Prediction of Fraudulent Insurance claims

By: Katya Mora

Table of Contents

Background Information

- In 2018, insurance sectors worldwide amassed a revenue exceeding $5 trillion.
- Types of insurance fraud
  - staged accidents
  - fake claims
  - exaggerated claims.
  - Each type of fraud requires a different approach to detection, and machine learning can be used to develop targeted models for each type.

Why Predict

1. Cost-saving:
    1. Save insurance companies a significant amount of money.
    2. The ability for insurance companies to conduct more accurate policy pricing
2. Improved customer service:
    1. Fraudulent claims take time and resources to investigate, and they can delay the processing of legitimate claims.
3. Risk management:
    1. Insurance companies use fraud prediction models to identify high-risk areas and customers. This allows them to implement preventative measures to reduce the risk of fraud, such as increasing premiums or requiring additional verification steps.

Development

- Machine learning is a useful tool for detecting insurance fraud, since it can analyze vast amounts of data and identify patterns that may be identify fraudulent activity.
- Machine learning techniques can used on historical data to recognize patterns of fraudulent behavior, which can also be updated in real-time to adapt to changing fraud patterns.

Data

- https://www.kaggle.com/datasets/buntyshah/auto-insurance-claims-data
- 39 Variables
  - Focus on the following 5:
    - Age
    - Sex
    - Marital Status
    - Police Report Filed
    - Total Claim amount

months_as_customer	Age	policy_number	policy_bind_date	policy_state	policy_csl	policy_deductable	policy_annual_premium	umbrella_limit	insured_zip	Sex	insured_education_level	insured_occupation	insured_hobbies	MaritalStatus	capital-gains	capital-loss	incident_date
328	48	521585	10/17/2014	OH	250/500	1000	1406.91	0	466132	Male	MD	craft-repair	sleeping	Married	53300	0	1/25/2015
228	42	342868	6/27/2006	IN	250/500	2000	1197.22	5000000	468176	Male	MD	machine-op-inspct	reading	Single	0	0	1/21/2015
134	29	687698	9/6/2000	OH	100/300	2000	1413.14	5000000	430632	Female	PhD	sales	board-games	Married	35100	0	2/22/2015
256	41	227811	5/25/1990	IL	250/500	2000	1415.74	6000000	608117	Female	PhD	armed-forces	board-games	Single	48900	-62400	1/10/2015
228	44	367455	6/6/2014	IL	500/1000	1000	1583.91	6000000	610706	Male	Associate	sales	board-games	Single	66000	-46000	2/17/2015
256	39	104594	10/12/2006	OH	250/500	1000	1351.1	0	478456	Female	PhD	tech-support	bungie-jumping	Single	0	0	1/2/2015

Type of Methods

- Naive Bayes
  - Binary
- Artificial Neural Network
  - Binary

SAS Naive Bayes Binary Classification

```
proc import out=insurance
datafile="\\vdi-fileshare01\UEMprofiles\026374944\Desktop\hhh\insurance_claims.csv"
dbms=csv replace;
run;

/*SPLITTING DATA INTO 80% TRAINING AND 20% TESTING SETS*/
proc surveyselect data=insurance rate=0.8 seed=177937
out=insurance outall method=srs;
run;

data train (drop=selected);
set insurance;
if selected=1;
run;

data test (drop=selected);
set insurance;
if selected=0;
run;

/*COMPUTING PRIOR PROBABILITIES*/
proc freq data=train noprint;
 table fraud_reported/out=priors;
run;

data priors;
set priors;
 percent=percent/100;
 if fraud_reported='N' then call symput('prior_N', percent);
 if fraud_reported='Y' then call symput('prior_Y', percent);
run;


/*COMPUTING POSTERIOR PROBABILITIES FOR CATEGORICAL PREDICTORS*/
proc freq data=train noprint;
 table fraud_reported*Sex/out=gender_perc nocum list;
run;

data gender_perc;
set gender_perc;
 percent=percent/100;
if fraud_reported='N' and Sex='Female' then call symput('Female_No', percent);
if fraud_reported='N' and Sex='Male' then call symput('Male_No', percent);
if fraud_reported='Y' and Sex='Female' then call symput('Female_Yes', percent);
if fraud_reported='Y' and Sex='Male' then call symput('Male_Yes', percent);
 run;
```

```
proc freq data=train noprint;
 table fraud_reported*MaritalStatus/out=MaritalStatus_perc
   nocum list;
run;


data MaritalStatus_perc;
set MaritalStatus_perc;
 percent=percent/100;
if fraud_reported='N' and MaritalStatus='Married' then call symput('Married_No', percent);
if fraud_reported='N' and MaritalStatus='Single' then call symput('Single_No', percent);
if fraud_reported='Y' and MaritalStatus='Married' then call symput('Married_Yes', percent);
if fraud_reported='Y' and MaritalStatus='Single' then call symput('Single_Yes', percent);
 run;


proc freq data=train noprint;
 table fraud_reported*PoliceReportFiled/out=PoliceReportFiled_perc
   nocum list;
run;

data PoliceReportFiled_perc;
set PoliceReportFiled_perc;
 percent=percent/100;
if fraud_reported='N' and PoliceReportFiled='No' then call symput('PoliceReportFiled_No_No', percent);
if fraud_reported='N' and PoliceReportFiled='Yes' then call symput('PoliceReportFiled_Yes_No', percent);
if fraud_reported='Y' and PoliceReportFiled='No' then call symput('PoliceReportFiled_No_Yes', percent);
if fraud_reported='Y' and PoliceReportFiled='Yes' then call symput('PoliceReportFiled_Yes_Yes', percent);
 run;



/*COMPUTING MEAN AND STANDARD DEVIATION FOR NUMERICAL PREDICTORS*/
proc means data=train mean std noprint;
 class fraud_reported;
  var Age total_claim_amount;
output out=stats;
run;


data stats;
 set stats;
if fraud_reported='N' and _stat_='MEAN' then
  do;
   call symput('Age_mean_no',Age);
   call symput('total_claim_amount_mean_no',total_claim_amount);
  end;
```

