MA5800 Foundations for Data Science
Capstone Report
Student: Sacha Schwab

Fit for claim prediction: Preprocessing and analysing raw insurance claim data

1. Abstract

i. This study addresses the topic of whether an investigation using the skills and knowledge acquired in this course suffice to preprocess and analyse an insurance claim dataset, such that it becomes suitable for further building machine learning algorithms to predict whether a travel insurance policy may have a claim during its lifecycle. The data used in this study appears messy and may not be suitable for building such algorithms, however my assumption is that it is in fact.

ii. The purpose of this report is to document the preprocessing and visualisation steps I have undertaken to conclude whether the collected dataset on travel insurance claims is indeed suitable to be further used for machine learning tasks as described above. It provides insight into the applied methods, and how they are computed, and, in its Appendix, the code used.

iii. Methodological approach: As the main question of this report is the analysis of suitability of the data at hand for further machine learning tasks, the analytical part is at the center of the methods used. Prior to the analysis I applied data wrangling methods like identifying and removing unnecessary data, variable transformation and type conversion, and variable selection. Then, I applied a data representation method, which set the ground for the analytical work. Lastly, after drawing the necessary conclusions out of the analytical work, I further preprocessed the data by subsetting it.

iv. Findings: The dataset used in this study is indeed (after preprocessing) likely to be used for further machine learning tasks. A number of variables were eliminated, others transformed.

v. As a conclusion, my assumption that the dataset is likely to be used for further usage in machine learning is confirmed. Moreover, I believe that the work for this study has brought me further in my data science skills. As another benefit, I can apply the methods and code directly in my professional life.

2. Introduction

Working in the financial industry (with experience in insurance), I am interested in knowing and exploring relevant data. Talks with friends working in the data fields of insurance companies have shown me that very often data science methods are still not used for even ‘basic’ tasks such as claim processing, but can improve performance significantly (see Mizgier / Wagner 2018, page 2). Thus, and due to missing approval at my workplace (one reason why I am currently changing job) and thus having had to select a public dataset, I decided to investigate into a public insurance claim dataset.

This study shall provide insight into how a public travel insurance claim dataset can be preprocessed and (visually) analysed so to provide a useful basis for its further usage in e.g. machine learning algorithms. The latter could then, in a corporate context, be used for automatisation of processes that are currently handled manually.

Looking into the first few rows of the dataset my assumption (hypothesis) for this work is that there are only very few of the 10 independent variables that may have an impact on predicting the dependent variable (request for claim).

3. Data

i. Data source is an online published dataset of a third-party travel insurance servicing company based in Singapore, available on Kaggle.
ii. The data appears to be an outflow of an observational study.
iii. There are 63'326 observations included in this dataset
iv. The dataset is structured into 11 variables, of which are 3 categorical nominal, 4 categorical binary, 4 numerical continuous variables.
v. No interventions or pre-processing has been undertaken that precede the ones described below in this report.
vi. Of the 11 variables, 10 are independent variables, 1 (claim status) is the dependent variable.

4. Methods

General remark: For application of the methods in this report I used R Studio Desktop version 1.1.463.
The following was used for **preprocessing** the data:

i. **Data cleaning: Identify and eliminate unnecessary records**
A manual inspection of the data frame in R View (and counting using nrow() function) provides that a number of records provide a negative or null-value for duration (66 rows). Also, the data include records with clients with age 118 (984 rows). Finally, there are rows with empty (string) values (45'107) rows. Removal of the latter cluster seems substantial; however, I prefer clean data; also, an appropriate number of records remain after removal.
These records are eliminated by using the filter() command in R, using a pipe (%>%).

ii. **Variable transformation and type conversion:** Render the data easier to process for data representation and machine learning:
The Distribution Channel variable has the values "Offline" and "Online" which for the purpose of processing can be replaced by 1 and 2, respectively, i.e. converted into integer type. Likewise, for the Agency Type variable values ("Airlines" / "Travel Agency"), with denomination 1 and 2, respectively. The operations in R are executed using the as.integer() function.

iii. **Data representation: Proximity measures**
The dataset contains variables with mixed types, a Gover dissimilarity (distance) matrix is computed and visualised using the daisy() function from package cluster, with the visual result as shown in Figure 1. The yellow colors represent a higher dissimilarity, down to green -> blue -> red with decreasing dissimilarity. As green and blue colors appear to dominate it appears that the similarity between the records is generally not high.

