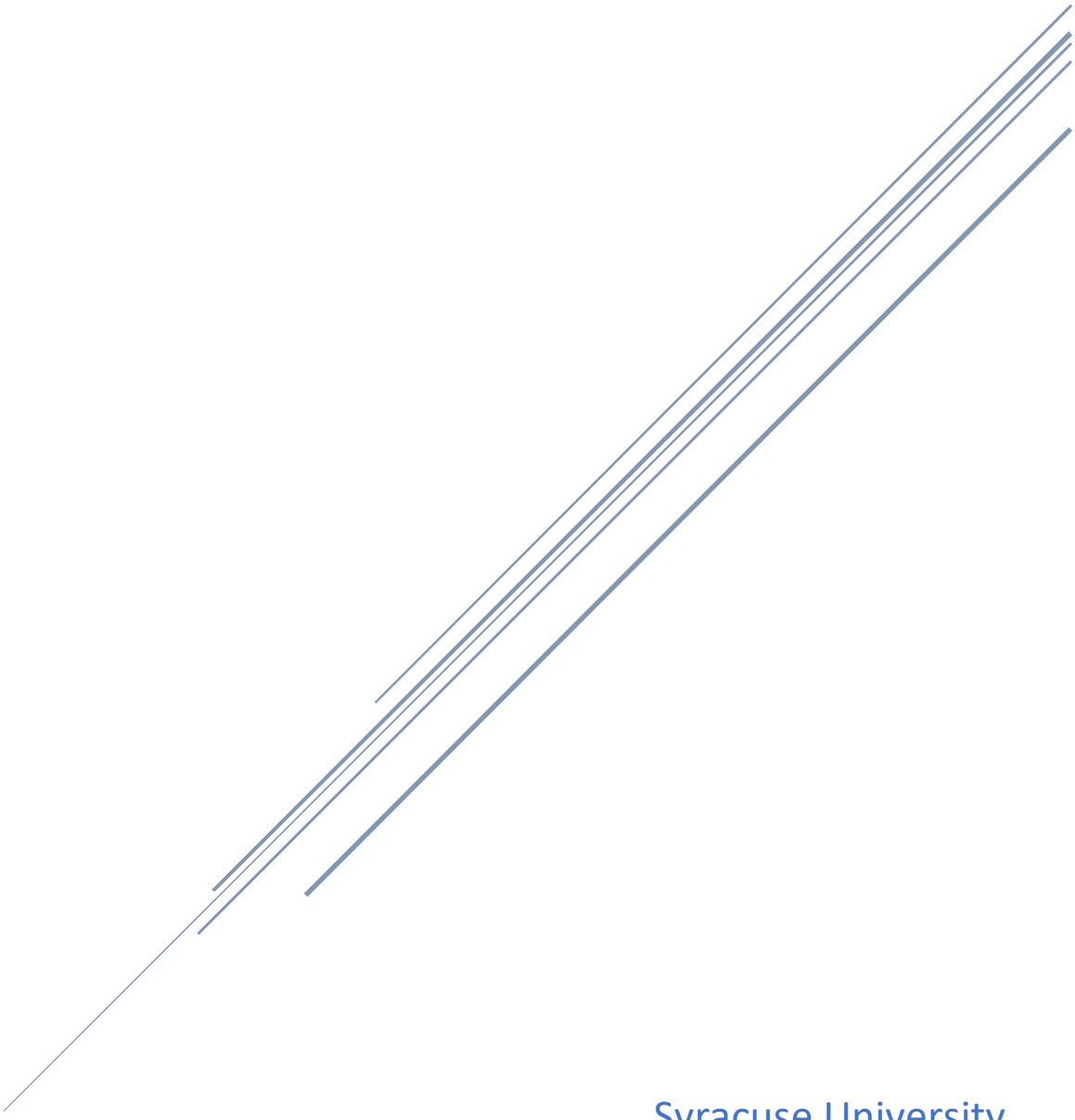
# INSURANCE CLAIMS MODELNG

Tia Jones & Elizabeth Jones

10 May 2022



Syracuse University

IST 707: Applied Machine Learning

## Introduction

Insurance can be a risky business. Insurance companies must charge their customers before knowing what type of customer they are covering, so understanding the risk characteristics of their customers is paramount, especially if an insurance company intends to make a profit. This is where modeling comes in handy. When used properly, modeling can help to assess customers and set the most appropriate premiums. In this report we will utilize Naïve Bayes, Random Forest and Decision Tree modeling with car insurance customer data to rank new customers. The insurance company will use this information to rank customers and use the ranking to decide what rate to charge customers.

