Math 448 Final Project

Which factors affect the length of Employment?

Pyimoe Than

May 17,2022

Contents

1. Executive Summary

I used data from Kaggle “HR Data for Analytics” to learn to predict the length of the employment. I split the data into 80% training and 20% testing sets. In order to perform prediction, I apply Linear Regression, Ridge Regression, Lasso Regression, PCR, PLS, Decision Tree, and bagging.

First, I apply Linear Regression. MSE using Linear Regression is 1.97. RMSE using Linear Regression is 1.40. Our prediction using Linear Regression is 1.4044 years away from the test value. However, the disadvantages of Linear Regression is that it assumes linearity between inputs and outputs. In addition, it can underfit with small and high-dimensional data. I later applied Ridge, Lasso, PCR and PLS to see which one would give the lowest MSE than Linear Regression. I found out that those modeling also give the same RMSE value just like Linear Regression.

I further fit the Decision Tree and bagging method on the training data set. They significantly reduce the RMSE. RMSE using Decision Tree is 1.368. RMSE using Bagging is 1.028. Thus, Decision Tree and Bagging improve prediction accuracy. I conclude that the Decision Tree and Bagging method are good for predicting how long employees would stay with the company.

This time I assume that the data is linear. In the future work, I might need to assume the data is non linear. Hence, I need to use non-linear models such as Polynomial regression to see which one gives better prediction accuracy.

2. Introduction

I have worked at the Amazon delivery station as a Learning Ambassador for almost 2 years. Within my 2 years working at Amazon, they hired so many employees because most of them quit before their 1 year anniversary. Employees leaving the company can impact the

company productivity. Because of that, I've trained countless new employees about "Amazon work of standard" which is basically about how to do the job safely. One day at work, I was thinking how many employees would quit the company. My interest in the company motivates me to learn more about how many years employees would stay with the company. In order to perform prediction, I would use Linear Regression, Ridge Regression, Lasso Regression, PCR, PLS, Decision Trees and Bagging.

3. **The Data**

I could not find any data sets about Amazon on the internet. Thus, I use an alternative data set from Kaggle. The data set name is HR data for analytics. This data is about Human resources in a specific company. Each record represents an employee. This data set has 14999 rows and 10 columns(variables). Variable overview:

- employee satisfaction level(it means whether employee satisfied or not: 0 means less satisfied, 1 most satisfied)
- Last_evaluation(it means employee's evaluation score for last month(0 bad, 1 excellent)
- Number_project(it means the number of projects employees work during the employment),
- Average_monthly_hours(it means average months employee spends at work per month), Time_spend_company(it means years the employee spent in a company)
- Work_accident(0 if he did not have an accident , 1 if had at least one)
- promotion_last_5years : 0 if he did not have any promotion in last 5 years , 1 if had at least one
- sales : department in which employee works
- Left : 0 if employee did not leave , 1 if left company
- salary(it means employees salary)

4. **Data Cleaning**

My data cleaning process involves checking data integrity which means checking accuracy, completeness and consistency. Fortunately, this data doesn't have any missing values and replication data. Originally, variables work_accident, left, promotion_last_5years are in int

format. I transformed those into binary variables. Transform variables time_spend_company, number_project, average_monthly_hours from int to numeric format. In addition, I renamed the 'sales' column into a 'department' to make it easy for me to analyze. Additionally, I split the data into 80% training set and 20% test set.

5. Model Selection

a. Linear Regression

First, I decided to apply Linear Regression on the training set because Linear Regression is faster to train than other machine learning models. However, the disadvantages of using linear Regression is that it assumes linearity between inputs and output. I use the full model in this prediction. I found out that all predictors are significant except the work_accident column.

```
Call:
lm(formula = time_spend_company ~ satisfaction_level + last_evaluation +
    number_project + average_montly_hours + promotion_last_5years +
    salary + Work_accident, data = HR.train)

Residuals:
    Min      1Q  Median      3Q     Max
-2.5858 -0.7806 -0.2700  0.4693  7.0919

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)            2.6790112  0.0857239  31.252  < 2e-16 ***
satisfaction_level    -0.5569696  0.0531844 -10.472  < 2e-16 ***
last_evaluation        0.7245990  0.0840014   8.626  < 2e-16 ***
number_project         0.1633131  0.0121525  13.439  < 2e-16 ***
average_montly_hours   0.0011265  0.0002927   3.849 0.000119 ***
promotion_last_5years  0.7291273  0.0906904   8.040 9.85e-16 ***
salarylow             -0.2928591  0.0488182  -5.999 2.04e-09 ***
salarymedium          -0.2111233  0.0491278  -4.297 1.74e-05 ***
Work_accident          0.0487588  0.0372156   1.310 0.190163
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.416 on 11990 degrees of freedom
Multiple R-squared:  0.06175,   Adjusted R-squared:  0.06112
F-statistic: 98.64 on 8 and 11990 DF,  p-value: < 2.2e-16

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.746   3.228   3.442   3.488   3.672   4.873
[1] 2.008001
[1] 1.41704
```