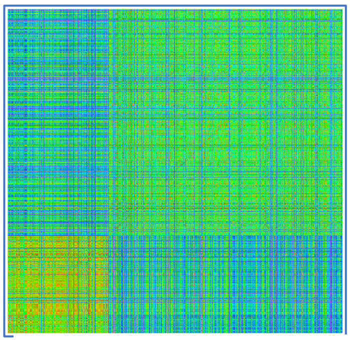

Figure 1: Visual of Gover distance matrix

iv. Variable selection, variable transformation and visualisation (with group-based summarisation): Getting to the ground

As in the task to follow this study (machine learning for prediction of the Claim value) the variables need to be further analysed for a pre-selection.

- The variables Agency, Agency Type, Distribution Type, Product (Name) may have an impact on whether there is (will be) a claim on the policy, as the different agencies, with their distribution channels and preferences for certain products, may attract different client groups, which may, according to their travel styles, cause or get involved in different levels of claims. This is however is to be further analysed.
- The variable Gender variable may lead to a gender-biased result, which may be ethically problematic, and therefore unwanted in the dataset.
  Without further analysing I assume that the duration of the policy may potentially have an effect on whether a claim occurs. The same applies for the destination, and age of the policy holder.
- The two numerical variables Net Sales and Commission however exclusively relate to the performance on the insurance third-party provider side; it would in my view be illogical if these two variables would have an effect on the claim status of a policy. (Note: For e.g. a fraud related study the sales / performance side could be interesting to investigate.)

For further usage of the dataset I therefore select the variables Agency, Agency Type, Distribution Channel and Product Name, Claim, Duration, Destination, Age (using the select() function in R), excluding all other variables.

v. Exploratory visualisation

For the purpose of this report I understand that the input data for a machine learning algorithm shall not be extremely skewed (see e.g. Uguroglu 2013, p. 1) but present at least some variance, i.e. shall not be equal. As the four variables Agency, Agency Type, Distribution Platform and Product are categorical, skewness cannot be taken into consideration. For the variable Duration however, skewness may be relevant.

Categorical variables: To investigate their spread in relation to the target variable Claim, it is first necessary to express the percentage of claims (Yes/No) per category, which is achieved by **group-based summarisation** in R using the dplyr group_by() / summary() functions in a pipe, summarising the percentage of policies with claims in relation to the variables.

The result is shown in Figures 2-5 below. They visually indicate that the values for all four variables appear to have significant differences in percentage of claims provided, with the following standard deviations (computed using the sd() function):

Agency	Agency Type	Distribution Platform	Product
2.255187	0.8383584	1.878616	3.391255

Due to this spread it appears appropriate to say that these 4 categorical variables are suitable for further machine learning tasks as the input

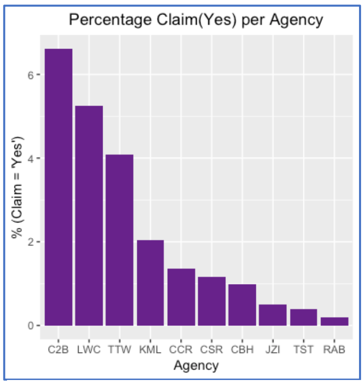

Figure 2: Spread of policies with claims per agency

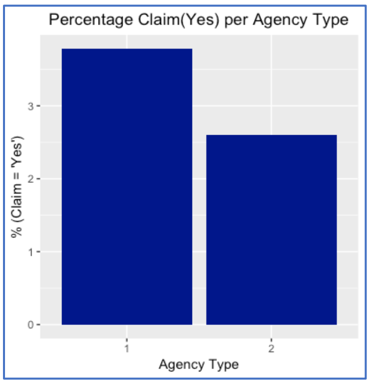

Figure 3: Spread of policies with claims per agency type

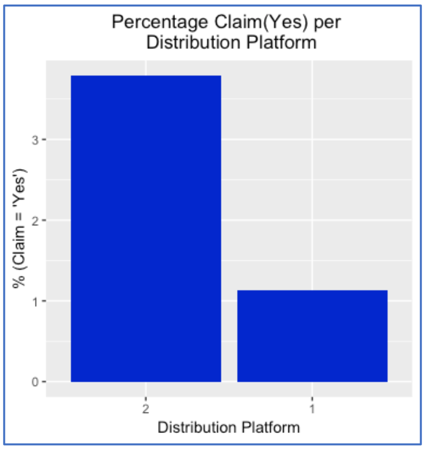

Figure 4: Spread of policies with claims per distribution platform (1=online, 2=offline)

