Subject: Advanced Business Analytics Data Imputation Techniques

Customer segmentation of Motor Third Party Liability insurance

Author:

Andrzej Teżyk

Warsaw 2025

Introduction

The dataset we are working with, “Motor Third Party Liability insurance” contains 413 960 observations and 11 columns. Our objective is to segment customers and identify the primary factors influencing claim severity, enabling the actuarial department to more accurately evaluate risk for this specific line of business.

It was decided to use Python as our primary tool for analysis. After importing the necessary libraries and reading the dataset, we began our exploratory analysis by identifying missing values.

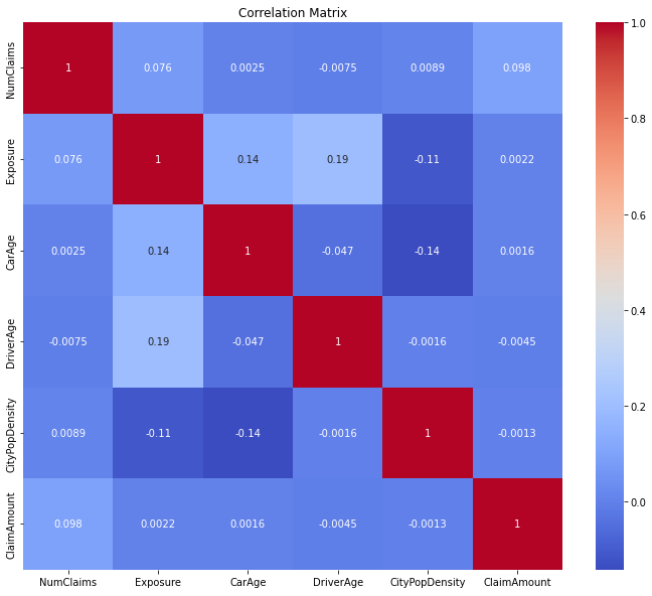
Data transformation and EDA

First, we found that column ClaimAmount contains 397 779 missing values (96,1%). As it is our target column we decided to fill them with the value of 0 as they are not actual NA, but unclaimed policies (NumClaims is 0 in such cases).

While checking duplicate values we discovered that when considering all columns, including ClaimAmount, there is only one duplicated row, and we deleted it. However, when excluding the ClaimAmount column, there are 791 duplicate rows in the data. This suggests that while the claim amounts might vary, many records share identical values across the remaining features because the policy users requested the policy money many times for the same PolicyID. We approached this issue by aggregating those rows by summing ClaimAmount values for the same PolicyID.

In our data we also make use of some transformations. We deleted the column containing row indexes, converted categorical columns that hold text data into binary values using one hot encoding, converted the PowerClass categorical column from letters to numbers and the data type of Exposure column was changed to float.

As part of our analysis, we created a correlation matrix to examine the relationships between the newly engineered variables.



