Project Report

Automated Underwriting

Pulkit Sheokand

DDA1610146

PG Diploma in Data Analytics

IIIT – B and UpGrad

Abstract

Many life insurance companies are turning to technology to speed and remove cost from the underwriting process. Automated underwriting systems have been developed to reduce the manpower, time and/or data necessary to underwrite a life insurance application, while maintaining the quality of underwriting decisions. Although these systems have been in existence for some time, not much is known about how they are used in the industry.

This report covers an aspect of how we can undertake automated underwriting systems. Life insurers have a strong desire to reduce the time and cost, and increase the consistency of underwriting. While the penetration of automated underwriting systems is still relatively low, life insurers are quite interested in their potential to improve the efficiency of underwriting. Despite some concerns with implementation and maintenance challenges, life insurers who are using automated underwriting believe they are beneficial to their organizations. If given the opportunity to reconsider whether to implement automated underwriting, even insurers who are less satisfied with their own systems would choose a different option, rather than forgo automated underwriting altogether.

Contents

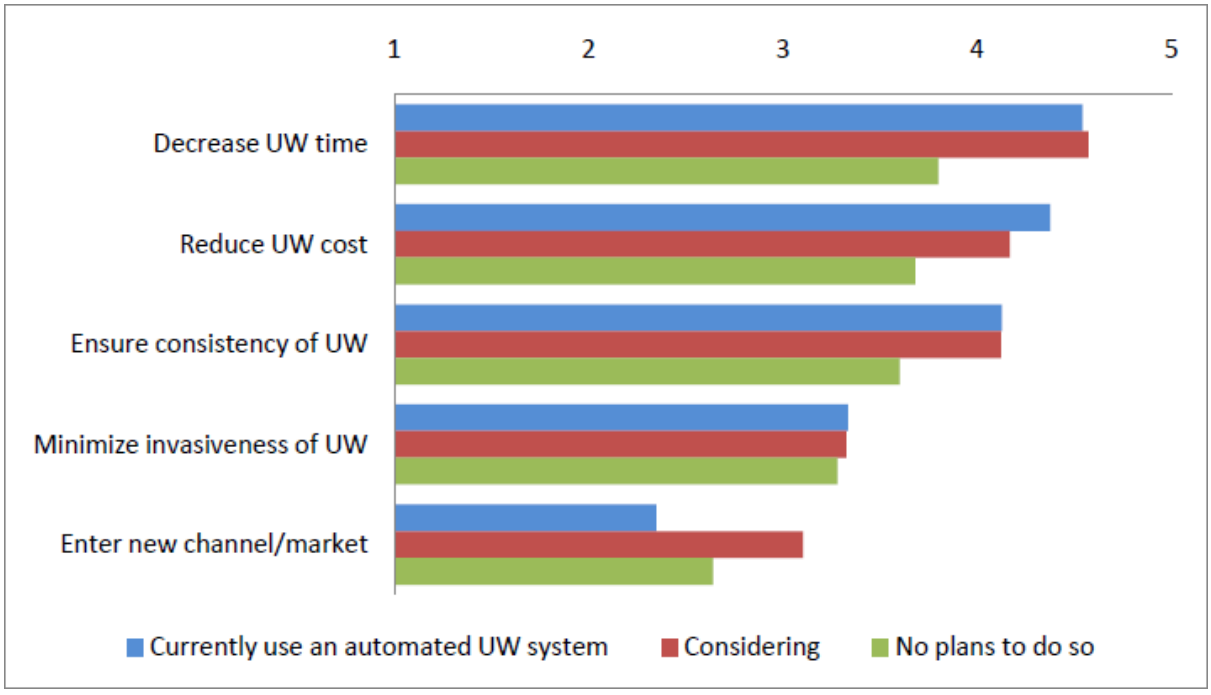
Overview

The penetration of automated underwriting in the insurance industry, currently, is given in the table below:

Percentage of Life Insurers	
Currently using an automated system	29%
Not currently, but considering	49%
Not using, and no plans to consider	22%

The data reveal that while only slightly more than one-quarter of respondents are using automated systems, many more are considering implementing one. While automated systems are still not commonplace, they are also not obscure, and appear to be positioned for growth within the industry. The potential bias respondents may express on this question in particular should be reiterated because it pertains explicitly to the level of interest in automated underwriting.

