

Machine learning models for claim prediction in car insurance

Barbara GENDRON-AUDEBERT

December 28, 2022



A car facing dangers in a hostile planet, protected with an umbrella, DALL.E generated illustration

Contents

1	Data and related insurance problem	2
1.1	Dataset description	2
1.2	The insurance context	2
2	Preliminary analysis of the data	2
2.1	Outliers and missing values	2
2.2	Data balance with respect to the <code>OUTCOME</code> feature	2
2.3	Remaining anomalies in the data	3
2.4	Select only the relevant features	3
3	Brief description of the models used	3
3.1	Logistic regression	4
3.2	Decision tree classifier	4
3.3	Random forest classifier	4
3.4	XGBoost	5
4	Analysis of the results and conclusion	5
A	Appendix	5

Foreword

This report gives some insights about the use of machine learning techniques in car insurance context. The goal of this work is to leverage data on policyholders related to their driving experience to predict the occurrence of a claim. The dataset used to conduct this analysis comes from this [Kaggle page](#).

In the following sections, we first explore the dataset and describe the related insurance problem in part 1, before diving in a deeper analysis of the provided features in part 2. In part 3, one can find insights about the models used to solve this problem, such as a representation of the decision process leading to the model predictions. Finally, part 4 brings a discussion about the provided results, along with some take-aways of this study about car insurance claim prediction.

All the technical specifications, the data, all the plots and especially the code (in R) are available in the attached files of the report, and online in this [Git repository](#).

1 Data and related insurance problem

1.1 Dataset description

The above mentioned dataset contains 19 pieces of information (for now denoted as *features*) for 10000 policyholders. Among such features, some are closely related to the driving behaviour of the policyholder (driving experience, number of past accidents, ...), whereas other are more related to its living conditions and family (education, income, ...). Lastly, the **OUTCOME** feature indicates whether or not the policyholder already experienced a claim. A complete description of the features is given in table 2 in the appendix.

If most of the features names are clear enough at first sight, some need to be clarified. The credit score reflect the ability for a policyholder to pay for his debts. The higher the score, the more creditworthy the policyholder is. This parameter has been observed to have a significant influence on the premium in car insurance. The feature **DUIS** refers to **DUI**, which stands for *Driving Under the Influence* (whether it be alcohol or drugs).

1.2 The insurance context

Machine learning models can be used in the car insurance context for claim prediction by analyzing historical claims data and identifying patterns and trends that can help predict the likelihood of future claims for given policyholders. This can allow insurance companies to better assess risk and price policies accordingly, potentially leading to cost savings for both the insurer and the insured. In this case, the aim is to predict the **OUTCOME** feature from the others, using some machine learning models.

2 Preliminary analysis of the data

The purpose of this section is to give a more quantitative description of the data and to go over points of attention to ensure proper modeling.

2.1 Outliers and missing values

First, it is usual to compute some basic statistics about each feature on the whole data, such as the minimum and the maximum values, the mean and median. This allows to notice rough anomalies, such as a negative age values. In this dataset, it appear that no anomalies of this type were found.

Descriptive statistics used for this step can be displayed simply using the function `summary()` in R. They are provided along with the number of NA's values for each feature, which corresponds to missing values (NA stands for *Not Available*). Here, the features **CREDIT SCORE** and **ANNUAL MILEAGE** respectively have 982 and 957 NA's values. This represents approximately 1% of the whole data for each variable, that's why we can consider simply delete them. Thus, the remaining dataset contains 8149 rows.

2.2 Data balance with respect to the OUTCOME feature

Insurance claims are rare events, so there is typically not a lot of data available about them. This limited availability of data on insurance claims can lead to challenges when using machine learning models, as it is well-known that such models require a significant amount of data in order to perform well. Therefore, if the data is too imbalanced with respect to the **OUTCOME** (far more "no" than "yes"), the model won't be able to learn well about the claim, which is precisely the point here. Figure 1 shows the value count of 0 and 1, the two possible values for the **OUTCOME** feature. Contrary to what can be expected, claims seem to appear more often than

in real life. This would suggest that data was selected on purpose, which implies that no further preprocessing is required on this point.

2.3 Remaining anomalies in the data

A fairly important step at this stage would be to check the data for more subtle anomalies, not detected at first glance. One typical way to handle that is to look at the values distribution for each feature and check for relevancy according to what the feature describes in real world. Some plots are provided in the appendix: figure 5 shows histograms for quantitative values, whereas figure 6 gives the value counts for categorical features with respect to the **OUTCOME** value. This last graph is also useful to check for balance of data, and gives some insights about the relevancy of the features. From this last exploratory data analysis, no further anomalies have been found.

