# 1. Introduction

In the face of increased market competition, companies are looking to allocate their resources at the most lucrative customer groups (Epetimehin, 2011). Similar to other industries, insurance companies are also reshaping their market strategies to effectively target the customers segments that complement their profitability. The study conducted by Biswamohan and Bidhubhusan (2012) found that in developed countries the growth of insurance market is relatively low, while the competition is very high. Therefore, besides developing innovative products and services, insurance companies are focus on better marketing penetration strategies. However, to formulate such strategies with optimal use of their resources, insurance companies require better understanding of the customer groups within the industry. Sophisticated customer segmentation allows the companies the ability to segment their customers, understand customer demands of each segment, identify profitable segments and formulate market strategies to attain and retain customers from those desired profitable segments (Soeini and Fathalizade, 2012).

This project uses various clustering techniques to create customer segments or clusters for a fictional insurance company. Based on the identified clusters the report also recommends potential marketing strategies that can be adopted by the insurance company.

# 2. Methodology

There are different approaches to Data Mining. For this particular customer segmentation analysis, Data Mining and Knowledge (KDD) Process is adopted. The process is shown in the figure below.

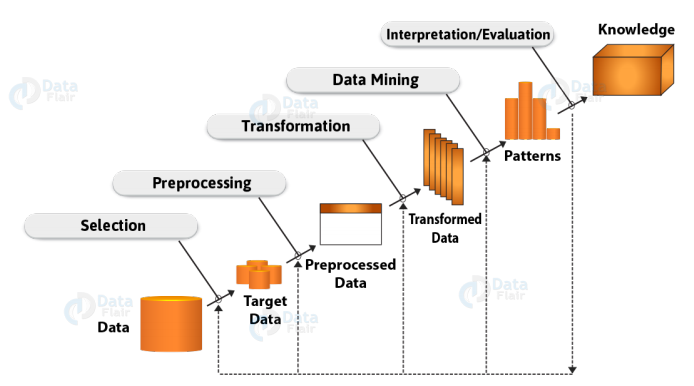


Figure 1 KDD Process

Since, the data is already selected for this analysis, the following chapters discusses how the data were processed, transformed, mined (clustered), interpreted and evaluated. Finally, the conclusion outlines the knowledge gathered from the analysis of the dataset.

**Python** was the preferred programming language, **Spyder** was the preferred environment. The group collaboration for data analysis was undertaken using **GitHub.[[1]](#footnote-1)**

# 3. Data Preparation

Data Preparation refers to the process of transforming raw data into useful and efficient data that can be analysed to gather information (Roh et al., 2018). Data preparation includes cleaning and transforming the data, dealing with inconsistent data and finally, handling the missing values.

## 3.1 Sample Overview

The sample used for this analysis has 10,296 rows with 14 columns. The columns or variables of the sample dataset are shown in the table below. Also, to note that current year of the database is 2016, which is used throughout this analysis. For instance, if a customer’s birthday is 1990, the age is presumed to be 26 (= 2016 – 1990).

|  |  |
| --- | --- |
| **Variables (Columns)** | **Description** |
| Customer Identity | Unique ID of customers |
| First Policy’s Year | The first year as customer |
| Birthday Year | The birth year of the customer |
| Educational Degree | Level of academic education of the customer. |
| Gross Monthly Salary | Customer’s monthly salary |
| Geographic Living Area | Geographic area where the customer resides |
| Has Children (Y=1) | Whether the customer has children or not |
| Customer Monetary Value | Life Time Value of the customers.  *Life Time Value = (Annual Profit from Customer) x (Number of years that they are a customer) - (Acquisition Cost).* |
| Claims Rate | Rate of claims by the customers in the last two years. |
| Premiums in LOB: Motor | Annual (2016) Premiums paid for Motor Insurance |
| Premiums in LOB: Household | Annual (2016) Premiums paid for Household Insurance |
| Premiums in LOB: Health | Annual (2016) Premiums paid for Health Insurance |
| Premiums in LOB: Life | Annual (2016) Premiums paid for Life Insurance |
| Premiums in LOB: Work Compensations | Annual (2016) Premiums paid for Work Compensations |