The benefits of automated system can be seen in the image below:



The difference in the fraction not considering automated underwriting is statistically significant, while the difference in the fraction of large and small insurers using automated underwriting just misses the

95 percent confidence threshold. These data suggest that scale is an important factor in firms' cost-benefit analyses of automated underwriting implementation.

Clearly, these business objectives are of significant concern to life insurers. The top three concerns (cost and time of the process and consistency of the methodology and decisions) register greater than 4 out of 5 on the importance scale. Firms are less, but still somewhat, concerned with minimizing invasiveness of underwriting. Although invasiveness is of more direct concern to the applicant than the insurer, the inconvenience of traditional medical underwriting requirements is often considered a material barrier that inhibits some potential customers from purchasing insurance. Other reasons cited for pursuing automated underwriting systems include responding to a current or projected shortage of underwriters. The order of these concerns is statistically significant.

Not surprisingly, insurers expressing interest in automated underwriting believe the top three objectives are more important than those who do not. It is quite likely that the importance of these concerns contributes significantly to these insurers' interest in automated underwriting, while other life companies feel less pressure to look for solutions. Similarly, these differences are significant.

In addition to those using automated underwriting versus those who are not, actuaries and underwriters also express different opinions about the importance of these business objectives. While the order of significance is the same, underwriters from companies that have not implemented a system rate each objective an average of 0.4 units higher on the 5-point importance scale. However, this difference disappears among companies that are using automated underwriting.

Life insurers surveyed report that difficulty implementing and maintaining automated underwriting systems are the primary reasons they are hesitant. The cost of implementation, limitations on IT resources (both for implementation and ongoing maintenance) and challenges incorporating the technology with current IT infrastructure were all cited as significant barriers. Interestingly, respondents generally do not believe that automated underwriting systems would fail to accomplish their stated objectives. Neither unfamiliarity nor dissatisfaction with automated systems on the market is a major issue. Rather than a lack in confidence in their ability to deliver improvements in underwriting efficiency, logistical issues appear to be the larger impediment to overcome. This trend is apparent even among life insurers who are not currently interested in automated underwriting. In contrast to the importance of each business objective, actuaries and underwriters view the barriers to automated underwriting similarly. The same is true for large and small insurers.

Objective

As an Insurance Carrier, Use more indicators to assess the risk of extending a policy and determining what kinds of coverage to extend so that I can break the market into smaller segments and serve more specialized customer needs.

Data Description

The original data had 14 attributes of categorical and integer type.

5 irrelevant variables were removed.

The data was finally prepared with the remaining 9 variables.

Missing values were negligible in comparison to the whole dataset and were removed.

The current dataset has the following variables:

1. Age
2. Workclass
3. Education
4. Marital Status
5. Occupation
6. Relationship
7. Race
8. Sex
9. Hours per week
10. Income

Age and Hours per week are the integer variables. Outliers have been removed according to the requirement of insurance industry.

Dummy variables have been created for the other remaining factor variables.

The final dataset contains dummy variables and the other two integer variables.

Methodology

The dataset does not have any output variable and needs to be classified into different segments.

This can be done using the method of unsupervised classification.

Unsupervised classification can be done by K-means clustering and Hierarchical clustering.

K-means clustering has been performed on the whole dataset while hierarchical has been performed on a part of the dataset due to size restriction.

K- means clustering

K-means is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early groupage is done. At this point we need to re-calculate k new centroids as barycenters of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more.
Finally, this algorithm aims at minimizing an *objective function*, in this case a squared error function. The objective function

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 ,$$

where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j, is an indicator of the distance of the *n* data points from their respective cluster centres.

Hierarchical clustering

Given a set of N items to be clustered, and an N*N distance (or similarity) matrix, the basic process of hierarchical clustering is this:

1. Start by assigning each item to a cluster, so that if you have N items, you now have N clusters, each containing just one item. Let the distances (similarities) between the clusters the same as the distances (similarities) between the items they contain.
2. Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one cluster less.
3. Compute distances (similarities) between the new cluster and each of the old clusters.
4. Repeat steps 2 and 3 until all items are clustered into a single cluster of size N. (*)

Step 3 can be done in different ways, which is what distinguishes single-linkage from complete-linkage and average-linkage clustering.