### Data

The data set we will be using was found on Kaggle and contains customer demographic information as well as vehicle type and driving behavior information. The data set contains 10,000 rows and 19 columns and was collected over the course of one year. Our target variable is whether or not someone filed a claim which is identified in the "OUTCOME" column by a factor of 0 or 1. The rest of the variables consists of driver information:

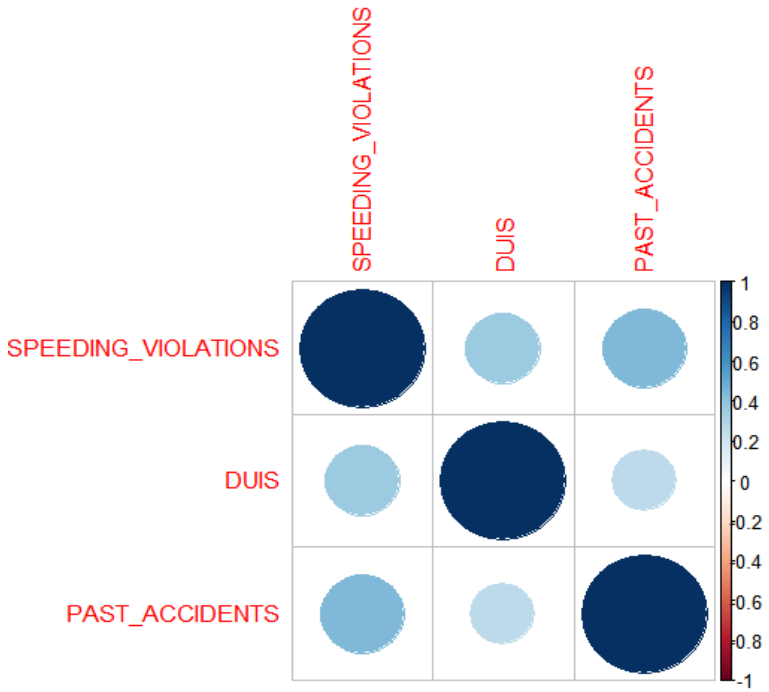
| | |
|---|---|
| ID | Unique identifier; integer |
| AGE | Segmented |
| GENDER | male or female |
| RACE | majority or minority |
| DRIVING_EXPERIENCE | 0-9 years, 10-19 years, 20-29 years, and 30+ |
| EDUCATION | high school, university, or none |
| INCOME | middle class, upper class, poverty, or working class |
| CREDIT_SCORE | integer |
| VEHICLE_OWNER | 0 or 1 (whether they own the vehicle) |
| VEHICLE_YEAR | after 2015, before 2015 |
| MARRIED | 0 or 1 |
| CHILDREN | 0 or 1 |
| POSTAL_CODE | integer |
| ANNUAL_MILEAGE | integer |
| VEHICLE_TYPE | sedan or sports car |
| SPEEDING_VIOLATION | integer |
| DUIS | integer |

| | |
|---|---|
| PAST_ACCIDENTS | integer |
| OUTCOME | 0 or 1 (whether they filed a claim) |

**Data Exploration**

We preformed data exploration to get a better idea of our data set. We separated our data set by customers who submitted claims and those that did not. 5,613 did not submit a claim, while 2,536 did. Generally, those that did not submit claims consisted of upper-class middle-aged people who had between 10-19 years of driving experience. Customers that did submit claims were typically very young drivers between the ages of 16-25 with 0-9 years of driving experience who fell within the "poverty" income category. With further exploration we also found that there appeared to be no correlation between past accidents, speeding violations, and DUIs.

