Assignment2-CSE780

Paria, Fakhrzad

Stuent ID: 400353290

7-October-2021

a.Describe the data

we will use a Car insurance claim dataset in 2021 that has collected based on customers of that insurance company dat (2021). there are 10000 customer as sample in this data frame. Also 17 features have been collected based on the customer's attributes. There is one column that shows this customer had accident claim last year or not. this is logical column. The data shows that 23percent of customers claimed the accident insurance last year.

b.Exploratory data analysis

b.1 Summary and Data Cleanup

Following table1 is summary of customer attributes. The rows with NA values in SpeedingViolation have been removed. Also the age and experience were interval that both have converted to Number(by mean of interval). Other char columns(gender, married,education,...), now are factors with this logic: Gender female=1 , male=0 - married True=1 , false=0 - children True=1 , false=0

Table 1: Customer dataset attributes

	age	**gender**	driving_experience	married	children	postal_code
type	num	Factor	num	Factor	Factor	num
X	Min. :20	0:9043	Min. : 5	0:9043	0:9043	Min.:10238
X.1	Median :33		Median :15			Median :10238
X.2	Mean :42.38		Mean :15.76			Mean :19788
X.3	Max. :65		Max. :35			Max. :92101

	race	education	income	credit_score	vehicle_year
type	Factor	Factor	Factor	num	Factor
X	Length:9043	0:9043	0:9043	Min. :0.0534	0:9043
X.1			upper class :3946	Median :0.5268	
X.2			working class:1546	Mean :0.5164	
X.3				Max. :0.9608	

	vehicle_ownership	vehicle_type	annual_mileage	speeding_violations	past_accidents
type	Factor	Factor	num	num	num
X	0:9043	0:9043	Min. : 2000	Min. : 0	Min.: 0
X.2			Median :12000	Median : 0	Median : 0
X.3			Mean :11697	Mean : 1.491	Mean : 1.066
X.5			Max. :22000	Max. :22.000	Max. :15
X.6			NA's :957		

b.2 pairwise correlation

In this section we tried to explore the association by visualization tools and some statistical evidence. In first step we used xtabs() to verify that all levels of factor variables had both claimed insurance. Then we use ggpairs() John W Emerson (2012) to see correlation between numeric features in figure1. the significant point is that Age and driving experience are collinear. In figure4 It appears that older customers who own the vehicle tended to not claim the insurance for accident(outcome in figure legend) than those who did not. Also in figure5 it seems there are relationship between credit score and number of claims in last year.