In single-linkage clustering (also called the connectedness or minimum method), we consider the distance between one cluster and another cluster to be equal to the shortest distance from any member of one cluster to any member of the other cluster. If the data consist of similarities, we consider the similarity between one cluster and another cluster to be equal to the greatest similarity from any member of one cluster to any member of the other cluster.

In complete-linkage clustering (also called the diameter or maximum method), we consider the distance between one cluster and another cluster to be equal to the greatest distance from any member of one cluster to any member of the other cluster.

Code

```
adult <- read.csv("C:/Users/Pulkit/Desktop/New folder (2)/adult.csv")      #reading dataset


df <- unique(adult)                              #extracting unique rows from dataset

adult1 <- subset(df, select = c(1,5,6,7,8,9,10,13,15))   #selecting columns as per the requirement


sum(is.na(adult1))                               #counting NA values in dataset

sapply(adult1, function(x) sum(is.na(x)))              #counting NA values column wise


quantile(adult1$age, probs = seq(0,1,0.05))            #checking quantile of age variable

adult1 <- subset(adult1, adult1$age<=63)               #removing outliers

quantile(adult1$hours.per.week, probs = seq(0,1,0.05))

adult1 <- subset(adult1, adult1$hours.per.week>=20 |adult1$hours.per.week<=60)


ggplot(adult1, aes(x = adult1$age)) + geom_bar()


table(adult1$workclass)                  #analysing the content of variables

table(adult1$education.num)

table(adult1$marital.status)

table(adult1$occupation)

table(adult1$relationship)

table(adult1$race)

table(adult1$sex)

table(adult1$Income)
```