SAS Naive Bayes Binary Classification

```
/*COMPUTING POSTERIOR PROBABILITIES FOR TESTING DATA*/
data test;
set test;
if (Sex='Female' and PoliceReportFiled='No') then
do;
pred_prob_N=&prior_N*&Female_No*&PoliceReportFiled_No_No*Married_No*1/(2*3.14)*1/(&Age_std_no*&total_claim_amount_std_no)
*exp(-(Age-&Age_mean_no)**2/(2*&Age_std_no**2)-(total_claim_amount-&total_claim_amount_mean_no)**2/(2*&total_claim_amount_std_no**2));

pred_prob_Y=&prior_Y*&Female_Yes*&PoliceReportFiled_No_Yes*Married_Yes*1/(2*3.14)*1/(&Age_std_yes*&total_claim_amount_std_yes)
*exp(-(Age-&Age_mean_yes)**2/(2*&Age_std_yes**2)-(total_claim_amount-&total_claim_amount_mean_yes)**2/(2*&total_claim_amount_std_yes**2));
end;

if (Sex='Male' and PoliceReportFiled='Yes') then
do;
pred_prob_N=&prior_N*&Male_No*&PoliceReportFiled_Yes_No*Married_No*1/(2*3.14)*1/(&Age_std_no*&total_claim_amount_std_no)
*exp(-(Age-&Age_mean_no)**2/(2*&Age_std_no**2)-(total_claim_amount-&total_claim_amount_mean_no)**2/(2*&total_claim_amount_std_no**2));

pred_prob_Y=&prior_Y*&Male_Yes*&PoliceReportFiled_Yes_Yes*Married_Yes*1/(2*3.14)*1/(&Age_std_yes*&total_claim_amount_std_yes)
*exp(-(Age-&Age_mean_yes)**2/(2*&Age_std_yes**2)-(total_claim_amount-&total_claim_amount_mean_yes)**2/(2*&total_claim_amount_std_yes**2));
end;

if (Sex='Female' and PoliceReportFiled='Yes') then
do;
pred_prob_N=&prior_N*&Female_No*&PoliceReportFiled_Yes_No*Married_No*1/(2*3.14)*1/(&Age_std_no*&total_claim_amount_std_no)
*exp(-(Age-&Age_mean_no)**2/(2*&Age_std_no**2)-(total_claim_amount-&total_claim_amount_mean_no)**2/(2*&total_claim_amount_std_no**2));

pred_prob_Y=&prior_Y*&Female_Yes*&PoliceReportFiled_Yes_Yes*Married_Yes*1/(2*3.14)*1/(&Age_std_yes*&total_claim_amount_std_yes)
*exp(-(Age-&Age_mean_yes)**2/(2*&Age_std_yes**2)-(total_claim_amount-&total_claim_amount_mean_yes)**2/(2*&total_claim_amount_std_yes**2));
end;

if (Sex='Male' and PoliceReportFiled='No') then
do;

pred_prob_N=&prior_N*&Male_No*&PoliceReportFiled_No_Yes*Married_No*1/(2*3.14)*1/(&Age_std_no*&total_claim_amount_std_no)
*exp(-(Age-&Age_mean_no)**2/(2*&Age_std_no**2)-(total_claim_amount-&total_claim_amount_mean_no)**2/(2*&total_claim_amount_std_no**2));

pred_prob_Y=&prior_Y*&Male_Yes*&PoliceReportFiled_No_No*Married_Yes*1/(2*3.14)*1/(&Age_std_yes*&total_claim_amount_std_yes)
*exp(-(Age-&Age_mean_yes)**2/(2*&Age_std_yes**2)-(total_claim_amount-&total_claim_amount_mean_yes)**2/(2*&total_claim_amount_std_yes**2));
end;
```