The MSE is 2.00. The RMSE is 1.417. That means my prediction using linear Regression is 1.417 years away from the test value.

b. Ridge Regression

Ridge Regression is used for minimizing RSS. Minimizing RSS will lead to better prediction accuracy by introducing the shrinkage penalty. Shrinkage penalty shrinks the coefficient estimates towards approximately zero. I perform Ridge Regression on the training data set because it is less prone to overfitting when using regularization. To perform Ridge Regression, I use cross-validation to figure out which tuning parameter lambda gives the smallest MSE.

I found out that lambda value 0.029 gives the smallest MSE. MSE using Ridge Regression is 2.00. RMSE using Ridge Regression is 1.417. Thus, our prediction using Ridge Regression is 1.417 years away from the test value.

c. Lasso Regression

Lasso Regression is similar to ridge regression. Lasso is considered to be a sparse regression model because it shrinks the coefficient estimates towards zero and only a small number are non-zero. The advantages of using Lasso Regression is that it solves the overfitting issue using the Linear Models. And also, it works well with a large number of features. In order to determine the best tuning parameter, lambda value, I use cross validation.

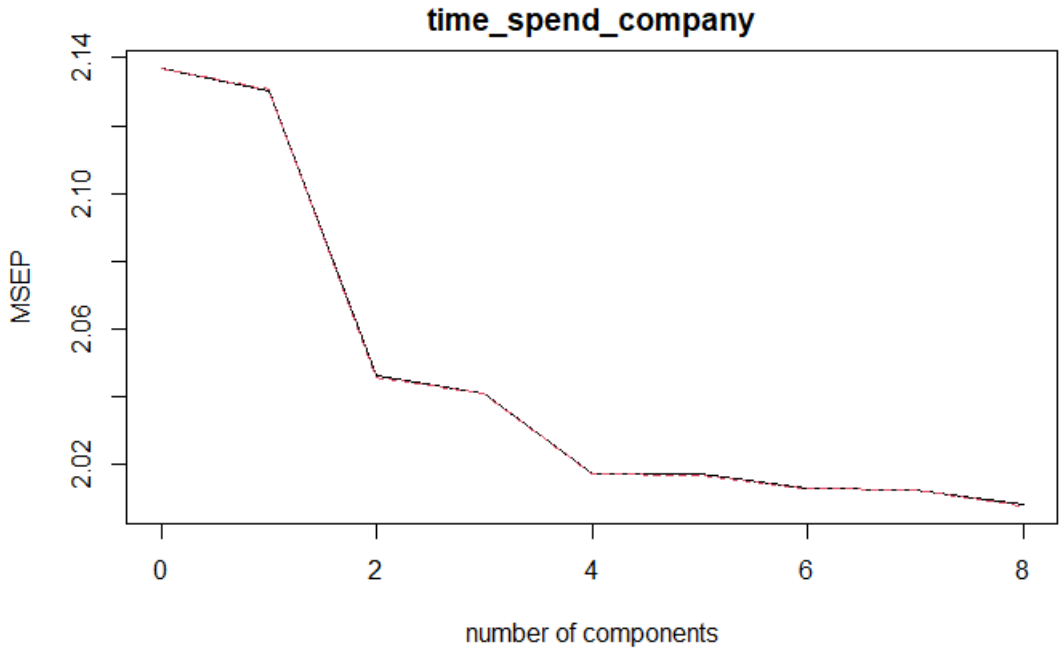
The Lambda value using Lasso Regression is 0.00051116. MSE is 2.00. RMSE is 1.417. Thus, our prediction using Ridge Regression is 1.417 years away from the test value.

d. PCR method

PCR is a dimension reduction method. PCR is for computing regression when the explanatory variables are highly correlated because it converts highly correlated explanatory variables into a set of linearly uncorrelated variables. The downside of using PCR is that PCR does not incorporate the response variable. Therefore, there is no guarantee that the directions that best explain the predictors will also be the best directors to predict the response.

Just like Lasso Regression, I use the Cross-validation method to see which M value gives the smallest MSE. The lowest MSE occurs when M=6.