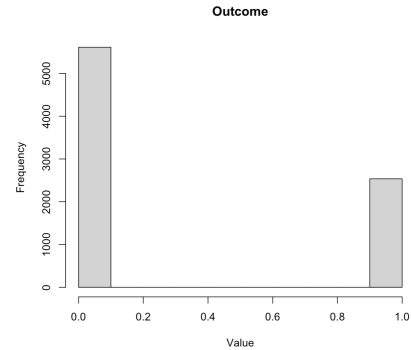


Figure 1: Distribution of the **OUTCOME** feature.

2.4 Select only the relevant features

Amongst the 18 features given to describe each policyholder, all are not bound to be helpful when predicted the claim probability. In addition, in order to meet the constraints of the insurance field, some features must be removed because some regulations prohibit their use. This way, let's remove the **GENDER** feature, along with **RACE** for ethical reasons. Now, what remains is to identify the useful features regarding the objective. A typical method for this is to look at the correlation between the **OUTCOME** and the other feature. A correlation coefficient is a number in $[-1, 1]$. The higher it is in absolute value, the stronger is the link between the two parameters. Figure 2 represents the top 10 features most correlated with **OUTCOME**.

	Outcome		
DRIVING_EXPERIENCE_0-9y	-0.5022658	GENDER_female	0.098575785
AGE_16-25	-0.4339322	POSTAL_CODE	-0.095939353
VEHICLE_OWNERSHIP_0	-0.3862170	INCOME_middle class	0.037907752
VEHICLE_OWNERSHIP_1	0.3862170	AGE_26-39	-0.037331661
INCOME_upper class	0.3361555	EDUCATION_high school	-0.024002644
INCOME_poverty	-0.3345763	RACE_majority	0.013125386
CREDIT_SCORE	0.3214357	RACE_minority	-0.013125386
PAST_ACCIDENTS	0.3127380	ID	0.006222300
SPEEDING_VIOLATIONS	0.2931377	VEHICLE_TYPE_sports car	-0.002804156
VEHICLE_YEAR_after 2015	0.2901292	VEHICLE_TYPE_sedan	0.002804156

Figure 2: Top 10 (left) and flop 10 (right) features most correlated with **OUTCOME**

This correlation analysis gives a first insight of which features can be dropped (those that are less correlated with **OUTCOME**). Finally, the dropped features are **GENDER**, **POSTAL_CODE**, **EDUCATION**, **RACE**, **ID** and **VEHICLE_TYPE**.

3 Brief description of the models used

This part gives short descriptions of the selected machine learning models for this analysis. The task we need to achieve here is a **binary classification** (claim or no claim), which is usually handled by the four models to be covered. In order to cover a relevant panel of methods, the

following models have been chosen because of their simplicity of structure, their explainability and/or the popularity of such approaches.

3.1 Logistic regression

Logistic regression is a type of statistical model that is used to predict the probability of an event occurring. Thus, it is a binary classifier. The outcome is modeled as a function of the features which are used to make the prediction of the claim occurrence. The term *logistic* simply refers to the shape of the function used to predict the outcome. Let f be such function and x the representation of the features, the outcome has the following form:

$$f(x) = \frac{1}{1 + e^{-x}}$$

3.2 Decision tree classifier

A decision tree classifier is a type of machine learning algorithm that uses a tree-like model to make predictions based on the characteristics or features of an input sample. It works by considering each feature in turn and asking a series of questions about the data, with the goal of identifying the feature or combination of features that best splits the data into classes in a way that maximizes the prediction accuracy. After fitting a decision tree classifier on the data, it is possible to plot the decision tree as it is represented in figure 3. For example, this model gives the highest probability of having a claim (0.86) for a driver with less than 10 years of driving experience, who does not own the vehicle and whose vehicle was marketed before 2015.

