Car Insurance Claims Classification Report

Hamed Vaheb
hamed.vaheb.001@student.uni.lu

January 14, 2023

Abstract

This report describes the project of analysis and prediction of the car insurance claims dataset. The dataset includes historical data of the policyholders (e.g., age, gender, vehicle details , etc.), among which the variable of interest is the outcome of insurance, indicating whether a customer's claim is approved or not. The `aiinsurance` R package [1] is developed to make this work reproducible, accessible, and equipped with advanced features, e.g., a pipeline that performs the main steps of this work in a single command, and an interactive app that displays the performance of the models. Using package's functions, two machine learning models, i.e., logistic regression and random forest are implemented to classify the outcomes based on solely the historical data. The results indicate promising prediction power with various evaluation metrics (e.g., accuracy, precision, recall, etc.) of over 80 percent. In addition to classification, informative insights from the dataset and models have been drawn, e.g., by dint of the random forest model, the variables of the dataset that perform more effective role in the prediction become evident.

1 Introduction

Rapid advances in artificial intelligence (AI) and machine learning are creating products and services with the potential not only to change the environment in which actuaries operate but also to provide new opportunities within actuarial science [2].

The use of statistical learning models has been a common practice in actuarial science since the 1980s. It was not long after since the field adopted classical models, such as linear models and generalized linear models (GLMs). While actuaries use GLMs frequently in practice, it was in the past few years that the use of AI and machine learning, and hence more modern models garnered significant attention in the field [3].

The goal of this work is to predict status of policyholders' claims. The status lies at the "outcome" column of the car insruance dataset, indicating whether a customer has claimed his loan or not.

A classical linear model and a modern nonlinear model is used and compared. The former model is logistic regression, which is an example of GLMs, accommodated to classification setting, i.e., for predicting discrete classes, in this work's case, status of claims. The latter model is a nonlinear tree-based model.

The remainder of this work is organized as the following: In section 2, main concepts included in the work are explained. In 2.1, the concept of car insurance is introduced followed by an elaboration on claims and how they interact among an insurer and a policyholder. Subsequently, the two machine learning models, logistic regression 2.2, and random forest 2.3 are briefly explained. As a contribution of this work, the `aiinsurance` package [1], introduced in 3.1, is developed, which automates many of the classification tasks, uses advanced features, e.g., runs a pipeline that performs the main steps of this work in a single command, and it provides an interactive shiny app to display the performance of the models.

2 Preliminary Concepts

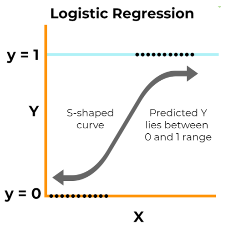
2.1 Car Insurance Claims

Car insurance is a type of insurance policy that financially protects drivers in the event of an accident or theft. There are several types of car insurance coverage, including liability insurance, collision insurance, comprehensive insurance, and personal injury protection.

An insurance claim is a formal request by a policyholder to an insurance company for coverage or compensation for a covered loss or policy event, in this work's case, a car accident. The insurance company either accepts or rejects the claim. If it is approved, the insurance company will issue payment to the insured or an approved interested party on behalf of the insured.

Predicting the outcome of claims can be utilized to better understand the customer strata and incorporate the findings throughout the insurance policy enrollment (including the underwriting and approval or rejection stages), triage claims and automate where possible, gradually obviating the need for human interaction, and optimize the entire insurance policy enrollment process flow [4].

2.2 Model: Logistic Regression

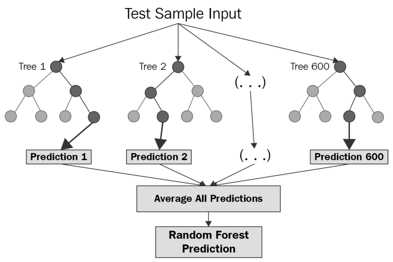


Logistic regression (logit), which is type of GLM, uses the logistic function to model the probability of the binary outcome by creating a model that takes in the input variable (e.g., client's information) and produces a probability that a "outcome" (the variable we aim to predict) is 1. The "outcome" in this work is whether a customer's claim is approved or not. The logistic function is defined as the following:

$$\frac{1}{1+e^{-x}}$$

This probability can then be used to make a prediction about the outcome. For instance, if the probability that the client's claim will be approved is greater than a certain threshold (e.g., 0.5), we predict that the claim will be approved. In another words, the goal of logit is to find the best set of coefficients for a set of independent variables that maximizes the likelihood (measuring how well parameters of a model fit the data) of the observed data.