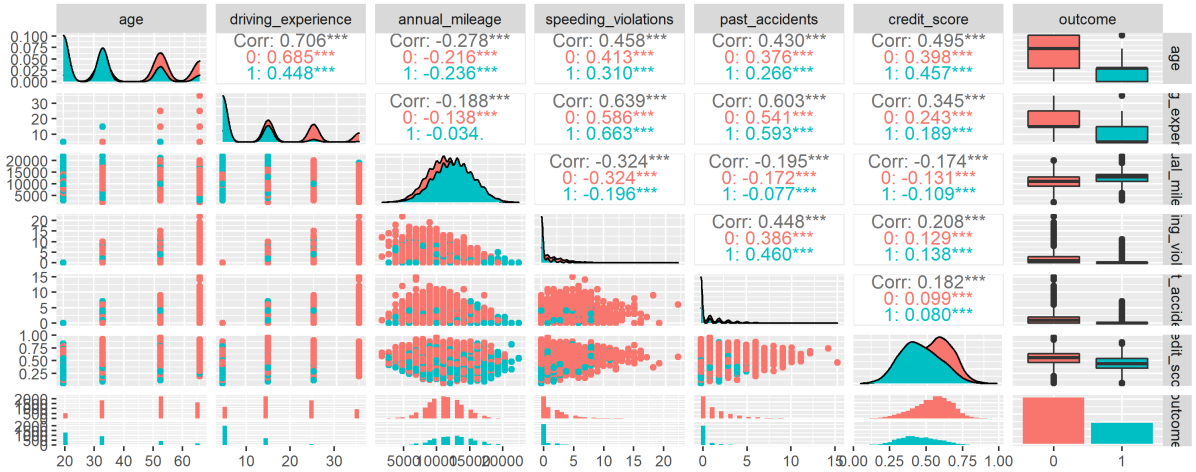

Figure 1: Pairwise Correlation

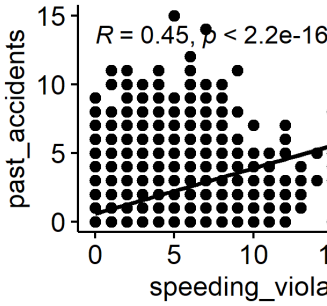

Figure 2

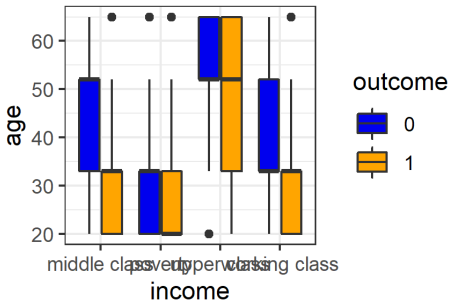

Figure 3

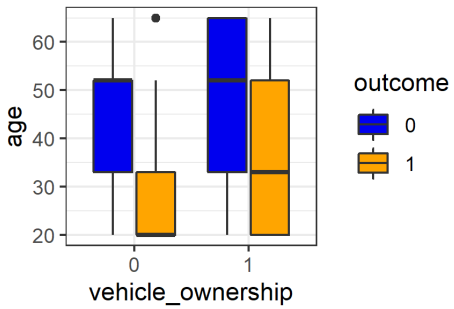

Figure 4

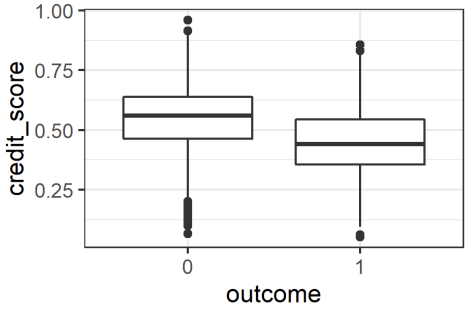

Figure 5: outcome is insurance claim

c.Classification

The data has splitted randomly to 0.75 as training and 0.25 as testing. and we fit the models with training data then we predict the response with test data. we used two KNN and logistic regression models. we used backward way for feature selection in our model and check our model's accuracy to see the result.

c.1 K-nearest neighbour

For this classification model we need K that based on this link generally one of the ways for calculating is sqrt(n) that n is sample size. Also we can run our model with different list of K and choose the best result that you can see the result in table.

Table 2: K determination

K	Accuracy
43	0.7938
33	0.7934
31	0.7930
27	0.7925
45	0.7921
29	0.7916

We fit the model with K=43 and train and test samples. we repeat the model 10 times for recording the accuracy as table3.

Table 3: Prediction Accuracy

Run	KNN Accuracy	LR Accuracy
1	0.79	0.80
2	0.78	0.81
3	0.75	0.83
4	0.79	0.83
5	0.78	0.82
6	0.77	0.82
7	0.79	0.82
8	0.78	0.81
9	0.78	0.81
10	0.77	0.80
Ave	0.7830	0.8139

c.2 logistic regression

We fit model logistic regression based on the train and test data and run it 10 times. Also we consider that labels with fitted probability more than 50% are correct and we plot the accuracy probability with samples as figure6. It clearly shows that the logistic regression model works well based on this dataset. Also based on figure8 it shows there is not much outlier.