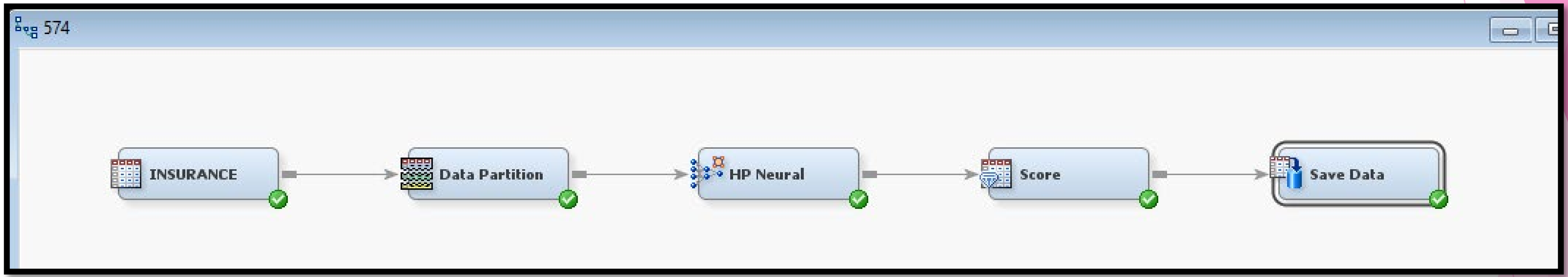
SAS Naive Bayes Binary Classification

```
/*COMPUTING PREDICTION ACCURACY*/

data test;
 set test;
  if pred_prob_N < pred_prob_Y then pred_class='Y';
  else pred_class='N';
  if fraud_reported=pred_class then pred=1; else pred=0;
 run;

 proc sql;
  select mean(pred) as accuracy
   from test;
quit;
proc print;
run;
```

The SAS System

accuracy
0.725

SAS Code:
ANN – Binary



SAS Code:
ANN – Binary

Prediction Accuracy

```
proc import out=sasuser.insurance
datafile="//vdi-fileshare01/UEMprofiles/026374944/Desktop/New folder/insurance_claims.csv"
dbms=csv replace;
run;


/*COMPUTING PREDICTION ACCURACY*/

data accuracy;
set tmpl.em_save_test;
match=(em_classification=em_classtarget);
run;

proc sql;
select mean(match) as accuracy
from accuracy;
quit;
```

The SAS System

accuracy
0.753333

R Code Naive Bayes Binary Classification

Naives Bayes binary classification

```
insurance_claims$fraud_reported<-ifelse(insurance_claims$fraud_reported=='Y',1,0)
insurance_claims$Sex<- ifelse(insurance_claims$Sex=='Female',1,0)
insurance_claims$MaritalStatus<-ifelse(insurance_claims$MaritalStatus=='Single',1,0)
insurance_claims$PoliceReportFiled<-ifelse(insurance_claims$PoliceReportFiled=='Yes',1,0)

insurance_claims2<-insurance_claims[,c("Age","Sex","MaritalStatus","PoliceReportFiled",
                    "total_claim_amount","fraud_reported")]
```

```
#splitting 80and 20%

sample <- sample(c(TRUE, FALSE), nrow(insurance_claims2), replace=TRUE, prob=c(0.8,0.2))
train<- insurance_claims2[sample,]
test<- insurance_claims2[!sample,]

test.complete <- test[complete.cases(test), ]

test.x <- as.matrix(test.complete[, c("Age", "Sex", "MaritalStatus", "PoliceReportFiled", "total_claim_amount")])
test.y <- as.matrix(test.complete[, "fraud_reported"])
#install.packages("caret")
# this gives coloumn 2 and 6 are near 0 variance- remove these variables had to correct
library(caret)
```