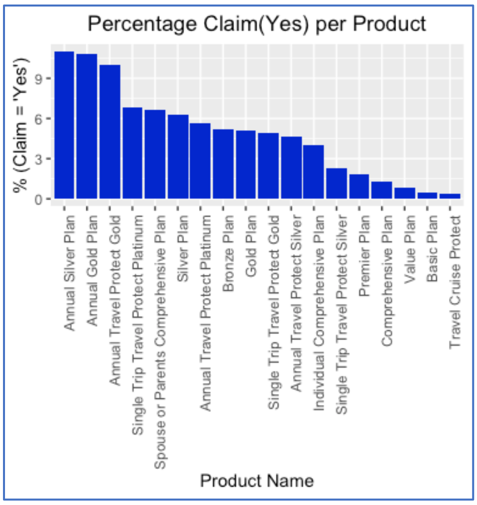

Figure 5: Spread of policies with claims per product

Duration variable:
The visual in Figure 6 shows a significant skewness (positive skew) of the data. It is therefore necessary to normalise the data by performing a log or square root transformation. I decided for a log **transformation** and used the log() function in R for this. The result is shown in Figure 7.

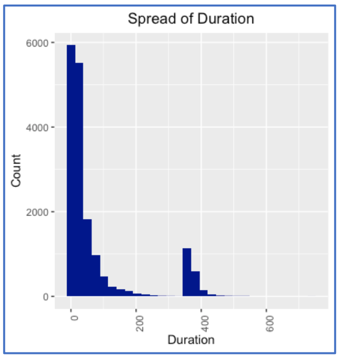

Figure 6: Spread of Duration

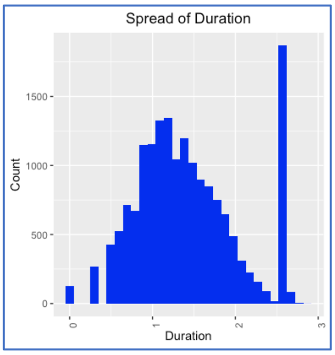

Figure 7: Spread of Duration after Log Transformation

vi. **Data subsetting:**
An analysis of the distribution of the target variable Claim per policy using the hist()

function (see Figure 1) indicates that the data is imbalanced (3.64% "No" matches). Imbalanced data can be of a disadvantage in the scope that our finalised dataset should be used for, i.e. machine learning (see e.g. Brownlee 2015). To mitigate this problem, undersampling instances of the majority class may be useful (see e.g. Soni 2018). I deem a 70/30 spread to be appropriate for many machine learning tasks. The Claim = "No" data of the data frame as above is therefore random sampled (sample_n() function) to the fit this number, then merged with the Claim = "Yes" data (using the rbind function).

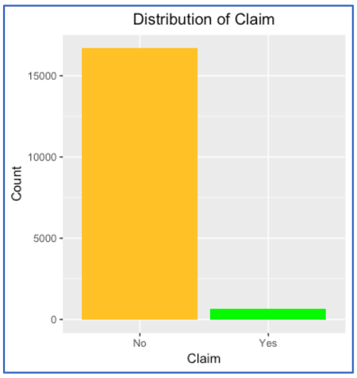

Figure 8: Distribution of Claim overall

5. Results and discussion

i. The data appears in first place messy and with a high number of categorical variables, however the insight gained during this study provides that methods and skills collected during this course suffice to provide a result that may well be used for further machine learning tasks.

ii. A number of variables had (or could) be eliminated, which in its process appeared helpful for my learning process.

iii. The necessary visual exploration of the data is not at a high level of sophistication, but in my view appeared to be appropriate for the task in this study.

iv. I can use the overall methodology used in this study (clean, analyse, further pre-process) and the concrete code provided directly also in my banking job for a number of tasks related to a current project relating to transaction data.