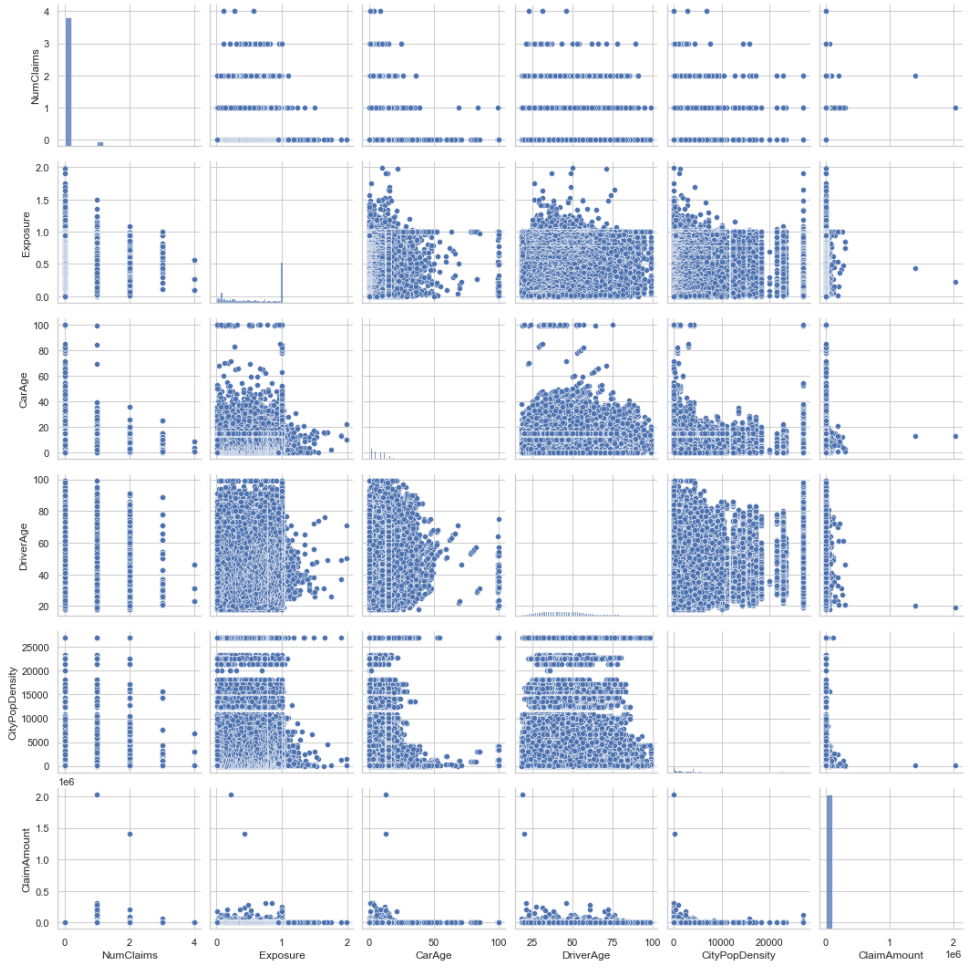
Picture 1. Correlation matrix

It was decided to build a correlation matrix to access the correlation between the continuous data in our datatable. According to the correlation matrix above, the strongest positive correlation was present between Driver Age and Exposure (0.19), but it still can be considered weak. The strongest negative correlation equaled –0.14 between Car Age and CityPopDensity, which is still weak. Overall, there are no variables in the data, that are strongly correlated with each other.

	feature	VIF
0	const	14.138953
1	NumClaims	1.016459
2	Exposure	1.080441
3	CarAge	1.042894
4	DriverAge	1.045959
5	CityPopDensity	1.030198
6	ClaimAmount	1.009694