Table 1 Variables in the dataset

Summary statistics of the dataset is shown in the table below.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Variables** | **Data Types** | **Mean** | **Min** | **Max** | **Distinct Count** | **Missing Values** |
| Customer Identity | int64 | - | - | - | 10296 | 0 |
| First Policy’s Year | int64 | 1991 | 1974 | 53784 | 27 | 30 (0.3%) |
| Birthday Year | float64 | 1968.00 | 1028 | 2001 | 69 | 17 (0.2%) |
| Educational Degree | Category | - | - | - | 5 | 17 (0.2%) |
| Gross Monthly Salary | int32 | 2506.68 | 333 | 55215 | 3566 | 36 (0.3%) |
| Geographic Living Area | Category | - | - | - | 5 | 1 (<0.1%) |
| Has Children (Y=1) | Category | - | - | - | 3 | 21 (0.2%) |
| Customer Monetary Value | float64 | 177.89 | -165680.42 | 11875.89 | 7012 | 0 |
| Claims Rate | float64 | 0.74 | 0 | 256.2 | 165 | 0 |
| Premiums in LOB: Motor | float64 | 300.47 | -4.11 | 11604.42 | 1951 | 34 (0.3%) |
| Premiums in LOB: Household | float64 | 210.43 | -75 | 25048.8 | 1061 | 0 |
| Premiums in LOB: Health | float64 | 171.58 | -2.11 | 28272 | 1007 | 43(0.4%) |
| Premiums in LOB: Life | float64 | 41.86 | -7 | 398.3 | 612 | 104 (1%) |
| Premiums in LOB: Work Compensations | float64 | 41.28 | -12 | 1988.7 | 899 | 86 (0.8%) |

Table 2 Summary Statistics of the variables (‘-’ indicates Not Applicable)

It is to note that distinct count for categorical variables such as ‘Has Children’ includes ‘Missing Value’ to be a distinct count. Therefore, disregarding missing value count, distinct variables will be (n-1, where n is missing value), and for ‘Has Children’ distinct value count is 2. The table also indicates high number missing values for Life insurance premiums. While, unusual values can be observed for Birth year (min Birth year being 1028) and First Policy’s year (max first policy year being 53784).

## 3.2 Data Cleaning

3.2.1 Variables

For the ease of analysis, the variable names were transformed and shown in the table below.

|  |  |
| --- | --- |
| **Variables** | **Transformed Variables** |
| Customer Identity | ID |
| First Policy’s Year | First\_Policy |
| Birthday Year | Birthday |
| Educational Degree | Education |
| Gross Monthly Salary | Salary |
| Geographic Living Area | Area |
| Has Children (Y=1) | Children |
| Customer Monetary Value | CMV |
| Claims Rate | Claims |
| Premiums in LOB: Motor | Motor |
| Premiums in LOB: Household | Household |
| Premiums in LOB: Health | Health |
| Premiums in LOB: Life | Life |
| Premiums in LOB: Work Compensations | Work\_Compensation |

Table 3 Transforming Variable Names

Also, to note, ‘ID’ variable was set as the index of the dataset.

3.2.2 Inconsistent Data

Table 2 indicated a number of inconsistencies with the data. Along with these two, the following inconsistent data were identified and handled:

* Any Customer with birth year before 1900 (df["Birthday"] < 1900) were identified and assigned a NaN value.
* Any Customer with First Policy after 2016 (df["First\_Policy"] > 2016) were identified and assigned a NaN value.
* Any Customer with First Policy before their birthday (df["First\_Policy"] < df["Birthday"]) were identified and assigned a NaN value.
* Duplicate records were identified, and kept just one identical record.