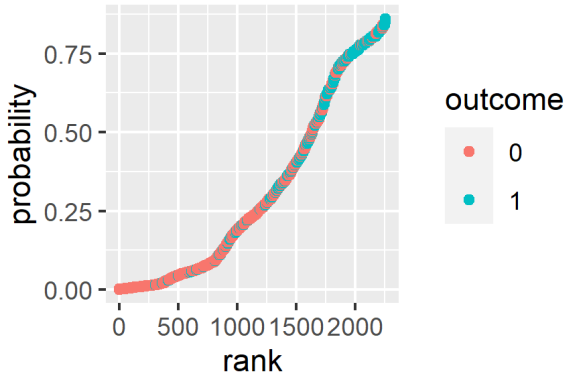

Figure 6: logistic regression fitted probability

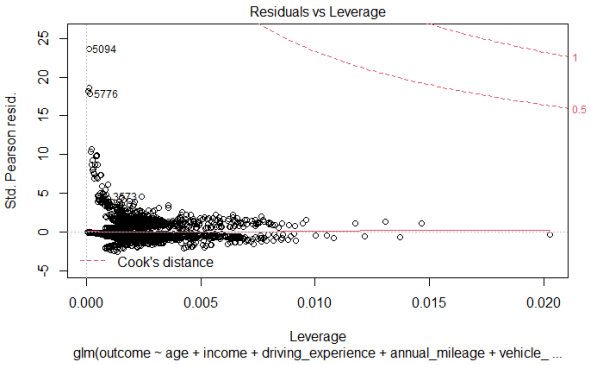

Figure 7: residual Vs Fitted

c.3 Evaluate the model performance

we predict the output label based on two models. for comparing the accuracy we look at confusion matrix and count the correct classes. that shows accuracy of Logistic regression is a litter more. The total accuracy is around 82% which shows that there is more opportunity to improve the model performance. one of reasons is that KNN works well with the numeric features, and in this dataset we have mix of both numeric and and categorical. Since logistic regression is a parametric model and Also in summary of model for statistical tests, we see that median of "Deviance Residuals" is 0.2600. In the coefficient part Pvalues of vehicle year and driving experience is quite small and shows both are statistically significantGareth James (2013). Also based on result of Vif() for our logistic regression it shows all variables are less than 10 that shows well independence between predictors, in table

Table 4: measure of multicollinearity

	GVIF	Df	GVIF^(1/(2*Df))
age	1.87	1.00	1.37
income	1.63	3.00	1.08
driving_experience	2.48	1.00	1.58
annual_mileage	1.14	1.00	1.07
vehicle_year	1.09	1.00	1.04
vehicle_type	1.00	1.00	1.00
speeding_violations	1.84	1.00	1.36
past_accidents	1.49	1.00	1.22

References

(2021). insurance. https://www.kaggle.com/racholsan/customer-data. URL.

Gareth James, Daniela Witten, T. H. R. T. (2013). *An introduction to statistical learning : with applications in R*. New York :Springer.

John W Emerson, W. A. G. (2012). The generalized pairs plot. *Journal of Computational and Graphical Statistic*.

Assignment 2- Supplementry

Paria Fakhrzad

10/3/2021

Contents

Part a

loading libraries

```
library(dplyr)
library(Hmisc)
library(magrittr)
library(readr)
library(ggplot2)
library(ISLR2)
library(class)
library(ggpubr)
```

```
library(GGally)
library(PreProcess)
library(caTools)
library(caret)
```

loading dataset

the source of data[1]

```
customer_claim <- readr:: read_csv("customer-data.csv")
customer_claim <- dplyr::select(customer_claim,-id)
```