R Code Naive Bayes Binary Classification

```r
library(e1071)
nb.class<- naiveBayes(fraud_reported ~ Age + Sex + MaritalStatus + PoliceReportFiled + total_claim_amount, data=t
rain)
```

```r
# Make predictions and calculate accuracy
pred.y <- as.numeric(predict(nb.class, test.x)) - 1
match <- ifelse(test.y == pred.y, 1, 0)
accuracy <- mean(match) * 100

# Print accuracy
print(paste("The accuracy of the Naive Bayes classifier is", round(accuracy, digits=2), "%"))
```

```
## [1] "The accuracy of the Naive Bayes classifier is 75.79 %"
```

R Code:
Artificial Neural Network

```
scale01 <- function(x){
  (x-min(x))/(max(x)-min(x))
}

insurance_claims2<- insurance_claims2 %>% mutate_all(scale01)
```

```
set.seed(177937)
sample <- sample(c(TRUE, FALSE), nrow(insurance_claims2), replace=TRUE, prob=c(0.8,0.2))
train<- insurance_claims2[sample,]
test<- insurance_claims2[!sample,]

train.x<- data.matrix(train[-6])
train.y<- data.matrix(train[6])
test.x<- data.matrix(test[-6])
test.y<- data.matrix(test[6])

library(neuralnet)
```

R Code: Artificial Neural Network

fitting ANN with logistic activation fcn

```
ann.class<- neuralnet(fraud_reported ~ Age + Sex + MaritalStatus + PoliceReportFiled + total_claim_amount,
data=train, hidden=3, act.fct="logistic")
plot(ann.class)
```

compute prediction for testing data

```
pred.prob<- predict(ann.class, test.x)[,1]

pred.y <- rep(0, length(test.y))

match<- c()
for (i in 1:length(test.y)){
  pred.y[i]<- ifelse(pred.prob[i]>0.5,1,0)
  match[i]<- ifelse(test.y[i]==pred.y[i],1,0)
}

print(paste("ANN accuracy is =", round(mean(match), digits=4)))
```

```
## [1] "ANN accuracy is = 0.726"
```

fitting ANN with logistic activation fcn

```
ann.class<- neuralnet(fraud_reported ~ Age + Sex + MaritalStatus + PoliceReportFiled + total_claim_amount,
data=train, hidden=c(2,3), act.fct="logistic")

plot(ann.class)
```

R Code: Artificial Neural Network

prediction accuracy for test data

```
pred.prob<- predict(ann.class, test.x)[,1]

match<- c()
pred.y<- c()
for (i in 1:length(test.y)){
  pred.y[i]<- ifelse(pred.prob[i]>0.5,1,0)
  match[i]<- ifelse(test.y[i]==pred.y[i],1,0)
}

print(paste("ANN with logistic activation fcn accuracy is =", round(mean(match), digits=4)))
```

```
## [1] "ANN with logistic activation fcn accuracy is = 0.7163"
```

fitting ANN with TANH activation function

```
ann.class<- neuralnet(fraud_reported ~ Age + Sex + MaritalStatus + PoliceReportFiled + total_claim_amount,
data=train, hidden=2, act.fct="tanh")
plot(ann.class)
```

prediction accuracy for test data

```
pred.prob<- predict(ann.class, test.x)[,1]

match<- c()
pred.y<- c()
for (i in 1:length(test.y)){
  pred.y[i]<- ifelse(pred.prob[i]>0.5,1,0)
  match[i]<- ifelse(test.y[i]==pred.y[i],1,0)
}

print(paste("ANN with TANH activation function accuracy=", round(mean(match), digits=4)))
```

```
## [1] "ANN with TANH activation function accuracy= 0.726"
```

- ANN with TANH accuracy = 72.6%

Summary of Results compared

Naive Bayes-Binary

- R Code:
  - 75.79%
- SAS Code
  - 72.6%

ANN- Binary

- R Code:
  - ANN with TANH accuracy = 72.6%
- SAS Code
  - 75.33%

Future Work

- I will use this set in a combination with a different set using Logistic regression.
- This is a very simple idea of predicting as there are can involve complex schemes and multiple parties.
- Some common machine learning techniques used in insurance fraud detection include anomaly detection, clustering, and classification.(Need better datasets).

Thank You!