3.2.3 Missing Values

Missing values are common trait in large dataset such as the one used in this analysis. As shown in table 2, some of the variables such as Life premiums (1.0%) have missing values. Besides, the already existing missing values in the dataset, the process of handling inconsistent data further created more missing values. Disregarding missing values is problematic, as Little and Rubin (2014) argue that by dropping missing values from a dataset, one creates an analysis that is reflected of a selected sample of the dataset, rather than the entire dataset. Furthermore, by dropping missing values, especially mechanical (unrecorded by mistake) missing values, companies can be deprived of valuable information and insights (Provost and Fawcett, 2013). Therefore, missing values for this analysis are not disregarded, rather imputed. Imputation was performed following input space reduction and outlier handling.

# 4. Data Pre-processing

In this step, data are transformed to reduce noises in the dataset, to reduce the size of the input space and to normalise the data.

## 4.1 Reduce the Size of input space

The reduce of input space is essential in data mining for not to reduce computational time and costs, but also to reduce the impact of Curse of Dimensionality. Curse of Dimensionality refers to *‘various phenomena that arise when analysing and organising data in high-dimensional spaces that do not occur in low-dimensional settings’* (Bellman, 2003, p.10). Among other issues and problems, when the dimensionality of the data increases, data become sparse as the volume of the space increases. Such sparsity is highly problematic for results that are statistically significant. Therefore, to limit the curse of dimensionality, dimensionality reduction or reducing size of input space is often necessary.

There are a number of different techniques available to reduce dimensionality. These techniques fall into two categories: *feature extraction* and *feature selection* (Shmueli et al., 2016). Feature extraction construct new features that are usually combinations of original features. Examples of such techniques include Principle Component Analysis (PCA). Feature selection on the other hand involve techniques that select subset of original features to maximise relevance and minimise redundancy to the target. Tang et al. (2014) argue that in feature extraction techniques linking the new features with original features is very difficult. Furthermore, newly extracted features have no physical meaning. On the contrary, Feature selection maintain physical meanings of the original features, allowing better readability and interpretability. Given, this analysis aims to find desirable customer clusters or segments, identifiable attributes or features are therefore important. Hence, feature selection is preferred.

In line with the recommendations made by Tang et al. (2014), to select the required features for this analysis, each feature is examined to determine the usability in the outcome.

4.1.1 Feature Transformation

Two of the variables: ‘Birthday’ and ‘First\_Policy were transformed into

* ‘Age’ (Birthday – 2016)
* ‘Customer\_Years’ (2016 – First\_Policy).

The transformation is done as the new features are able to provide more information about customers that are relevant for the analysis.

4.1.2 Features Selection

In order to reduce features, Pearson’s Correlation is calculated between the variables. The Correlation Matrix with Heatmap is shown in the figure below.

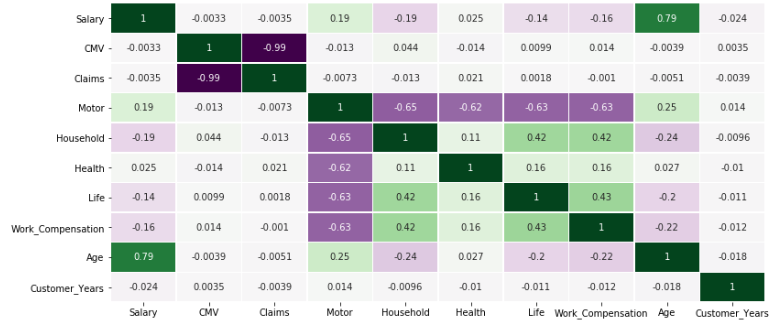


Figure 2 Correlation Matrix with Heatmap

The threshold for correlation between two variables that deem the association to be highly correlated can slightly vary. For instance, Albon (2017) proposes the threshold to be 0.95, indicating if two features have correlation higher than 0.95, one of the features need to be dropped. Vishal (2018) on the other hand proposes this threshold to be 0.90.