2.3 Model: Random Forest



An ensemble learning model is a model that is constructed from multiple models, to obtain combined results, expected to be improved compared to any of the constituent models. The random forest is an ensemble model built from decision trees, i.e., flowchart-like tree structures, wherein each internal node represents a "test" on a variable (e.g., vehicle type), each branch represents the outcome of the test, and each leaf node represents a class label (e.g., 1 for claim approval and 0 for claim rejection). The intuition behind a decision tree is to recursively split the data into subsets based on the values of the input variables, such that the subsets are as "pure" as possible in terms of their class labels. The less pure a subset is, the more the data belongs to the same class, and the more pure it is, the more data is evenly split among all classes. The goal is to create a tree that can accurately classify new examples by traversing the tree from the root to a leaf node, making decisions at each internal node based on the values of the input variables.

As a single random tree might not be able to capture the proper inherent complexity of a data, and may either overfit (become too complex and remembers data rather than learning from it) or underfit (become too simple and hence unable to learn the inherent complicated patterns in the data). By training multiple decision trees and combining their predictions by taking a majority vote, a random forest is able to capture a more robust and accurate representation of the data. The randomness in the random forest comes from randomly selecting subsets of the data to train each decision tree, and randomly selecting subsets of input variables to consider at each split point in the decision tree. This helps to decorrelate the trees, i.e., reducing the correlation or dependence between the trees, consequently, make the model more robust to overfitting.

3 Implementation

3.1 `aiinsurance` Package

AI INSURANCE

Icon Source

The `aiinsurance` R package [1] is developed to make this work reproducible, accessible, and equipped with advanced features.

Instructions on how to install and use the package's functions and features is provided in the README part of package's Github repository.

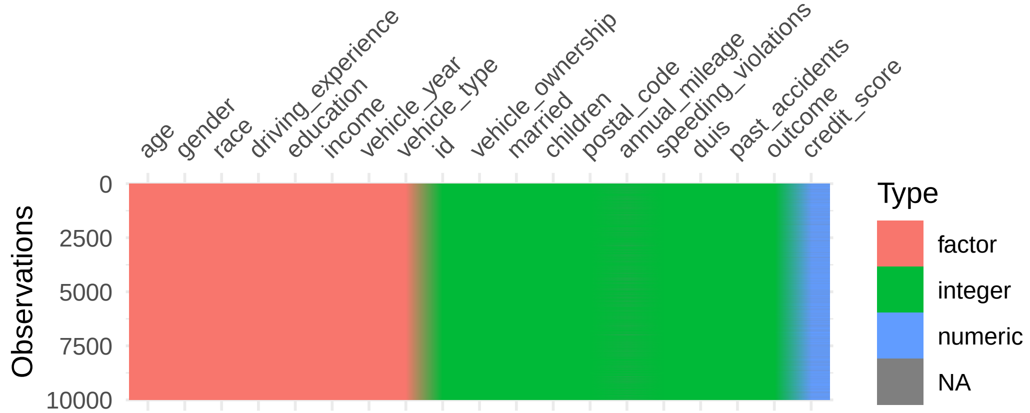
Noteworthy features of the package are explained in the following: A pipeline that performs the main steps of this work in a single command, and an interactive app that displays the performance of the models. Using package's functions