tidying data

```
#finding null observation
customer_claim <- as_tibble(customer_claim)
table(sapply(customer_claim,function(x)all(is.na(x))))#Columns are totally empty
table(lapply(customer_claim,function(x){length(which(is.na(x)))})) #Columns with NA
customer_claim  <- dplyr::filter(customer_claim , !is.na(annual_mileage))
#customer_claim  <- dplyr::filter(customer_claim , !is.na(credit_score))

#change gender/car ownership/married/children to factor
customer_claim$gender <- as.factor
(ifelse(customer_claim$gender=="female",1,0))
customer_claim <- mutate(customer_claim, outcome =
                           factor(outcome, levels = c("FALSE", "TRUE"),
                                  labels = c(0, 1)))
customer_claim$married <- as.factor(ifelse(customer_claim$married=="TRUE",1,0))
customer_claim$children <- as.factor(ifelse(customer_claim$children=="TRUE",1,0))
customer_claim$vehicle_ownership <- as.factor(
  ifelse(customer_claim$vehicle_ownership=="TRUE",1,0))
customer_claim$income <- as.factor(customer_claim$income)
customer_claim$education <- as.factor(customer_claim$education)
customer_claim$income <- as.factor(customer_claim$income)
customer_claim$vehicle_year <- as.factor(customer_claim$vehicle_year)
customer_claim$vehicle_type<- as.factor(customer_claim$vehicle_type)

# changing the age to numbers
customer_claim$age <- ifelse(customer_claim$age=="65+",65,customer_claim$age)
customer_claim$age <- ifelse(customer_claim$age=="16-25",20,customer_claim$age)
customer_claim$age <- ifelse(customer_claim$age=="26-39",33,customer_claim$age)
customer_claim$age <- ifelse(customer_claim$age=="40-64",52,customer_claim$age)
customer_claim$age <- as.numeric(customer_claim$age)
```

[1] https://www.kaggle.com/racholsan/customer-data

```
# changing the driving experience to number
customer_claim$driving_experience <- ifelse(customer_claim$driving_experience=="0-9y",5,
                                            customer_claim$driving_experience)
customer_claim$driving_experience <- ifelse(customer_claim$driving_experience=="10-19y",15,
                                            customer_claim$driving_experience)
customer_claim$driving_experience <- ifelse(customer_claim$driving_experience=="20-29y",25,
                                            customer_claim$driving_experience)
customer_claim$driving_experience <- ifelse(customer_claim$driving_experience=="30y+",35,
                                            customer_claim$driving_experience)
customer_claim$driving_experience <- as.numeric(customer_claim$driving_experience)
```

Part b

Summary of variables

- `customer_claim` is a data frame with 10000 observations on 18 variables.

```
xtable::xtable(summary(dplyr::select(customer_claim,c(age,gender,driving_experience
               ,married,children,postal_code,education))))
xtable::xtable(summary(dplyr::select(customer_claim,c(race,education,income,
                credit_score, vehicle_year))))
xtable::xtable(summary(dplyr::select(customer_claim,c(vehicle_ownership, vehicle_type,
                annual_mileage, speeding_violations,past_accidents))))

dim(customer_claim)
str(customer_claim)

table(as.numeric(customer_claim$outcome))/sum(as.numeric(customer_claim$outcome))
```

Explore data

- There are eighteen variables in the data set
- outcome *label* is a factor column that has two labels, false(has claimed) or true(has not claimed)
- age x_1, a range variable that shows the age of customer is in which interval
- gender x_2,
- race x_3,
- driving_experience x_4,
- income x_5,
- credit_score x_6,
- vehicle_ownership x_7,
- vehicle_year x_8,
- married x_9,
- children x_{10},

- annual_mileage x_9,
- vehicle_type x_{10},
- speeding_violations x_{11},
- past_accidents x_{12},

```
# calculatin the number of response per each categorical variable
xtabs(~outcome+age,customer_claim)
xtabs(~outcome+gender,customer_claim)
xtabs(~outcome+race,customer_claim)
xtabs(~outcome+driving_experience,customer_claim)
xtabs(~outcome+education,customer_claim)
xtabs(~outcome+income,customer_claim)
xtabs(~outcome+vehicle_ownership,customer_claim)
xtabs(~outcome+vehicle_type,customer_claim)
xtabs(~outcome+vehicle_year,customer_claim)
xtabs(~outcome+married,customer_claim)
xtabs(~outcome+children,customer_claim)
```

using visualization methods

- using ggpairs() for founding the correlation of numeric features:

```
customer_corr_matrix <- dplyr::select(customer_claim,c("age","driving_experience","annual_milea
                                                       "speeding_violations","past_accidents",'

figure <- ggpairs(customer_corr_matrix)
figure1 <-ggpairs(customer_corr_matrix, columns = 1:7,
                  ggplot2::aes(colour=outcome))
ggsave("figure1.png",figure1, width=10, height = 4)
figure1
figure
```