From the heatmap in the above figure, it is clear that Claims and CMV (Customer Monetary Value) are highly correlated (-0.99) albeit negatively. This indicates when claim rates increase customer value decreases and vice versa. Since, the correlation meets all the thresholds discussed below, one of these features should be dropped.

Tang et al. (2014) proposes dropping features that provide less information. While, Claims provides information regarding the amount of insurance paid by the company relative to the premiums charged in the last two years, CMV provides Lifetime value of a customer. Therefore, CMV is persevered and Claims is disregarded.

## 4.2 Noise Reduction

Noise refers to any undesired disturbance in the relevant information, as this disturbance affects a signal and distort any information carried by the signal (Garcia et al., 2014). Noisy data includes errors and outliers in a dataset. According to Shmueli et al. (2016), while outliers are most often noise in the data, not all outliers are noises and not all not all noises are outliers. But for the purpose of this analysis, all outliers are deemed noises and therefore removed.

An Outlier is a data point that differs significantly from other observations (Shmueli et al., 2016). Outliers can generate from incorrect measurements, data collection errors and also from unusual but correction situations. Irrespective of the source, outliers can cause significant problems in statistical analysis (Garcia et al., 2014). Therefore, for this analysis, all outliers are removed from the dataset.

There are a number of different measures to detect outliers. Various techniques use Z-score, Inter-quartile range (IQR) score, 3 standard deviations to detect outliers. Clustering techniques such as Self-Organising Map (SOM) and K-Means clustering are also used to detect outliers. For this analysis, outliers are detected using IQR score.

According to Raschka (2016), a data point (x) is considered to be outliers if

x < Q1 - 1.5 \* IQR or x > Q3 + 1.5 \* IQR

where,

Q1 = first quartile

Q3 = third quartile

IQR = Inter Quartile Range

All identified outliers were assigned NaN values

## 4.3 Data Imputation

4.3.1 Imputation with Random Forests

Random Forests (RF) are ensemble learning method which by constructing multitude of decision trees is able to perform classification, regression and other tasks (Ho, 1998). According to Tang and Ishwaran (2017), RF has been gaining traction for missing value imputations because:

1. RF can handle missed types of missing data,
2. RF can address interactions and nonlinearity
3. RF can scale to high-dimensions while avoiding overfitting, and
4. RF yield measures of variable importance useful for variable selection.

For this project, RandomForestClassifier from sklearn.ensemble was used to impute categorical variables (Children, Education and Area).

4.3.2 Imputation with Zero (0)

There are some missing values for insurance premiums. It is assumed that any record/customer that is missing a particular insurance premium value is not an insurance holder of that policy. Therefore, missing insurance premium values were imputed with Zero (0).

4.3.3 Imputation with Mean

The remaining values were imputed with the mean of the column.

The summary of the imputation for each variable is shown in the table below.

|  |  |
| --- | --- |
| **Variables** | **Imputation Technique** |
| ID | *Set as index* |
| Customer\_Year | Imputed with Mean |
| Age | Imputed with Mean |
| Education | Imputed with Random Forest Classifier |
| Salary | Imputed with Mean |
| Area | Imputed with Random Forest Classifier |
| Children | Imputed with Random Forest Classifier |
| CMV | Imputed with Mean |
| Claims | *Dropped* |
| Motor | Filled with Zero |
| Household | Filled with Zero |
| Health | Filled with Zero |
| Life | Filled with Zero |
| Work\_Compensation | Filled with Zero |

Figure 3 Imputation Summary

## 4.4 Data Normalisation

Data normalisation can have various meanings. In the context of this analysis, Data Normalisation refers to the process of transforming values into shifted and scaled versions with the intention that these values can be used for comparison with other normalised values (Dodge, 2003). In other words, if values of two or more variables have different scales, normalisation creates uniform scale and allow comparison of those variables.

In order to normalise all the numeric variables in this dataset, StandardScaler from sklearn.preprocessing was used. StandardScaler transforms a distribution such that the mean value of the distribution is 0 and standard deviation is 1.