| | SPEEDING_VIOLATIONS | DUIS | PAST_ACCIDENTS |
|---|---|---|---|
| SPEEDING_VIOLATIONS | 1.0000000 | 0.3656997 | 0.4455310 |
| DUIS | 0.3656997 | 1.0000000 | 0.2648336 |
| PAST_ACCIDENTS | 0.4455310 | 0.2648336 | 1.0000000 |



**Data Preprocessing**

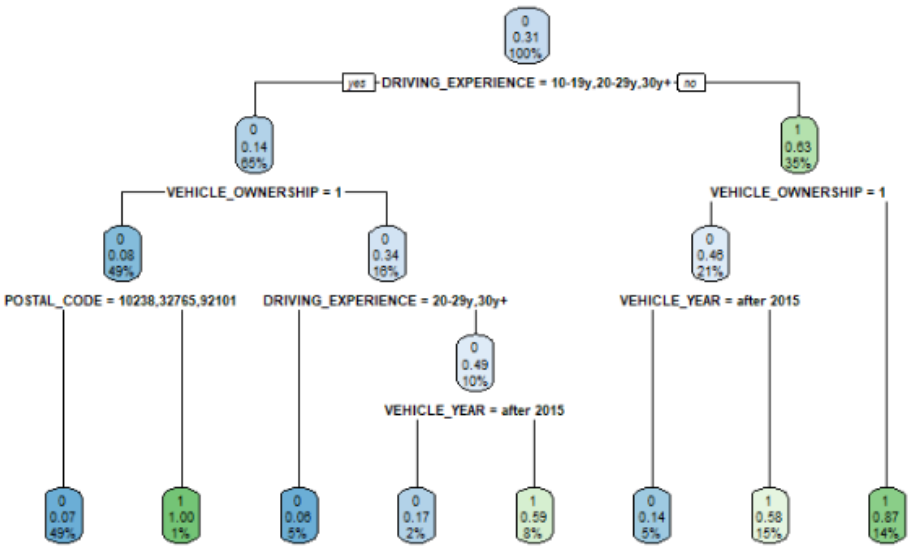
To prepare the data for our model we checked whether there were any NA's or missing data points in our data after importing it into R. We found that were 1,939 instances of missing data and removed those rows. This left us with 8,140 rows of data. We then removed any columns that would cause bias in our model which were the age, race and also removed the ID columns as it was unnecessary data that would not improve our model outcome. In some cases postal codes could also create bias but we decided to keep the column after checking that there was no correlation with race. Next, we added a new column to the data set which gave a rank of each customer depending on the number of past accidents. Customers with 0 past accidents were ranked good, customers which 1 past accident were ranked ok and customer with more than 1 past accident were ranked bad. After preprocessing our data set included 17 columns with 8,140 rows.

**Decision Tree Model**

We used the CART package to create our Decision Tree. Overall accuracy received from the model was 83% percent with an error rate of 17%. This accuracy was significant as it was better than the no information rate which was 68.8%. The model was better at identifying those who did not file claims versus those who did.

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 2801  255
         1  566 1266

               Accuracy : 0.832
                 95% CI : (0.8213, 0.8424)
    No Information Rate : 0.6888
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.629

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.8319
            Specificity : 0.8323
         Pos Pred Value : 0.9166
         Neg Pred Value : 0.6910
             Prevalence : 0.6888
         Detection Rate : 0.5730
   Detection Prevalence : 0.6252
      Balanced Accuracy : 0.8321

       'Positive' Class : 0
```

The model correctly identified 2,801/3,367 and incorrectly identified 566/3,367 for those who did not file claims. The model identified people who filed a claim correctly with a ratio of 1,260/1,515 and incorrectly with a ratio of 255/1,515 for those who did file a claim. The features identified that held the most importance in the creation of the tree were Driving Experience, Age, Vehicle Ownership.




To improve the model's accuracy for identifying people who file claims we added a loss matrix to the model that penalized misclassifying those who did file a claim. After adding the loss matrix, the model's overall accuracy decreased to 76% but specificity increased to 93% percent.

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 2297  102
         1 1070 1419

               Accuracy : 0.7602
                 95% CI : (0.748, 0.7721)
    No Information Rate : 0.6888
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5238

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.6822
            Specificity : 0.9329
         Pos Pred Value : 0.9575
         Neg Pred Value : 0.5701
             Prevalence : 0.6888
         Detection Rate : 0.4699
   Detection Prevalence : 0.4908
      Balanced Accuracy : 0.8076

       'Positive' Class : 0
```