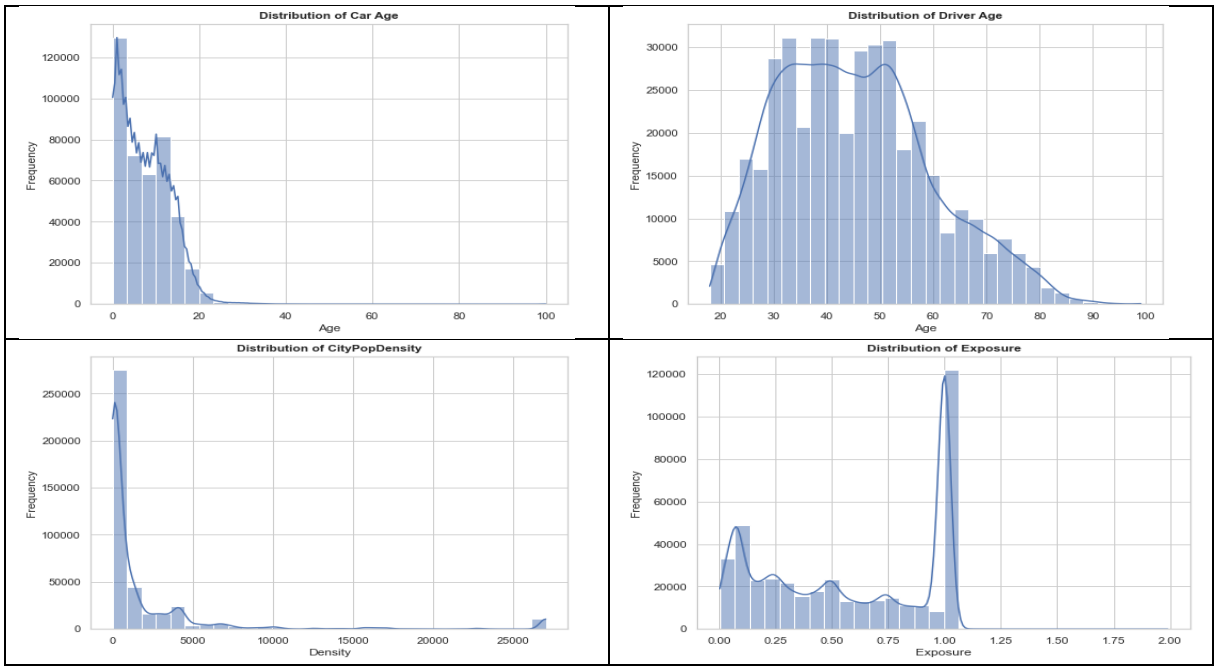
Picture 2. VIF values

In our analysis we also constructed the VIF. High VIF (VIF>5) values indicate that a predictor is highly collinear with other predictors in the model, which means the predictor is not adding much unique information to the regression model. But in our case we can conclude that there is no correlation between the predictor variable and the other variables, meaning there is no inflation in the variance.



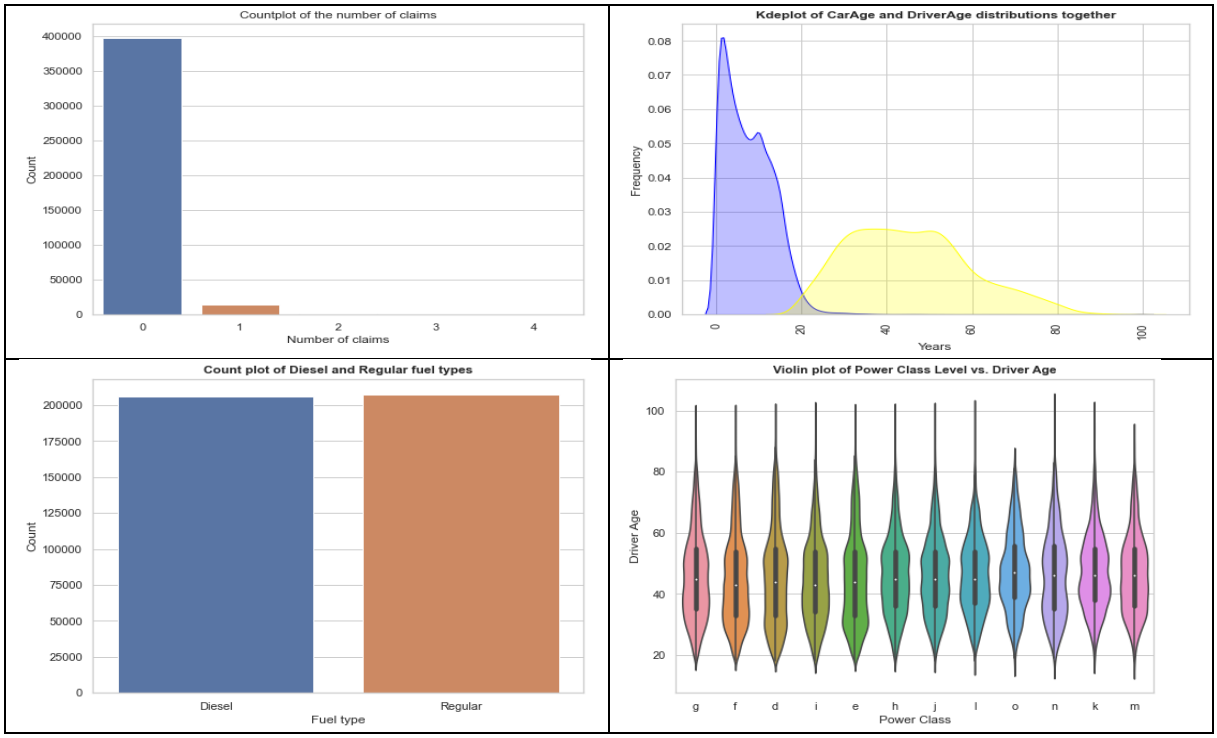
Picture 3. Plots between different variables

The picture above visualizes the plots of dependencies between variables in our data.



Picture 4. Different plots

The above table contains 4 plots. On left upper plot we can observe that most cars are new or relatively new. The top right plot shows that most drivers' age is between 30 and 55 years old. On the lower left plot we can see the distribution of CityPopDencity. We can conclude from the chart, that there majority of data lies up to 5000 points. The lower right one is about the distribution of exposure, from it we can conclude that the median exposure of policies in our data is 1.

