Assignment4 Feature selection-CSE780

Paria, Fakhrzad
Stuent ID: 400353290

21-Nov-2021

Part a. Data

Nowadays having property insurance is one of the most important sections in people life. One of insurance contracts is related to cars. In this report we will use a dataset related to an insurance company [1]. There are 10000 customers in this data. Also there are 17features.The response column is a binomial factor variable that shows whether customer had insurance claim last year or not. We will see that how with these features to predict the customer behavior in claiming the insurance for accidents.
There are 1939 NA observations in dataset that all have removed.

Part b. Spars and non-spars models

In this report, we will discuss the result of fitting three models 1)Ridge regression, 2)LASSO, 3)PLS and also will use best subset selection algorithm for choosing the best predictors that have relevant to our prediction. These models have been choosen because all of them help us to find the most significant features that can explain the response. The comparison metric in this report is train MSE and accuracy percentage of test prediction.

Before fitting models to our dataset, since we have factor and categorical features, so we convert them to dummy features for better prediction. So the number of features will be 24 after this transformation. Also regarding features are measured in different scales we standardize the dataset with mean of zero and standard deviation 1. We split the observations to train(50%) and test(50%) to estimate the test error.

b.1 Ridge Regression

In ridge regression model we will have all the predictors/features in our model and as can seed in figure1 the number of features is 24. We need to specify the optimal tuning parameter λ and here we used cross validation in the train part of our dataset. The best λ is 0.01. Then we predict with the test part of our dataset and we can see that MSE=0.268 and accuracy based on confusion matrix is 57%. In tabel1 we see the lowest regression coefficient for ridge regression that as our expectation, non of them are zero.

Table 1: Feature Coefficient

Feature	Ridge Regression	Feature	LASSO
raceminority	-0.0028	raceminority	0
educationuniversity	0.0016	educationuniversity	0
educationnone	-0.0016	educationnone	0
credit_score	-0.0022	credit_score	0.0014

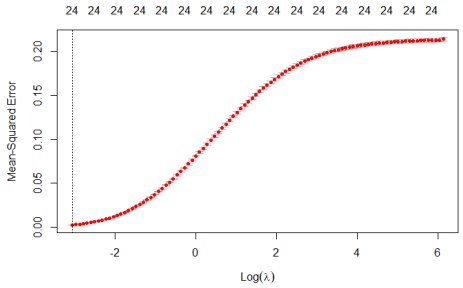


Figure 1: MSE and Lambda under Ridge regression

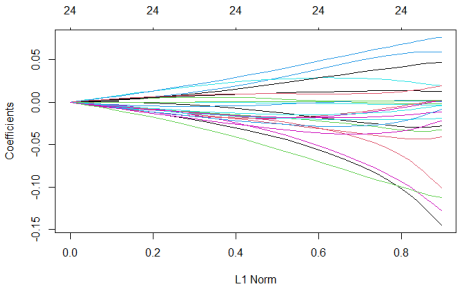


Figure 2: Coefficient under Ridge regression

b.2 LASSO

We fit lasso model for the same train data and in figure 2 shows that in this model some of coefficient would be zero. The best λ in this model is 0.001 and MSE=0.267, Accuracy is 57%.

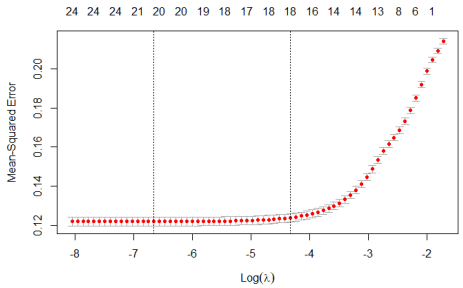


Figure 3: MSE and Lambda under LASSO

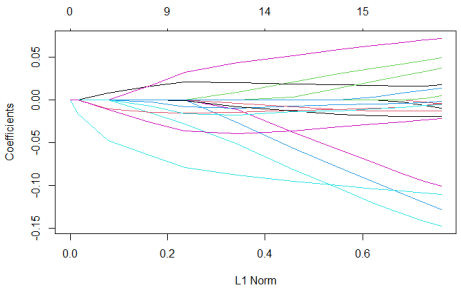


Figure 4: coefficient and Lambda under LASSO

b.3 Partial Least Square PLS

With fitting PLS in this dataset we can see in figure6 that CV error minimized at component=3. When we predict the response with test data, can see than the accuracy is 93% and MSE=0.12.

b.4 Best subset selection

With performing the best subset selection in this dataset we will see that DUI, raceminority, educationnone, educationuniversity, incomeupper class,credit_score and speeding_violations have the least effect in selecting different models and with comparing to Tabel1 , somehow we can see that the features with zero coefficient are in this list. Based on adjusted R^2 in feature5, the number of features selected 17 from 24.

