	10	50 - 460	4291	2
1	10	1	260	486	16

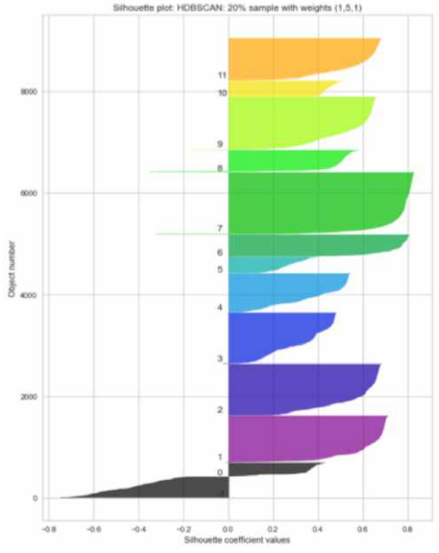
⭐ Numerical : 1, Categorical : 5, Histogram : 1 ⭐



The distribution of clusters



Outliers Sil scores



Clusters Sil scores

General Limitations & Further Work

- Dataset is too large.
  - ➡️ Use greater percentage of data.
- Lack of unsupervised learning and Confidential nature of data.

- Other methods to evaluate cluster results.
  - ➡️ Davies-Bouldin Index / Dunn index.
  - ➡️ Define quality measure.
  - ➡️ Inspect centroids.
- Other methods to find lower dimensional representation of the data.
  - ➡️ Feature agglomeration.
  - ➡️ Neural network encoders.

💡*Alan : It is always the way with these thing, that our first step of analysis lead to more question, but that's how we make progress...*

Thank Sabre Insurance and Alan for the opportunity!