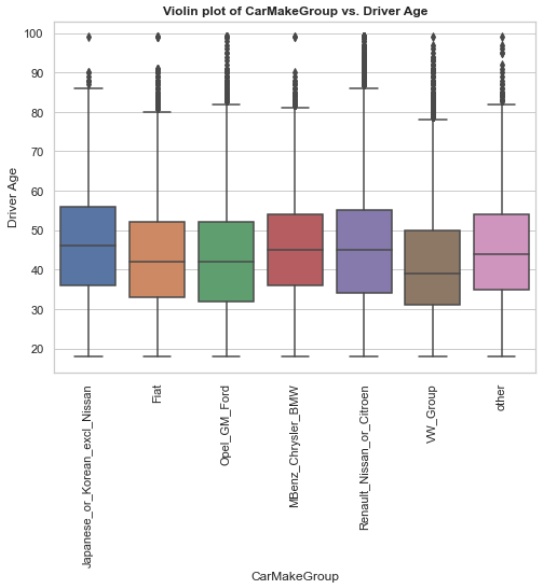


Picture 5. Different plots

The above able contains 4 plots. From the upper left chart which is about the number of claim we can conclude that the majority of policy buyers did not lodge a claim at all. From the upper

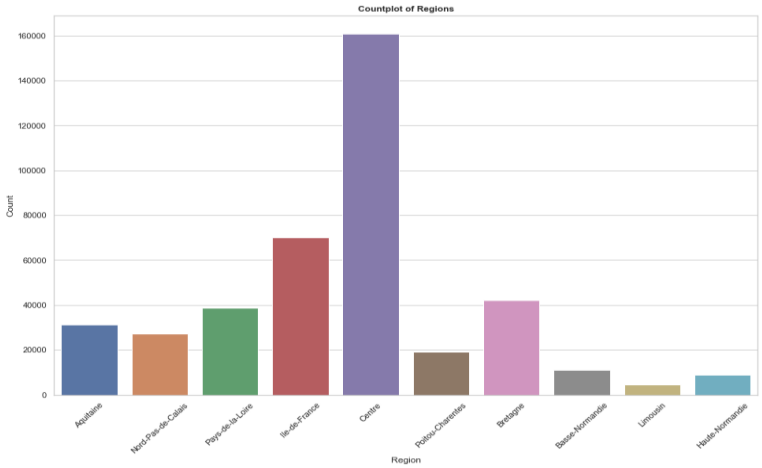
right chart above we can conclude that the distributions of car age and driver age are significantly different. From the bottom left count plot we can see that the number of cars that use diesel as a power source does not differ from the number of regular cars that use petrol. And on the right bottom chart is a violinplot showing the distributions of different values in the power class variable.

The next chart shows the distribution of driver ages for different car make groups. Some groups, like Japanese_or_Korean_excl_Nissan and Renault_Nissan_or_Citroen, have wider age ranges, while others, such as MBenz_Chrysler_BNW, have more concentrated distributions.



Picture 6. Box plot of CarMakeGroup vs. Driver Age

Looking at plot below, we can conclude that the most policies come from Centre region (about 160 000). Second biggest region by number of insurance policies is IIe-de-France (about 70 000).



Picture 7. Histogram of regions

Building the segmentation model

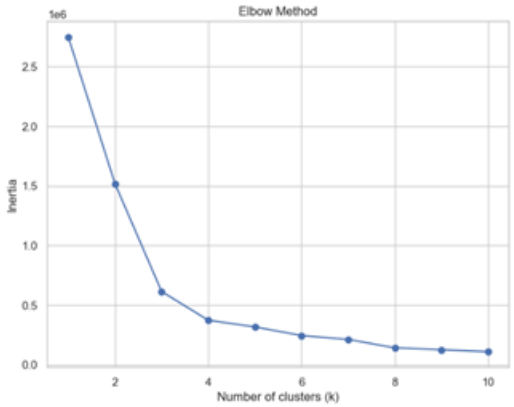
It is necessary to convert categorical features into numerical form, since most machine learning algorithms work only with numerical data. Next, we standardized the data necessary to bring

all the features to the same scale. This is especially important if the signs have different scales. In order for the original type and scale not to change the contribution to the model, and for all functions to be clear, we performed these steps.

For successful segmentation, it is necessary to choose the optimal number of clusters correctly. In this work, we have used various methods.

Since the data set was quite extensive, it was necessary to do a principal component analysis (PCA) with two components to reduce the dimension for visualizing multidimensional data, reduce computational complexity in data processing and remove redundant features.