6. Conclusion

In sum, the original expectation / assumption mentioned in the introduction (dataset is likely usable for machine learning tasks) is met, and moreover it a) provided interesting insights into the capabilities of data preprocessing and visual exploration, b) advanced my data science and R skills, and c) can be directly applied in my professional life.

References

Brownlee 2015	Jason Brownlee, 8 Tactics to Combat Imbalanced Classes in Your Machine Learning Dataset Machine Learning Mastery, 2015 URL: https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/
Mizgier / Wagner 2018	Mizgier / Wagner, Zurich Insurance Uses Data Analytics to Leverage the BI Insurance Proposition Researchgate, 2018
Soni 2018	Devin Soni, Dealing with Imbalanced Classes in Machine Learning Towards Data Science, 2018 URL: https://towardsdatascience.com/dealing-with-imbalanced-classes-in-machine-learning-d43d6fa19d2
Uguroglu 2013	Selen Uguroglu, Robust Learning with Highly Skewed Category Distributions Carnegie Mellon University, 2013

Appendix: R Code

```
# Student: Sacha Schwab
# R Code to Capstone Assessment MA5800 Foundations for Data Science

#~~~~~~~~~~~~
# Libraries
#~~~~~~~~~~~
library(dplyr)
library(ggplot2)
library(tidyr)

#~~~~~~~~~~~~~~~~
# Preparations
#~~~~~~~~~~~~~~~~

# Tidy up workspace
rm(list=ls())
# Set work environment
setwd('/Users/sachaschwab/Dropbox/JCU/03 Foundations/Assessments/Capstone')
df=read.csv('travel insurance.csv')

View(df)
# Variables
str(df)
# Number of rows
nrow(df)
# Number of variables
ncol(df)

# Get R version for report
RStudio.Version()
# Remove unnecessary records and variables
# Summary
summary(df)
colnames(df)

# Data cleaning: Identify unnecessary records
nrow(df[df$Duration <= 0,])
nrow(df[df$Age > 99,])
nrow(df[df$Gender == "",])

# Data cleaning: Eliminate unnecessary records
df <- df %>%
  filter(Duration > 0) %>%
  filter(Age < 100) %>%
  filter(Gender != "")

# Type conversion: Convert 2 categorical binomial variable values into integer 1/2 for more efficient processing
```

```
df$Distribution.Channel <- as.integer(df$Distribution.Channel)
df$Agency.Type <- as.integer(df$Agency.Type)

# Investigate whether the data is balanced or imbalanced
df$Claim <- as.factor(df$Claim)
ggplot(data.frame(df$Claim), aes(x=df$Claim)) +
  geom_bar(fill=c('goldenrod1', 'green')) + ggtitle("Distribution of Claim") +
  theme(plot.title = element_text(hjust = 0.5)) + xlab("Claim") + ylab("Count")

# Data representation:
# Investigate proximity measures
library(cluster)
Dist <- daisy(df_sampled, metric = "gower")
Dist <- as.matrix(Dist)
dim <- ncol(Dist)  # used to define axis in image
image(1:dim, 1:dim, Dist, axes = FALSE, xlab="", ylab="", col = rainbow(100))

# Variable selection
df <- select(df, Agency, Agency.Type, Distribution.Channel, Product.Name, Claim, Duration,
Destination, Age)
View(df)
# Variable visualisation and transformation
# Investigate visual spread of variables according to Claim
# Claim per Agency
agency_summary <- df %>%
  group_by(Agency, Claim) %>%
  summarise(n = n()) %>%
  mutate(percClaim = 100 / sum(n) * n) %>%
  arrange(desc(Agency))
#View(agency_summary)
agency_summary <- agency_summary[agency_summary$Claim == "Yes", ]
agency_summary <- arrange(agency_summary, desc(percClaim))
ggplot(agency_summary, aes(x=reorder(Agency, -percClaim), y=percClaim)) +
geom_col(fill="darkorchid4") +
  xlab("Agency") + ylab("% (Claim = 'Yes')") +