Tuning Parameter Selection



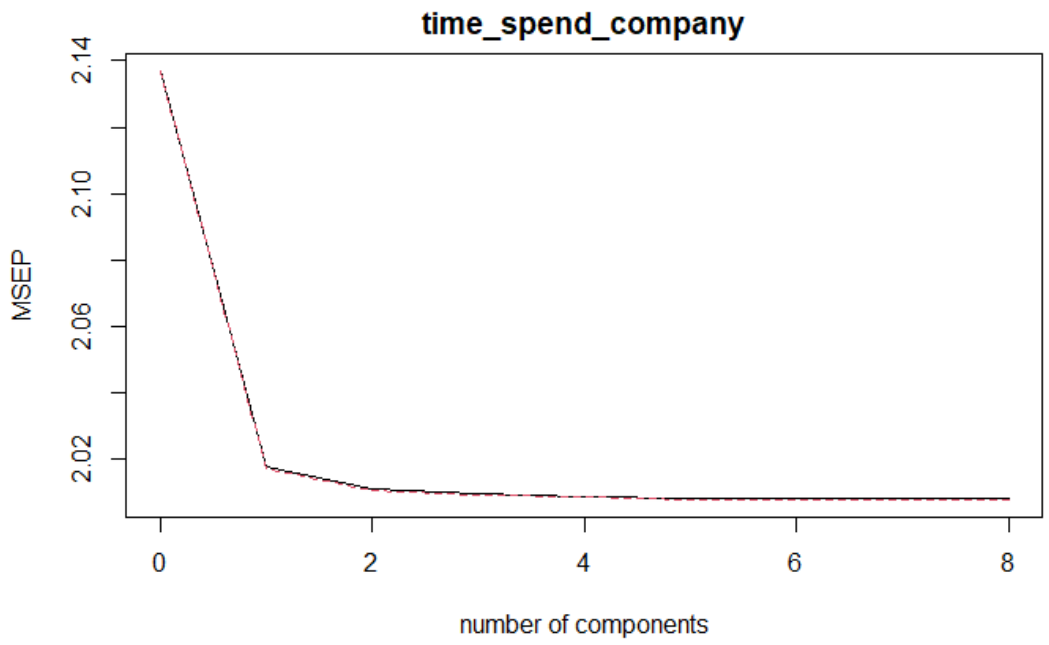
MSE using PCR is 2.00. RMSE is 1,417. Thus, our prediction using PCR is 1.417 years away from the test value.

e. PLS method

Unlike PCR, PLS is incorporated with the response variable. Therefore, there is a high chance that the directions that best explain the predictors will also be the best directors to predict the response.

Just like Lasso Regression, I use the Cross-validation method to see which M value gives the smallest MSE. The lowest MSE occurs when M=4.