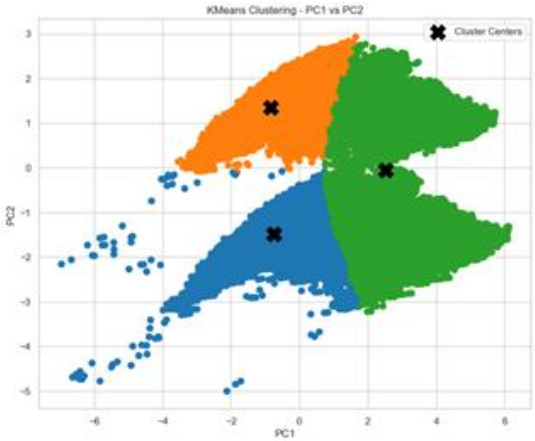
The elbow method displays inertia (the sum of the squares of the distances from points to their nearest cluster center) depending on the number of clusters k.



Picture 8. Elbow method plot

The optimal number of clusters is located at the point where the graph forms an "elbow" — that is, at the point of a sharp decrease in inertia, followed by a slowdown, in this case k = 3. The subsequent addition of clusters may lead to overfitting of the model or minor changes.

Using the K-means algorithm, the division into 3 clusters was visualized, as determined by the elbow method. Each data point receives a cluster label indicating which cluster it belongs to. The black marks indicate the average positions of the points of each cluster in the space of the main components.



Picture 9. K-means clustering

The graph shows that the data is divided into three distinct clusters, each of which is colored in a separate color. This means that the data has been divided into groups, which is probably a good result. The more separated the clusters are and the less overlap they have with each other, the better the result.

Another method, Silhouette Score, indicates the quality of clustering by determining the similarity with its own cluster as opposed to other clusters. Maximizing the average silhouette score from the optimal number of clusters indicates the clearest definition.

- Silhouette Score for 2 clusters: 0.50857
- Silhouette Score for 3 clusters: 0.61009
- Silhouette Score for 4 clusters: 0.61501
- Silhouette Score for 5 clusters: 0.56655

The highest Silhouette Score is achieved at k=4, which means that four clusters provide the best quality of data separation. However, at k=3, Silhouette Score is also high (0.61009) and can be considered a reasonable choice. Further values still indicate a fairly good clustering, but it is lower than for 4 clusters. This may mean that adding more clusters leads to some deterioration in the separability of objects, possibly due to an increase in the number of clusters, which leads to a more complex structure and may lead to overfitting.

To determine how much each group contains elements of only one true category, the cluster homogeneity was calculated. Despite the fact that 3 and 4 showed the same homogeneity. The optimal cluster value is 3, as it ensures maximum homogeneity (equal to 0.9540). Increasing the number of clusters to 4 or 5 does not improve homogeneity and may be redundant.

```
Homogeneity for 2 clusters: 0.058
Homogeneity for 3 clusters: 0.954
Homogeneity for 4 clusters: 0.954
Homogeneity for 5 clusters: 0.928
```

Picture 10. Homogeneity of clusters

Business analysis

Cluster statistics from cluster analysis, where three clusters (cluster 0, cluster 1 and cluster 2) are characterized by six variables in tabular form:

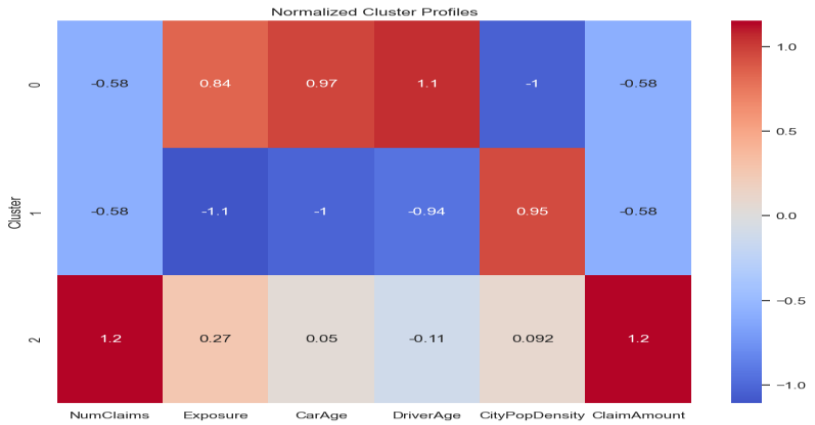
Variable	Cluster 0	Cluster 1	Cluster 3
NumClaims	0.00	0.00	1.05
Exposure	0.89	0.25	0.70
CarAge	8.91	6.29	7.70
DriverAge	49.65	41.49	44.90
CityPopDensity	910.60	2933.97	2061.83
ClaimAmount	0.00	0.00	2239.37