  ggtitle("Percentage Claim(Yes) per Agency") + theme(plot.title = element_text(hjust = 0.5))
# Calculate standard deviation
sd(agency_summary$percClaim)

# Claim per Agency Type
agency_type_summary <- df %>%
  group_by(Agency.Type, Claim) %>%
  summarise(n = n()) %>%
  mutate(percClaim = 100 / sum(n) * n) %>%
  arrange(desc(Agency.Type))
View(agency_type_summary)
agency_type_summary <- agency_type_summary[agency_type_summary$Claim == "Yes", ]
agency_type_summary <- arrange(agency_type_summary, desc(percClaim))
ggplot(agency_type_summary, aes(x=reorder(Agency.Type, -percClaim), y=percClaim)) +
geom_col(fill="blue4") +
  xlab("Agency Type") + ylab("% (Claim = 'Yes')") +
  ggtitle("Percentage Claim(Yes) per Agency Type") + theme(plot.title = element_text(hjust =
0.5))
# Calculate standard deviation
sd(agency_type_summary$percClaim)

# Claim per Distribution Platform
distr_summary <- df %>%
  group_by(Distribution.Channel, Claim) %>%
  summarise(n = n()) %>%
  mutate(percClaim = 100 / sum(n) * n) %>%
  arrange(desc(Distribution.Channel))
View(distr_summary)
distr_summary <- distr_summary[distr_summary$Claim == "Yes", ]
distr_summary <- arrange(distr_summary, desc(percClaim))
ggplot(distr_summary, aes(x=reorder(Distribution.Channel, -percClaim), y=percClaim)) +
geom_col(fill="blue3") +
  xlab("Distribution Platform") + ylab("% (Claim = 'Yes')") +
  ggtitle("Percentage Claim(Yes) per \n Distribution Platform") + theme(plot.title = element_text(hjust = 0.5))

ggplot(df, aes(x=Agency, fill=Claim)) + ggtitle("Distribution of Claim \n among Agencies") +
  geom_bar(position="Fill") + theme(axis.text.x = element_text(angle = 90)) +
  theme(plot.title = element_text(hjust = 0.5)) + ylab("Ratio")
# Calculate standard deviation
sd(distr_summary$percClaim)

# Claim per Product
prod_summary <- df %>%
  group_by(Product.Name, Claim) %>%
  summarise(n = n()) %>%
  mutate(percClaim = 100 / sum(n) * n) %>%
  arrange(desc(Product.Name))
View(prod_summary)
prod_summary <- prod_summary[prod_summary$Claim == "Yes", ]
prod_summary <- arrange(prod_summary, desc(percClaim))
ggplot(prod_summary, aes(x=reorder(Product.Name, -percClaim), y=percClaim)) +
geom_col(fill="blue3") +
  xlab("Product Name") + ylab("% (Claim = 'Yes')") +
  ggtitle("Percentage Claim(Yes) per Product") + theme(plot.title = element_text(hjust = 0.5)) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
# Calculate standard deviation
sd(prod_summary$percClaim)

# Skewness of duration
ggplot(data=df, aes(df$Duration)) + geom_histogram(fill="blue4") +
  xlab("Duration") + ylab("Count") +
  ggtitle("Spread of Duration") + theme(plot.title = element_text(hjust = 0.5)) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
# Normalise with log transform
```

```
df$Duration <- log(df$Duration)
# Re-do the histogram
ggplot(data=df, aes(df$Duration)) + geom_histogram(fill="blue2") +
  xlab("Duration") + ylab("Count") +
  ggtitle("Spread of Duration") + theme(plot.title = element_text(hjust = 0.5)) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

# Calculate percentage of Yes (Claims)
100 / nrow(df) * nrow(df[df$Claim == "Yes",])
# Calculate number of Yes (Claims)
total_yes <- nrow(df[df$Claim == "Yes",])
total_yes

# Subsampling with Total Claims=Yes * 10
yes_data <- df[df$Claim == "Yes",]
no_data <- df[df$Claim == "No",]
no_data_sample <- sample_n(no_data, total_yes / 3 * 7)
df_sampled <- rbind(yes_data, no_data_sample)
View(df_sampled)
```