# 5. Cluster Analysis

There are a number of different clustering techniques/algorithms available. In this analysis, the following Clustering algorithms are used:

* Hierarchical Methods (Agglomerative)
* Partitioning Methods (K-means)
* K-modes
* Self-Organising Maps
* Mean-Shift Clustering

## 5.1 Agglomerative Clustering

Agglomerative Clustering is type of Hierarchical Clustering. Hierarchical Clustering builds hierarchy of clusters, and Agglomerative Clustering is a bottom-up approach where each observation is treated as a single cluster at the beginning and then successively agglomerates or merges pairs of clusters until all the observations are agglomerated into a single cluster (Manning et al., 2008). Hierarchical clustering is visualised by Dendrogram, which is a tree diagram showing taxonomic relationships.

The advantages of hierarchical clustering are the ease of implementation and its ability to provide easier understanding of the data. Another advantage is that it does not require the specification of cluster numbers.

The disadvantages include tome complexity, difficulty with large datasets and its inability to undo any previous step.

5.1.1 Application of Agglomerative Clustering in the current dataset

## 5.2 K-means Clustering

K-means clustering refers to the clustering algorithm that partitions n observations into k clusters, where each observation belongs to the cluster with nearest mean (Pelleg and Moore, 2000). This algorithm minimizes within-cluster variances (calculated by Squared Euclidean distances).

The advantage of this partitioning clustering algorithm is that it is easy to implement, guarantees convergence, and scales to large datasets. The disadvantages include high sensitivity to outliers, need to choose k manually, and while this algorithm is better suited for spherical-shaped clusters, it is not better suited for clusters with different densities and complex shapes. Also, K-means is inflexible when it comes to probability of belonging for an observation; an observation either belongs to a cluster or it does not.

5.2.1 Application of K-means Clustering in the current dataset

## 5.3 K-modes

K-modes identifies clusters based on the number of matching categories between two data points (Huang, 1998). K-modes clustering algorithm is an extension of K-means. Rather than using distances used by K-means, K-modes uses dissimilarities, and rather than using means, K-modes uses modes (Huang, 1998). Hence, K-modes is used for clustering categorical data.

5.3.1 Application of K-modes Clustering in the current dataset

## 5.4 Self-Organising Map (SOM)

SOM is a type of artificial neural network that is trained by using unsupervised learning to produce low-dimensional (two or three dimensional) discretized representation of input space of the training samples (Chon and Park, 2008). The use of competitive learning and neighbourhood function makes SOM different from other artificial neural networks. The low dimensional representation of high dimensional input space also makes visualisation easier for SOMs.

The advantages of SOMs include the ease of understanding, and the ease of interpretation. SOMs can also useful for clustering complex and large datasets.

However, there are certain disadvantages. Getting the right data is imperative to develop meaningful clusters. Therefore, if adequate data for each dimension are not available, then, SOM is not a suitable option (Kohonen, 1997). Furthermore, similar grouping can appear on the map distant from each other.

5.4.1 Application of K-modes Clustering in the current dataset

## 5.5 Mean-Shift Clustering

Mean-shift clustering is a sliding-window based algorithm that tries to areas with dense data points. This is also a centroid-based algorithm, and it works by first defining a window, placing the window on a datapoint, then calculating the mean for all points in that window, move the window to the location of calculated mean, and repeating this process until there is a convergence (Carreira-Perpinan, 2015).

The advantage of this clustering is that it only needs single parameter (window size) to be passed, does not require the specification of number of clusters. It is also robust to outliers.

The disadvantages include time complexity, which increases computational costs, and trivial nature of defining window size. Also, there is no direct control over the number of clusters generated by the algorithm.

5.5.1 Application of Mean-Shift Clustering in the current dataset

# References

Albon, C. (2017). *Drop Highly Correlated Features*. [online] Available at: https://chrisalbon.com/machine\_learning/feature\_selection/drop\_highly\_correlated\_features/ [Accessed 7 Jan. 2020].