Table 1. Cluster statistics

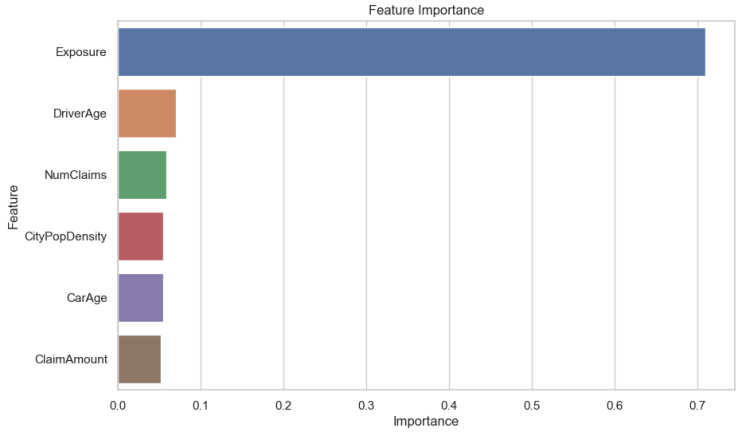
When describing Cluster0, despite relatively higher exposure, which usually causes some risk for a longer period, the persons have no recorded claims. They tend to be older drivers with older cars, living in moderately dense areas. Being generally older and keeping cars longer and have been insured for more months/years without incident. This suggests that they are not at risk.

Persons in Cluster1 also have no claims, but because of other reasons, than in case of Cluster1. Mostly because of short coverage duration (their policies haven't run as long). They are younger, have newer cars, and reside in the densest urban areas, which is seen by much city population density characteristic. Their zero-claim profile might reflect simply being insured for a shorter period of time.

The Cluster2 persons have actual claims on record (both frequency and severity). They're not necessarily the youngest or oldest, nor in the densest or least dense regions, but their average claim cost is significant.



Picture 11: Heatmap of normalized Cluster Profiles



Picture 12. Feature importance

From the bar plot we can clearly see that Exposure is by far the most important predictor. In other words, how long the policy has been in force is a major signal for determining membership in a risky vs. non-risky segment. Other features are considered as almost equally important to each other.

Alternative clustering methods

As for alternative method of clustering we decided to choose DBSCAN algorithm, applying it on the same variables, but reducing the initial dataset, taking 25% of the dataset for initial analysis. Therefore, we use 75% of the sampled data for training and 25% for testing. As the sample is taken randomly, it will define the population.

Key concepts of DBSCAN are as follows: a point is considered a core point if there are at least 15 points (including itself) within its eps-radius neighborhood. Core points form the "dense"

regions of the dataset. A point, for example, p, is directly density-reachable from a core point q if p is within the eps = 1.5 distance of q. Two points p and q are density-connected if there exists a chain of points where each point is directly density-reachable from the next. Points that are not density-reachable from any core point are classified as noise and pointed as a **Cluster -1**.

We initialize parameters by setting the two key parameters: eps (radius of neighborhood) and min_samples (minimum number of points required to form a dense region). Then algorithm classifies points by identifying if the point is a core point (if at least min_samples points are within eps).