Although the specificity increased significantly the sensitivity of the model suffered as the accuracy decreased to 68%. As our training dataset was skewed towards the positive class

(those who did not file a claim) by a ratio of 2,246 for those who did not file to 1,015 for those who did; we felt using the SMOTE function could improve the model's accuracy even further. Balancing the training set improved our model exactly how we hoped. The ratio of the dataset was transformed into buckets of 2,030 for both classes. Specificity stayed above 90% while sensitivity increased to 72%.

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 2442  128
         1  925 1393

               Accuracy : 0.7846
                 95% CI : (0.7728, 0.796)
    No Information Rate : 0.6888
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5606

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.7253
            Specificity : 0.9158
         Pos Pred Value : 0.9502
         Neg Pred Value : 0.6009
             Prevalence : 0.6888
         Detection Rate : 0.4996
   Detection Prevalence : 0.5258
      Balanced Accuracy : 0.8206

       'Positive' Class : 0
```

**Random Forest**

Next, we aimed to compare the accuracies received from the Decision Tree model to Random Forests. As the Random Forest technique combines multiple decision trees to output the best accuracy, we expected to improve our model even further. We used the randomForest package to create this model and chose the best mtry through tuning. To our surprise, the model performed worse than the Decision Tree model did before any tuning took place. Against the first model overall accuracy, specificity and sensitivity had decreased.

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 2584  267
         1  783 1254

               Accuracy : 0.7852
                 95% CI : (0.7734, 0.7966)
    No Information Rate : 0.6888
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5415

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.7674
            Specificity : 0.8245
         Pos Pred Value : 0.9063
         Neg Pred Value : 0.6156
             Prevalence : 0.6888
         Detection Rate : 0.5286
   Detection Prevalence : 0.5833
      Balanced Accuracy : 0.7960

       'Positive' Class : 0
```

**Naïve Bayes Model**

Last, we created a Naïve Bayes model. Our first model had an accuracy of 76%, however, after tuning we were able to improve the accuracy to 78% with an error rate of 27%. The accuracy was also better than the no information rate and was better at identifying those who did not file claims versus those who did. We ran into some issues while tuning this model. We received multiple error messages which caused a good amount of data to be removed from the model. So, even though the tuning helped to improve the accuracy, its possible that it may not be entirely reliable.

In the end, the model correctly identified 887/1,113 and incorrectly identified 226/1,113 for those who did not file claims. The model identified people who filed a claim correctly with a ratio of 386/516 and incorrectly with a ratio of 130/516 for those who did file a claim. The features identified that held the most importance in the creation of the tree were Driving Experience, Age, Vehicle Ownership.

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 887 130
         1 226 386

               Accuracy : 0.7815
                 95% CI : (0.7606, 0.8013)
    No Information Rate : 0.6832
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5191

 Mcnemar's Test P-Value : 4.779e-07

            Sensitivity : 0.7969
            Specificity : 0.7481
         Pos Pred Value : 0.8722
         Neg Pred Value : 0.6307
             Prevalence : 0.6832
         Detection Rate : 0.5445
   Detection Prevalence : 0.6243
      Balanced Accuracy : 0.7725

       'Positive' Class : 0
```

**Conclusion**

The goal of this project was to create a model which could use the outcome variable to decide what rate to give customers. Out of the three models, the Decision Tree performed the best, however the Random Forest model may have given better results with additional tuning. Using the Decision Tree predictions from the third model we created two new columns, one titled "prediction" and the other "rate". The prediction column gives the model prediction to whether that person filed a claim while the rate column identifies what rate that customer should receive. Definition for that column is stated below.

- Good + Outcome 0/1 = Can result in a Best or Good final rating
- OK + Outcome 0/1 = Can result in a Good or Ok final rating
- Bad + Outcome 0/1 = Can result in a Bad or Ok final rating