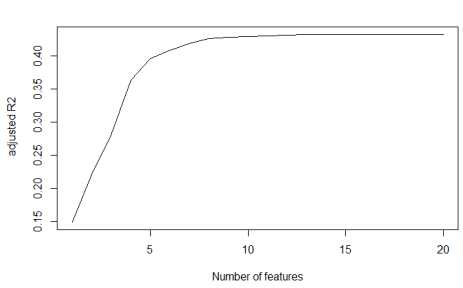


Figure 5: R^2 and number of features under best subset selection

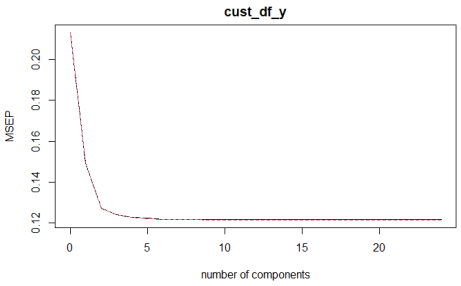


Figure 6: optimal number of components under PLS

Part c. Result

As we can see in tables1, features which in ridge regression had very small coefficient in the best λ in RASSO their coefficients are zero. Also and comparison in tabel2, the accuracy in RASSO is better than ridge regression. The result shows that the most accurate model in this case in PLS, since it uses the label for feature extraction also in this model all features will be used for forming the first three components and here we can conclude that there are small collinearity between features.

Table 2: Models comparison

Model	**Type**	**Test MSE**	**Test Accuracy**
Ridge regression	non-sparse	0.267	55%
LASSO	sparse	0.267	57%
PLS	feature extraction	0.127	92%

References

[1] Insurance dataset. [Online].Available: https://www.kaggle.com/racholsan/customer-data, 2021.

Assignment4- Sparse models // Supplementry

Paria Fakhrzad

11/21/2021

Contents

Part a- Data

1.1- 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)
```

```
library(tree) # CART
library(MASS)
library(mclust) # for Gaussian Mixtures
library(car)
library(boot)
library(e1071)
library(leaps)
library(glmnet)
library(pls)
```

1.2- loading dataset

In this assignment we are using a dataset related to insurance company[1]. There are 10000 samples in this dataset that are customers of this company and their 18 features in year 2020~[**?**]. There are 10000 customer as sample in this data frame.

```
cust_df_original <- readr:: read_csv("customer-data.csv")
cust_df <- cust_df_original
```

1.3- data tidying

- Outliers and missing values Here we can see that 957 samples have null value in annual_mileage and 982 samples have null value in credit_score.Therefore in we omit these `NA` observations from our dataset.

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

For using Spars method we need have numerical features so we use model.matrix() for this mean:

```
#change the label to factor
cust_df_new <- mutate(cust_df, outcome =
                        factor(outcome, levels = c("FALSE", "TRUE"),
                        labels = c(0, 1)))
cust_df_x<-model.matrix(outcome~.,cust_df)
cust_df_x <- scale(cust_df_x)
cust_df_x<-cust_df_x[,-2]
cust_df_y <-cust_df$outcome
cust_df_new <-as_tibble(cbind(cust_df_x,cust_df_y))
```

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

```
## Warning in cbind(cust_df_x, cust_df_y): number of rows of result is not a
## multiple of vector length (arg 2)
```

```
cust_df_new <- dplyr::select(cust_df_new,-1)
cust_df_y <-ifelse(cust_df_y=="TRUE",1,0)

#Split data to tarin and test
 ts_split <- createDataPartition(cust_df_new$cust_df_y, p = 0.5, list = FALSE)
 train_data<- cust_df_new[ts_split,]
 test_data<- cust_df_new[-ts_split,]
 train_matrix <- as.matrix(train_data)
 test_matrix<- as.matrix(test_data)

 label_train <- dplyr::pull(train_data, cust_df_y)
 label_test <- dplyr::pull(test_data, cust_df_y)
```

Partb- feature selection Methods

2.1- subset selection

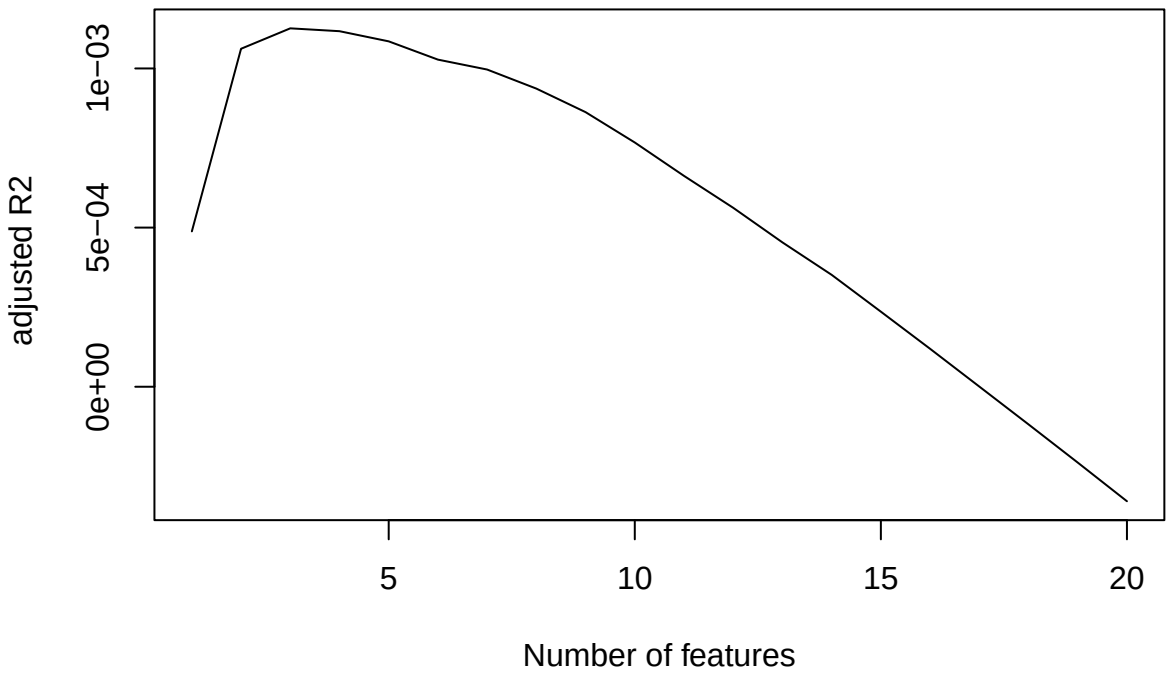
- Best Subset selection