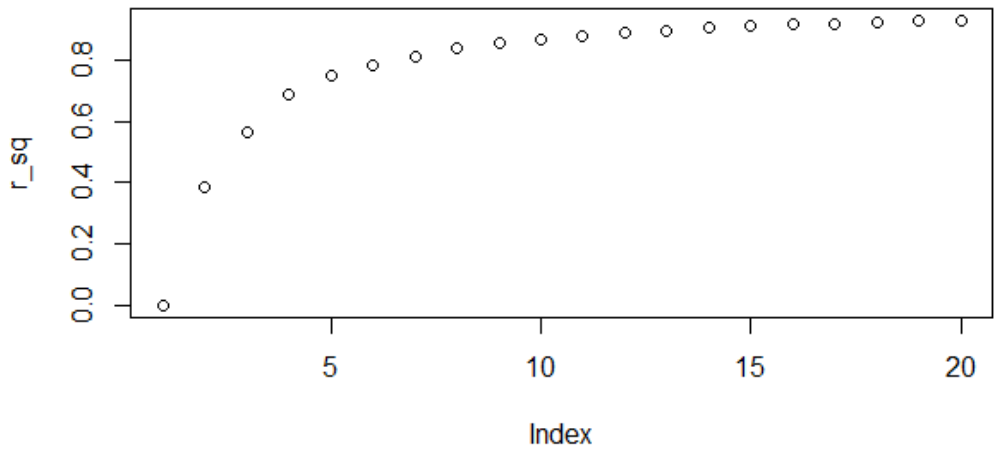
```
dummy <- data.frame(model.matrix(~adult1$workclass, data = adult1))   #creating dummy variables
of categorical variables

dummy <- dummy[,-1]

dummy2 <- data.frame(model.matrix(~adult1$marital.status, data = adult1))

dummy2 <- dummy2[,-1]

dummy3 <- data.frame(model.matrix(~adult1$occupation, data = adult1))

dummy3 <- dummy3[,-1]

dummy4 <- data.frame(model.matrix(~adult1$relationship, data = adult1))

dummy4 <- dummy4[,-1]

dummy5 <- data.frame(model.matrix(~adult1$race, data = adult1))

dummy5 <- dummy5[,-1]

dummy6 <- data.frame(model.matrix(~adult1$sex, data = adult1))

dummy6 <- dummy6[,-1]

dummy8 <- data.frame(model.matrix(~adult1$Income, data = adult1))

dummy8 <- dummy8[,-1]


adult2 <- cbind(adult1[, c(1,9)], dummy8,dummy6,dummy5,dummy4,dummy3,dummy2,dummy)
#creating dataset with dummy variables and integer variables


clus <- kmeans(adult2, centers = 4, nstart = 50)   #creating 4 clusters throught k means

str(clus)      #checking structure of the clusters

r_sq<- rnorm(20)


for (number in 1:20)                          #calculation for checking optimum no. of clusters

{clus <- kmeans(adult2, centers = number, nstart = 50)
```

```
r_sq[number]<- clus$betweenss/clus$totss }


plot(r_sq)                #plot to check the no. of clusters suitable as per data


adult3 <- cbind(adult2, clus$cluster)         #creating dataset with cluster no.

colnames(adult3)[46] <- "ClusterID"

adult_cluster <- group_by(adult3, ClusterID)


tab <- summarise(adult_cluster, mean(age))          #checking mean age of each cluster

tab1 <- summarise(adult_cluster, mean(hours.per.week))   #checking mean hours per week of each
cluster


table(adult_cluster$age, adult_cluster$ClusterID)                #creating table of age vs cluster no.

table(adult_cluster$ClusterID, adult_cluster$hours.per.week)       #creating table of cluster no. vs
hours per week

table(adult_cluster$ClusterID, adult_cluster$adult1.sex.Male)      #creating table of cluster no. vs
male


plot(adult_cluster[c("age", "hours.per.week")], col=adult_cluster$ClusterID)   #plotting age vs hours
per week where colour represents cluster

plot(adult_cluster[c("age", "ClusterID")], col=adult_cluster$ClusterID)

plot(adult_cluster[c("hours.per.week", "ClusterID")], col=adult_cluster$ClusterID)


set.seed(100)

index = sample(1:nrow(adult1),0.2*nrow(adult1))

train = adult1[index,]
```

```
test = adult1[-index,]


adult_dist1 <- dist(train)              #creating dataset with 20% values of original dataset

adult_h1 <- hclust(adult_dist1, method="complete")  #performing hierarchical clustering with
complete method


plot(adult_h1)                          #plotting dendrogram


rect.hclust(adult_h1, h=110, border = "red")

clusterc <- cutree(adult_h1,k=5)              #cutting the dendrogram at height 5


adult_hc <- cbind(train, clusterc)            #merging cluster no. with dataset

colnames(adult_hc)[12] <- "ClusterID"

adult_cluster1 <- group_by(adult_hc, ClusterID)


tab2 <- summarise(adult_cluster1, mean(age), mean(hours.per.week))          #creating table of
cluster no. with mean age and mean hours per week

table(adult_cluster1$age, adult_cluster1$ClusterID)

table(adult_cluster1$ClusterID, adult_cluster1$hours.per.week)

table(adult_cluster1$ClusterID, adult_cluster1$sex)


plot(adult_cluster1[c("age", "hours.per.week")], col=adult_cluster1$ClusterID) #plotting age vs hours
per week where colour represents cluster
```

Output

K-means Clustering



The above plot shows the optimum no. of clusters that can be formed. As the slope of the curve tends to be decreasing at 5, we can form 4 or 5 clusters.

4 clusters have been formed after further analysis.