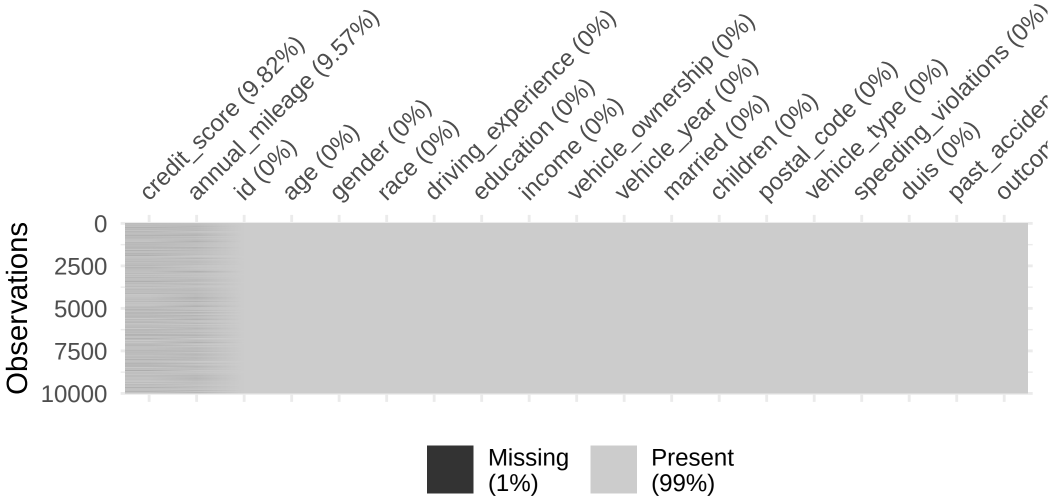
3.2 Exploratory Data Analysis (EDA)

```
## Rows: 10,000
## Columns: 19
## $ id                  <int> 569520, 750365, 199901, 478866, 731664, 877557, 93~
## $ age                 <chr> "65+", "16-25", "16-25", "16-25", "26-39", "40-64"~
## $ gender              <chr> "female", "male", "female", "male", "male", "femal~
## $ race                <chr> "majority", "majority", "majority", "majority", "m~
## $ driving_experience  <chr> "0-9y", "0-9y", "0-9y", "0-9y", "10-19y", "20-29y"~
## $ education           <chr> "high school", "none", "high school", "university"~
## $ income              <chr> "upper class", "poverty", "working class", "workin~
## $ credit_score        <dbl> 0.6290273, 0.3577571, 0.4931458, 0.2060129, 0.3883~
## $ vehicle_ownership   <int> 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1,~
## $ vehicle_year        <chr> "after 2015", "before 2015", "before 2015", "befor~
## $ married             <int> 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1,~
## $ children            <int> 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1,~
## $ postal_code         <int> 10238, 10238, 10238, 32765, 32765, 10238, 10238, 1~
## $ annual_mileage      <int> 12000, 16000, 11000, 11000, 12000, 13000, 13000, 1~
## $ vehicle_type        <chr> "sedan", "sedan", "sedan", "sedan", "sedan", "seda~
## $ speeding_violations <int> 0, 0, 0, 0, 2, 3, 7, 0, 0, 0, 6, 4, 4, 0, 0, 0, 10~
## $ duis                <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 1, 0, 2, 0, 2,~
## $ past_accidents      <int> 0, 0, 0, 0, 1, 3, 3, 0, 0, 0, 7, 0, 2, 0, 1, 0, 1,~
## $ outcome             <int> 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0,~
```

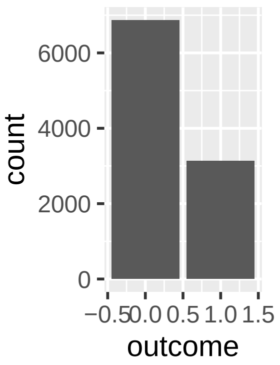
In the following, the nature of categorical columns are reported:

```
## [1] "List of categorical columns containing characters: "
## [1] "age"                "gender"             "race"
## [4] "driving_experience" "education"          "income"
## [7] "vehicle_year"       "vehicle_type"
## [1] "List of categorical columns containing numbers: "
## [1] "vehicle_ownership" "married"           "children"
## [4] "postal_code"       "outcome"
```





The proportion of outcome classes are shown in the following:



3.3 Preprocessing

3.3.1 Encoding Categorical Columns

3.4 Classification Models

4 Results and Discussion

5 Conclusion

Table 1: Sample table title

Part		
Name	Description	Size (μm)
Dendrite	Input terminal	~100
Axon	Output terminal	~10
Soma	Cell body	up to 10^6

6 Headings: first level

LaTeX command can be used to reference other section. See Section 6. However, you can also use **bookdown** extensions mechanism for this.

6.1 Evaluation

Below we can see how to use tables.

See awesome Table 1 which is written directly in LaTeX in source Rmd file.

You can also use R code for that.

References

[1] Hamed Vaheb. AI Insurance Package. Available at http://github.com/berserkhmdvhb/aiinsurance.

[2] Ronald Richman. Ai in actuarial science – a review of recent advances – part 1. *Annals of Actuarial Science*, 15(2):207–229, 2021.

[3] Christopher Blier-Wong, Hélène Cossette, Luc Lamontagne, and Etienne Marceau. Machine learning in p&c insurance: A review for pricing and reserving. *Risks*, 9(1), 2021.

[4] Mohamed Hanafy and Ruixing Ming. Classification of the insureds using integrated machine learning algorithms: A comparative study. *Applied artificial intelligence*, 36(1), 2022.