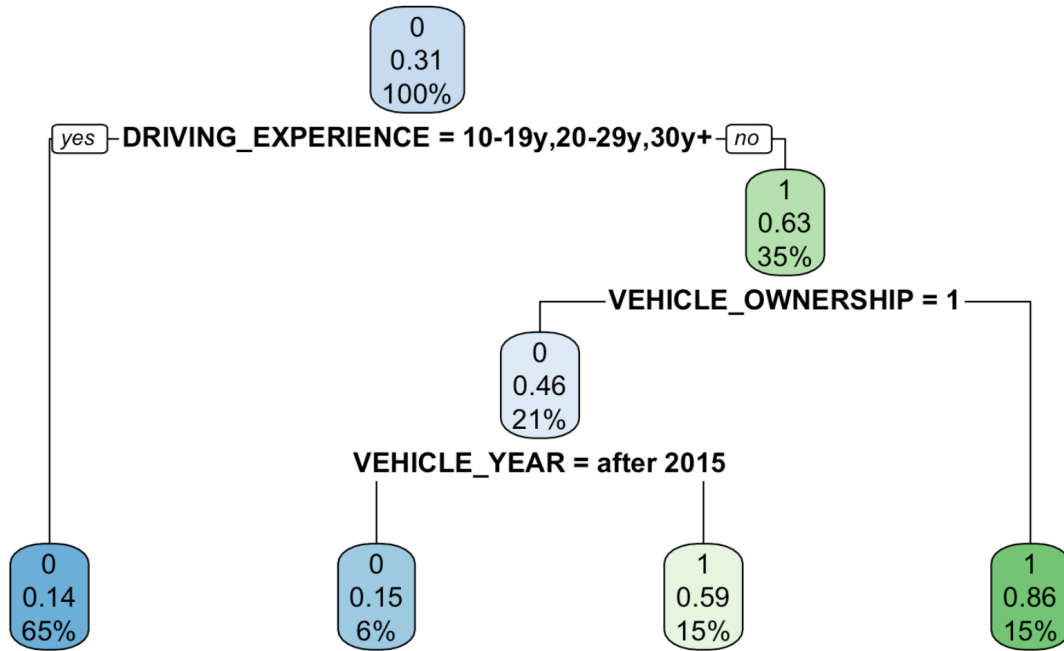


Figure 3: A representation of the decision tree classifier. A positive prediction (1) tend to be green whereas a negative (0) one tend to be blue.

3.3 Random forest classifier

This model is a logical extension of the previous one, in the sense that it contains several decision trees with random initialization. A random forest classifier is often able to make more accurate predictions because it combines the predictions of multiple decision trees, which reduces the variance of the model and helps it to better generalize to new data.

3.4 XGBoost

Finally, XGBoost (eXtreme Gradient Boosting) is a popular and powerful machine learning algorithm. It works by training multiple decision trees in a sequence - contrary to the random forest classifier which train them in parallel, with each tree attempting to correct the errors made by the previous tree. The final model is a weighted average of all the individual decision trees.

4 Analysis of the results and conclusion

Predicted			Predicted			Predicted			Predicted						
		0	1			0	1			0	1			0	1
Actual	0	1485	193	Actual	0	1484	194	Actual	0	1461	217	Actual	0	1484	255
	1	236	531		1	255	512		1	214	553		1	194	512
Logistic regression			Decision tree			Random forest			XGBoost						

Figure 4: Confusion matrices for the four models. 0 corresponds to no claim.

Model	Accuracy	Sensitivity	Specificity	PPV	NPV
Logistic regression	<u>0.8245</u>	0.8629	<u>0.7334</u>	<u>0.8850</u>	0.6923
Decision Tree Classifier	0.8164	0.8534	0.7252	0.8844	0.6675
Random Forest Classifier	<u>0.8245</u>	<u>0.8719</u>	0.7206	0.8725	0.7197
XGBoost	0.8164	0.8844	0.6675	0.8534	<u>0.7252</u>

Table 1: A sum-up table of the classification metrics for each model. PPV stands for Predicted Positive Values and NPV stands for Negative Predicted Values.

A Appendix

Variable	Type	Value ranges (if meaningful)
VEHICLE OWNERSHIP	Binary	0 or 1
MARRIED	Binary	0 or 1
CHILDREN	Binary	0 or 1
OUTCOME	Binary	0 or 1
AGE	Category	16-25, 26-39, 40-64, 65+
GENDER	Category	female, male
RACE	Category	majority, minority
DRIVING EXPERIENCE	Category	0-9y, 10-19y, 20-29y, 30y+
EDUCATION	Category	high school, none, university
INCOME	Category	middle class, poverty, upper class, working class
VEHICLE TYPE	Category	sedan, sports car
VEHICLE YEAR	Category	after 2015, before 2015
CREDIT SCORE	Float	From 0.0534 to 0.9608
ID	Integer	–
POSTAL CODE	Integer	–
ANNUAL MILEAGE	Integer	From 2000 to 22000
SPEEDING VIOLATIONS	Integer	From 0 to 22
DUIS	Integer	From 0 to 6
PAST ACCIDENTS	Integer	From 0 to 15

Table 2: A short description of the covariates, along with some insights about categorical variables.