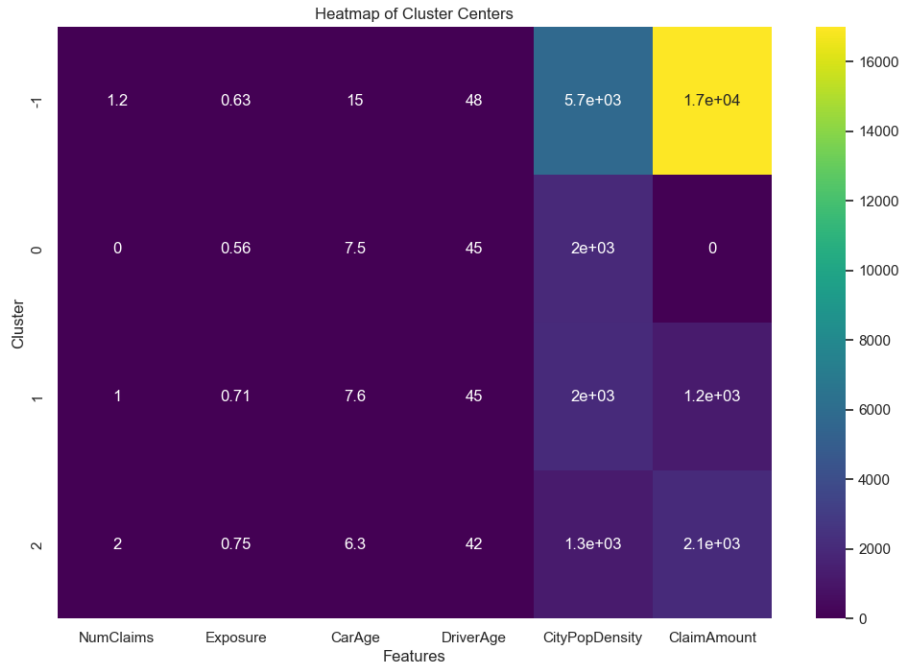
If the point is a core point, all points in its eps neighborhood are added to the cluster. It recursively checks all the neighbors of core points and include them in the cluster and continue until no more points can be added to the cluster.

Unlike K-Means, DBSCAN doesn't require you to specify the number of clusters in advance. It can identify and exclude noise points, which makes it fast for real-world datasets.

The results are as follows:

Variable	Cluster 0	Cluster 1	Cluster 3
NumClaims	0.00	1.00	2.00
Exposure	0.55	0.70	0.75
CarAge	7.49	7.63	6.33
DriverAge	45.28	44.92	41.77
CityPopDensity	1988.27	2018.66	1251.84
ClaimAmount	0.00	1213.83	2055.67

Table 2. DBSCAN cluster statistics



Picture 13. Heatmap of cluster centers

Cluster 0: This group has no recorded claims, medium driver and vehicle ages, and middle-of-the-road city density. They appear to be the lowest-risk cluster in this segmentation.

Cluster 1: They average about 1 claim, with a moderate claim severity around 1.2 k. Compared to Cluster 0, they have similar demographics/car ages but actually do file claims, albeit not with extremely high severity.

Cluster 2: This cluster exhibits a higher claim frequency (2 claims per policy on average) and larger than medium claim amounts. Compared to Cluster 1, they make more claims (twice as many) and pay out more. They also skew slightly younger (drivers around 42) and have somewhat newer cars (6 years).

DBSCAN also provides us with one new cluster, Cluster -1:

Variable	Cluster -1
NumClaims	1.217391
Exposure	0.627971
CarAge	15.108696
DriverAge	47.768116
CityPopDensity	5722.666667
ClaimAmount	16987.905797

Table 3. DBSCAN cluster -1 statistics

As mentioned before, DBSCAN explicitly set aside “extreme” or “sparse” observations (very high ClaimAmounts and older cars/drivers in dense areas) as “noise“. By contrast, k-means force every point into one of the main clusters, even if some are outliers.

Overall, DBSCAN is revealing that a few observations are so extreme in claim severity that they do not fit naturally into the main “core” clusters. This more nuanced segmentation could be valuable from a risk-management standpoint, because we can now see and handle “true outliers” separately, rather than averaging them into one big cluster.

Conclusion

The second ABA project consisted of several stages. In the first part, we described the dataset, did the exploratory analysis including handling missing observations, encoding some features, building the confusing matrix and visualizing the relations between variables. On the basis of this analysis some features were transformed. In the next step, we created the PCA based on the variables and used different methods to determine the optimal number of clusters which in our case was 3. Then we trained the K-means algorithm on K=3 clusters.
Form business analysis it appeared, that Clusters 0 and 1 are both “no-claim” groups, but for different reasons:

- Cluster 0 drivers are generally older, keep cars longer, and have been insured for more months/years without incident.
- Cluster 1 drivers are younger, drive newer cars, and have such a short exposure that they simply haven’t had time to file claims (or are less frequently on the road).

Cluster 2 clearly stands out because they do file claims, have moderate coverage periods, and produce significant average claim amounts.

To make segments more interpretable and specific additional exercise is done using wider range of variables. Both approaches to segmentation and output clusters are visualized and described from business point of view. Finally, the alternative clustering method DBSCAN is applied to compare its results to K-means, which appeared to be slightly different, but also reasonable.