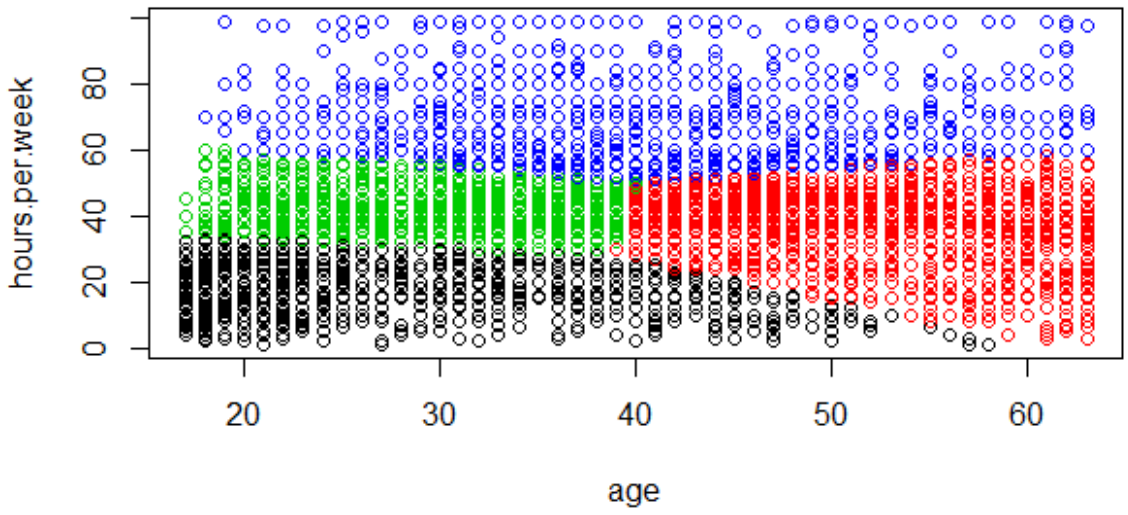
The below given tables show the average age and average hour per week of the 4 clusters that have been formed.

ClusterID		mean(hours.per.week)
1	1	20.62753
2	2	40.54886
3	3	41.66472
4	4	63.82642

ClusterID		mean(age)
1	1	24.19708
2	2	49.25750
3	3	29.90984
4	4	40.13246

In the plot given below, the cluster 1 is depicted by black colour. Red dots represent cluster 2.

Green dots represent cluster 3 and blue dots represent cluster 4.

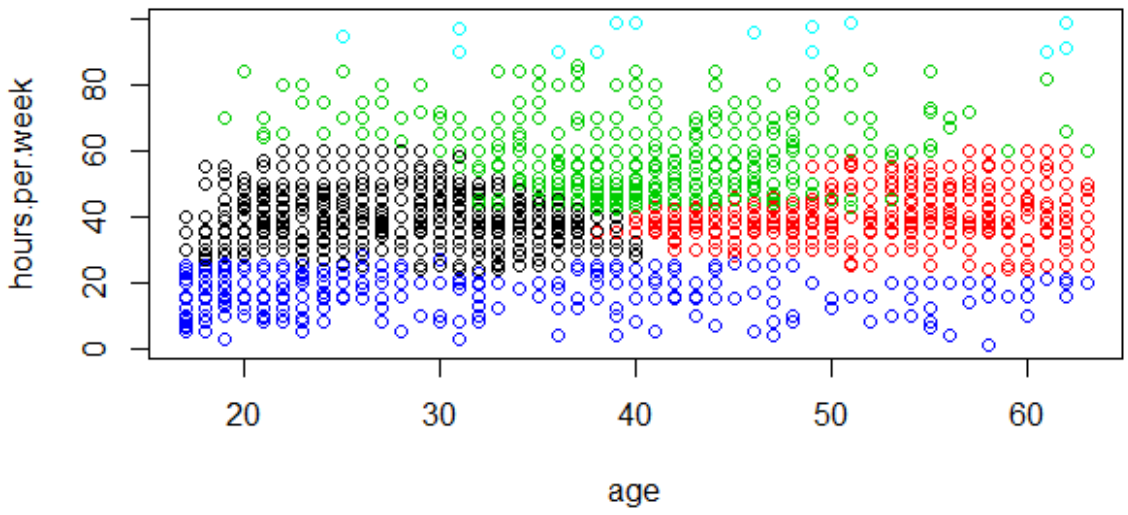


Hierarchical Clustering

The hierarchical clustering was performed on 20 % of the dataset to check the no. of clusters formed by this method. Five clusters have been formed in the process out of which 4 can be easily identified and the 5th cluster represented by light blue colour has little significance in the plot.

Finally, 4 clusters are also supported by this method of clustering.

(Due to size restriction in R, the hierarchical clustering could not be performed on whole dataset)



Conclusion

Based on the results, it can be interpreted that the dataset can be divided into 4 major clusters.

The insurance plans based on the features of the clusters can be designed for proper segmentation of the product.

This will help in easy underwriting of the policies and reduction in cost and time.