- using box-plot and scatter-plot for figuring out the relationships between some categorical variables with "response" that here is ourcome(1:TRUE, 0:FALSE)

```
# speeding_violations and past accidents
figure2 <-ggscatter(customer_claim,x ='speeding_violations',y ='past_accidents',
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "pearson")+
          scale_color_manual(values = c("green", "red"))

# income and age
figure3 <- ggplot(customer_claim) +
  geom_boxplot(aes(x = income, y = age, fill = outcome)) +
  theme(legend.position = "none") +
  theme_bw() +
```

```
  scale_fill_manual(values = c("blue2", "orange"))

# vehicle ownership and age
figure4 <- ggplot(customer_claim) +
  geom_boxplot(aes(x = vehicle_ownership, y = age, fill = outcome)) +
  theme(legend.position = "none") +
  theme_bw() +
  scale_fill_manual(values = c("blue2", "orange"))

# credit score and outcome
figure5 <- ggplot(customer_claim) +
  geom_boxplot(aes(x = outcome, y = credit_score)) +
  theme(legend.position = "none") +
  theme_bw()


figure2
figure3
figure4
figure5

ggsave("figure2.png",figure2,width = 3, height = 2)
ggsave("figure3.png",figure3,width = 3, height = 2)
ggsave("figure4.png",figure4,width = 3, height = 2)
ggsave("figure5.png",figure5,width = 3, height = 2)
```

We also have used cor() function to calculated the correlation between features.

Partc : Classification

Goal

We will use KNN to predict the label of outcome that shows if customer had insurance claim or not. for this mean we need two split our data as train and test. train_data: split customers by sample (75%) * test_data: split customers by sample (25%)

```
#defining sample size and randomly split data solution 1
training_size = floor(0.75*nrow(customer_claim))
train_data_old <- sample(seq_len(nrow(customer_claim)),size = training_size)
test_data_old <- customer_claim[-train_data_old,]
train_data_old <- customer_claim[train_data_old,]

#defining sample size and randomly split data solution 2
ts_split <- createDataPartition(customer_claim$outcome, p = 0.75, list = FALSE)
train_data<- customer_claim[ts_split,]
test_data<- customer_claim[-ts_split,]
```

```
#defining the column label for train and test
Direction_train_claim <- dplyr::pull(train_data, outcome)
Direction_test_claim <- dplyr::pull(test_data, outcome)
```

KNN

- for defining K in our model we use $squarerootofnumberpfsamples$ based on Thumb rule in this link[2]. Also after fitting KNN model We test our accuracy for K from 1 till 101
- fitting KNN model

```
predictor_train <-dplyr::select(train_data,age,driving_experience,
                                annual_mileage,past_accidents,speeding_violations)
predictor_test <- dplyr::select(test_data, age, driving_experience,
                                annual_mileage,past_accidents,speeding_violations)

knn_model <- knn(predictor_train, predictor_test, train_data$outcome,k = 43)
```

- Repeat the K in KNN model

```
accuracy_table <- data.frame(matrix(ncol = 2, nrow= 0))
col_name <- c('accuracy', 'k')
colnames(accuracy_table) <- col_name
for(i in seq(from = 1, to = 101, by = 2)){
  knn_model1 <- knn(predictor_train, predictor_test, train_data$outcome,k = i)
  accuracy <- mean(knn_model1 == Direction_test_claim)
  accuracy_table[i,'accuracy'] <- accuracy
  accuracy_table[i, 'k'] <- i
}
accuracy_table <- accuracy_table[order(accuracy_table$accuracy, decreasing = TRUE),]
head(accuracy_table,10)
xtable::xtable(head(accuracy_table,10))
```