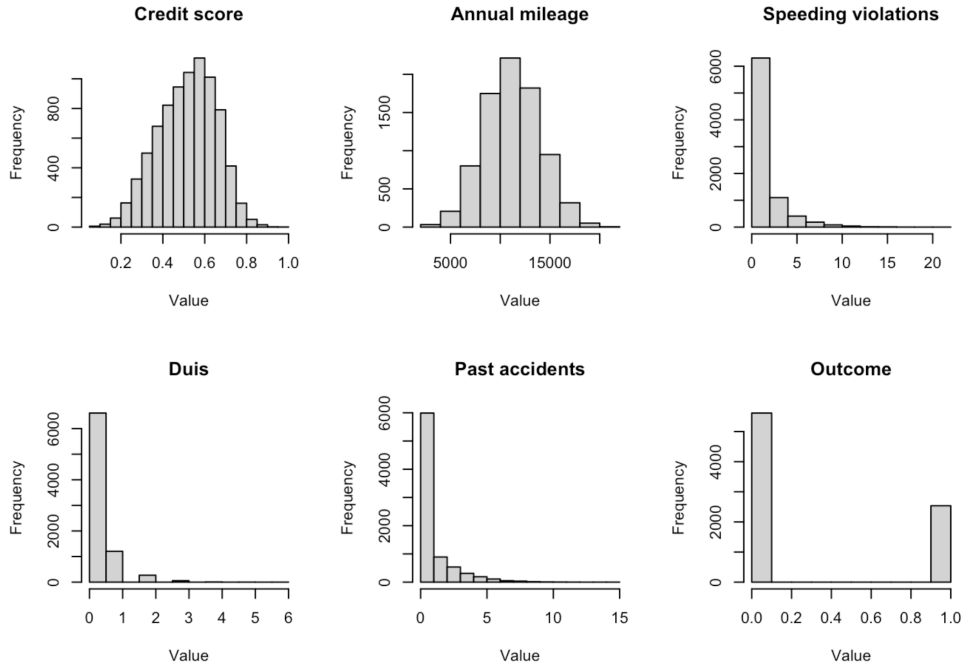


Figure 5: Histograms of the numerical variables.

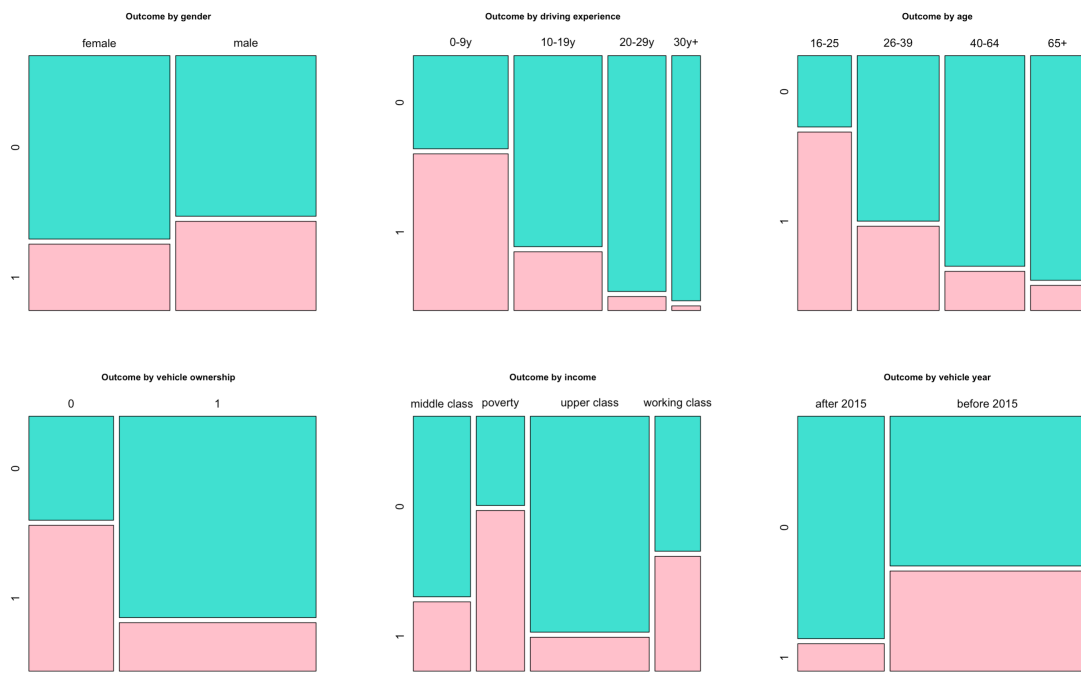


Figure 6: Value counts of categorical features with respect to the `OUTCOME` value.