Bellman, R. (2003). *Dynamic programming*. Courier Dover Publications.

Biswamohan, D. and Bidhubhusan, M. (2012). E-CRM Practices and Customer Satisfaction in Insurance Sector. *Research Journal of Management Sciences*, 1(1), pp.2-6.

Carreira-Perpinan, M. (2015). *A review of mean-shift algorithms for clustering*. [online] Faculty.ucmerced.edu. Available at: https://faculty.ucmerced.edu/mcarreira-perpinan/papers/mean-shift-review.pdf [Accessed 7 Jan. 2020].

Dodge, Y. (2006). *The Oxford Dictionary of Statistical Terms*. Oxford: Oxford University Press.

Epetimehin, F. (2011). Market Segmentation: A Tool for Improving Customer Satisfaction and Retention in Insurance Service Delivery. *Journal of Emerging Trends in Economics and Management Sciences*, 2(1), pp.62-67.

Garćia, L., de Carvalho, A. and Lorena, A. (2013). Noisy Data Set Identification. [online] Available at: https://link.springer.com/chapter/10.1007/978-3-642-40846-5\_63 [Accessed 7 Jan. 2020].

Ho, T. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8), pp.832-844.

Huang, Z. (1998). Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values. *Data Mining and Knowledge Discovery*, 2(3), pp.283-304.

Jørgensen, S. and Fath, B. (2008). *Encyclopedia of ecology*. Amsterdam, The Netherlands: Elsevier.

Kohonen, T. (1998). *Self-Organizating Maps*. 1st ed. New York: Springer-Verlag.

Little, R. and Rubin, D. (2014). *Statistical Analysis with Missing Data*. 2nd ed. New York, NY: John Wiley & Sons.

Manning, C., Raghavan, P. and Schütze, H. (2008). *Introduction to information retrieval*. 1st ed. Cambridge: Cambridge University Press.

Pelleg, D. and Moore, A. (2000). X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In: *ICML ’00 Proceedings of the Seventeenth International Conference on Machine Learning*. pp.727-734.

Provost, F. and Fawcett, T. (2013). *Data science for business*. 1st ed. Sebastopol, Calif.: O'Reilly.

Raschka, D. (2016). *Dixon's Q test for outlier identification*. [online] Available at: http://sebastianraschka.com/Articles/2014\_dixon\_test.html [Accessed 7 Jan. 2020].

Roh, Y., Heo, G. and Whang, S. (2018). A Survey on Data Collection for Machine Learning: A Big Data - AI Integration Perspective. *IEEE Transactions on Knowledge and Data Engineering*, pp.1-1.

Shmueli, G., Bruce, P. and Patel, N. (2016). *Data Mining for Business Analytics: Concepts, Techniques, and Applications*. 3rd ed. John Wiley & Sons.

Soeini, R. and Fathalizade, E. (2012). Customer Segmentation based on Modified RFM Model in the Insurance Industry. In: *Proceedings of 2012 4th International Conference on Machine Learning and Computing*. [online] Available at: https://pdfs.semanticscholar.org/438d/d6605c109fde72d900d9071e16d372e3e433.pdf [Accessed 7 Jan. 2020].

Tang, F. and Ishwaran, H. (2017). Random forest missing data algorithms. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 10(6), pp.363-377.

Tang, J., Alelyani, S. and Liu, H. (2014). Feature selection for classification: A review. In: C. Aggarwal, ed., *Data Classification*. Chapman and Hall/CRC.

Vishal, R. (2018). *Feature selection — Correlation and P-value*. [online] Available at: https://towardsdatascience.com/feature-selection-correlation-and-p-value-da8921bfb3cf [Accessed 7 Jan. 2020].

1. Github repo: <https://github.com/pedromlsreis/paranormal_distributions> [↑](#footnote-ref-1)