Logistic regression

```
LR_model <- glm(outcome~age+income+driving_experience+annual_mileage+
                vehicle_year+vehicle_type+speeding_violations+past_accidents,
                data=train_data,
                family = binomial("logit"))
# model Specification
summary(LR_model)
names(LR_model)
```

[2]https://discuss.analyticsvidhya.com/t/how-to-choose-the-value-of-k-in-knn-algorithm/2606/13

```
ggsave ("figure8.png",plot(LR_model))
summary(LR_model)$r.sq
xtable::xtable(vif(LR_model))
```

- Here we use test data to predict the response by logistic regression model

```
#predict with test data
LR_predict <-predict(LR_model, newdata = test_data, type = "response")

#set the predicted vector and relar vector and probability in a table
predicted.data <- data.frame(probability=LR_predict, outcome=test_data$outcome)
predicted.data <- predicted.data[order(predicted.data$probability, decreasing = FALSE),]
predicted.data$rank <- 1:nrow(predicted.data)
predicted.data$test_outcome <- as.factor(ifelse(predicted.data$probability >= 0.5,1,0))

# Probability per response
figure7 <- ggplot(predicted.data, aes(x= rank, y=probability, col=outcome))+
           geom_point()+scale_fill_manual(values = c("red", "green"))
ggsave("figure7.png",figure7,width = 3, height = 2)
```

Evaluation

- Confusion matrix

```
# KNN accuracy
KNN.predicted.data <-data.frame(knn_model, Direction_test_claim)
table(knn_model, Direction_test_claim)
mean(knn_model == Direction_test_claim)

#Logistic regression accuracy
table(predicted.data$test_outcom,predicted.data$outcome )
mean(predicted.data$test_outcome == predicted.data$outcome)
```

- repeat KNN and logistic regression models for 10 times. here we used for function.

```
KNN_running_table <- data.frame(matrix(ncol = 2, nrow= 0))
LR_running_table <- data.frame(matrix(ncol = 2, nrow= 0))
colnames(KNN_running_table) <- c('Accuracy', 'Run')
colnames(LR_running_table) <- c('Accuracy', 'Run')
for(i in seq(from = 1, to = 10, by = 1)){

  #Splitting data
  ts_split <- createDataPartition(customer_claim$outcome, p = 0.75, list = FALSE)
  train_data<- customer_claim[ts_split,]
  test_data<- customer_claim[-ts_split,]
```

```
  predictor_train <-dplyr::select(train_data,age,driving_experience,
                                  annual_mileage,past_accidents,speeding_violations)
  predictor_test <- dplyr::select(test_data, age, driving_experience,
                                  annual_mileage,past_accidents,speeding_violations)
  Direction_train_claim <- dplyr::pull(train_data, outcome)
  Direction_test_claim <- dplyr::pull(test_data, outcome)

  #knn model
  knn_model2<- knn(predictor_train, predictor_test, train_data$outcome,k = 43)

  #Logistic regression model
  LR_model <- glm(outcome ~age+education+income+postal_code+driving_experience+
                  annual_mileage+vehicle_year+vehicle_type+
                  speeding_violations+past_accidents,
                  data=train_data,
                  family = binomial("logit"))
  LR_predict <-predict(LR_model, newdata = test_data, type = "response")
  LR_predicted_table <-ifelse(LR_predict >= 0.5,1,0)

  #filling the accuracy table
  KNN_accuracy <- mean(knn_model2 == Direction_test_claim)
  KNN_running_table[i,'Accuracy'] <- KNN_accuracy
  KNN_running_table[i, 'Run'] <- i
  LR_accuracy <- mean(LR_predicted_table == Direction_test_claim)
  LR_running_table[i,'Accuracy'] <- LR_accuracy
  LR_running_table[i, 'Run'] <- i
}
#making table from output and calculating the average
xtable::xtable( KNN_running_table)
mean(KNN_running_table$Accuracy)
xtable::xtable( LR_running_table)
mean(LR_running_table$Accuracy)
```