```
best_subset_model <- regsubsets(cust_df_y~ .,
                                data=cust_df_new,
                                nvmax=20)
#summary(best_subset_model)
names(summary(best_subset_model))
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

It shows that driving_experience, vehicle_year and vehicle_ownership are the most significant features

```
plot(summary(best_subset_model)$adjr2,
     xlab="Number of features",
     ylab=" adjusted R2",
     type="l")
```




```
which.max(summary(best_subset_model)$adjr2)
```

```
## [1] 3
```

```
summary(best_subset_model)$which[which.max(summary(best_subset_model)$adjr2),]
```

```
##                 (Intercept)                  `age26-39`
##                        TRUE                       FALSE
##                  `age40-64`                   `age65+`
##                       FALSE                       FALSE
##                  gendermale                raceminority
##                       FALSE                        TRUE
## `driving_experience10-19y` `driving_experience20-29y`
##                       FALSE                       FALSE
##   `driving_experience30y+`              educationnone
##                       FALSE                       FALSE
##        educationuniversity              incomepoverty
##                       FALSE                        TRUE
##        `incomeupper class`      `incomeworking class`
##                       FALSE                       FALSE
##               credit_score      vehicle_ownershipTRUE
```

```
##                       FALSE                       FALSE
##  `vehicle_yearbefore 2015`                 marriedTRUE
##                       FALSE                        TRUE
##                childrenTRUE                 postal_code
##                       FALSE                       FALSE
##             annual_mileage   `vehicle_typesports car`
##                       FALSE                       FALSE
##        speeding_violations                        DUIs
##                       FALSE                       FALSE
##             past_accidents
##                       FALSE
```

2.2- Ridge Regression

```
#fit ridge regression with whole data
grid <- 10^seq(10,-2,length=100)
ridge_model <- glmnet(cust_df_x, cust_df_y, alpha = 0, lambda = grid)
plot(ridge_model)

#cross validation
set.seed(10)
cv_out <- cv.glmnet(train_matrix[,-cust_df_y], label_train , alpha = 0,lambda = grid)

plot(cv_out)
bestlam <- cv_out$lambda.min
bestlam

ridge_pred <- predict(ridge_model, s = bestlam, newx = test_matrix)
mean((ridge_pred - label_test)^2)

#Accuracy
mean(abs(ridge_pred - label_test)<.5)

#use whole dataset
ridge_model2 <- glmnet(cust_df_x, cust_df_y, alpha = 0)
predict(ridge_model2, type = "coefficients", s = bestlam)[1:26, ]
```

2.3- The Lasso

```
#fit lasso model
lasso_model <- glmnet(cust_df_x, cust_df_y, alpha = 1, lambda = grid)
plot(lasso_model )

#Cross validation
```

```
set.seed(10)
cv_lasso <- cv.glmnet(cust_df_x, cust_df_y, alpha = 1)
plot(cv_lasso)

bestlam <- cv_lasso$lambda.min
bestlam
lasso_pred <- predict(lasso_model , s = bestlam, newx = test_matrix)
mean((lasso_pred - label_test)^2)

#Accuracy
mean(abs(lasso_pred - label_test)<.5)
predict(lasso_model, type = "coefficients", s = bestlam)[1:26, ]
```

2.4- PLR

```
set.seed(10)

#fit PLS
pls_model <- plsr(cust_df_y ~ ., data = train_data, validation = "CV")
summary(pls_model)
```

```
validationplot(pls_model, val.type = "MSEP")
```

```
pls_pred <- predict(pls_model,test_data , ncomp = 3)
mean((pls_pred - label_test)^2)
mean(abs((pls_pred - label_test)<0.5))
```