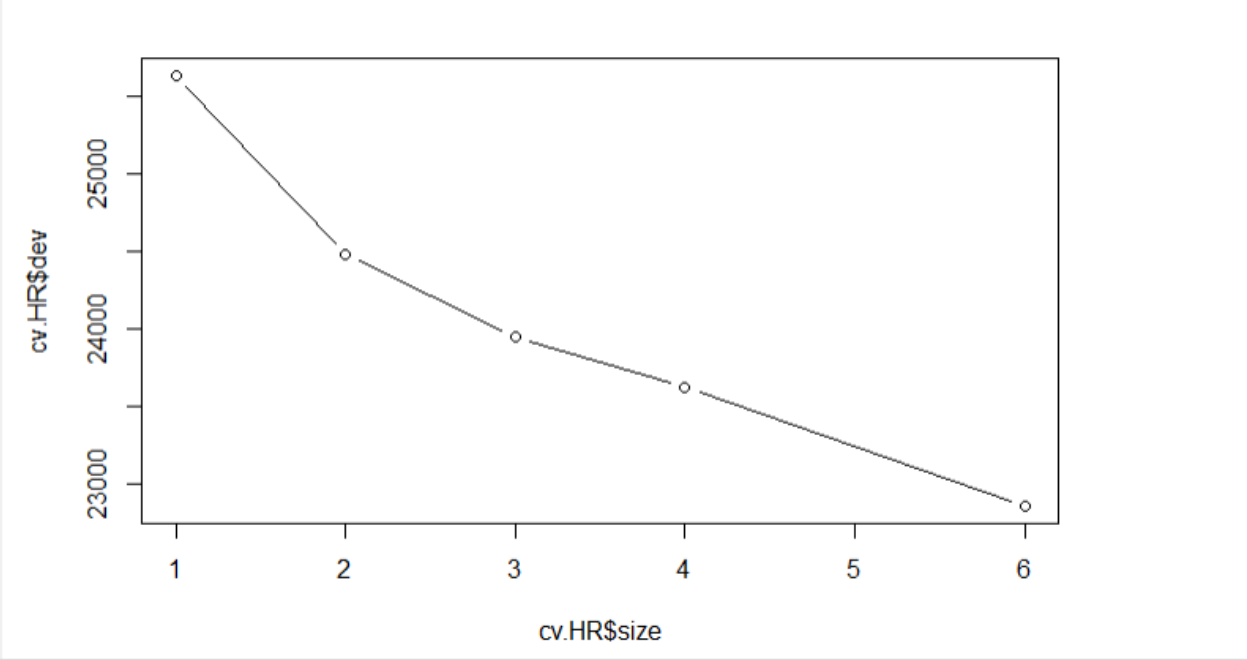
Tuning Parameter Selection



MSE using PLS is 2.00. RMSE using PLS is 1.417. Thus, our prediction using PLS is 1.417 years away from the test value.

f. Decision Trees

Decision trees can be applied to both regression and classification problems. It is a non-parametric supervised learning method. It is easy to interpret and understand. Prediction accuracy is based on bias-variance trade off. The advantages of using decision trees is that it is explainable and interpretable. However, it is prone to overfitting. Pruning a decision tree to prevent the training data from overfitting. Using a cross-validation method to see which alpha value gives the smallest MSE. Best alpha value is 6.



(i) Tuning parameter Selection

MSE using Decision Trees is 1.92. RMSE using Decision Trees is 1.386. Thus, our prediction is 1.386 years away from the test value.

g. Bagging

Decision trees such as regression or classification trees suffer from high variance. Bootstrap aggregation or bagging is a general-purpose procedure for reducing the variance of a statistical learning method. In the Bagging method, these trees are grown deep, and not pruned.

Averaging these trees reduces variance. Therefore, our final prediction would have low variance and bias. However, it decreases the model interpretability because we have hundreds of trees instead of a single tree. In order to find out which predictors are important we need to use a variable importance plot.

Variable Selection

```
[1] 1.028081
                         %IncMSE IncNodePurity
satisfaction_level    171.01809     5826.1065
last_evaluation       131.89442     4624.1636
number_project         95.61075     1550.9228
average_montly_hours  135.46039     5908.6094
work_accident          33.62277      576.0711
left                  108.52672     1815.3487
promotion_last_5years  45.70565      339.7950
department             68.96765     2271.7769
salary                 52.32328      865.8284
```

RMSE using bagging is 1.028. Using the importance() function, all the predictors are important except the work accident column.

6. **Conclusion**

Compare different classes of regression models

Model	Metrics	Results
Linear Regression	RMSE	1.417
Ridge Regression	RMSE	1.417
Lasso Regression	RMSE	1.417
PCR	RMSE	1.417
PLS	RMSE	1.417
Decision Trees	RMSE	1.386
Bagging	RMSE	1.028

Based on the above results, Bagging and Decision Trees are good for predicting how long the employees would stay with the company.

This time I assume that the data is linear. In the future work, I might need to assume the data is non linear. Hence, I need to use non-linear models such as Polynomial regression to see which one gives better prediction accuracy.

Data Source

https://www.kaggle.com/datasets/jacksonchou/hr-data-for-analytics.

Appendix

##library

```{r}
library(dplyr)
library(tidyr)
library(ggplot2)
library(glmnet)
library(pls)
library (e1071)
library(corrplot)
library(tree)
library(ipred)
library(rpart)
library(gam)
library(randomForest)
library(gbm)
```

##importing, renaming, and cleaning data

```{r}
HR=read.csv("HR_comma_sep.csv",header=T)
dim(HR)
str(HR)
```

sum(is.na(HR))#find the number of missing values
drop_na(HR)
HR=rename(HR,department=sales)
HR=HR %>% arrange(desc(department))
str(HR)
```
##Converting data
```{r}
HR$Work_accident=as.factor(HR$Work_accident)
HR$left=as.factor(HR$left)
HR$promotion_last_5years=as.factor(HR$promotion_last_5years)
HR$time_spend_company=as.numeric(HR$time_spend_company)
HR$number_project=as.numeric(HR$number_project)
HR$average_montly_hours=as.numeric(HR$average_montly_hours)
str(HR)
```
##Creating training and testing data
```{r}
train = sample(dim(HR)[1], dim(HR)[1]*0.8)
test=-train
HR.test=HR[test,]
HR.train=HR[train,]
```
##fit the linear model using lest squares on the training set, and report the test error obtained

```{r}
lm.fit=lm(time_spend_company~satisfaction_level+last_evaluation+number_project+average_m
ontly_hours +promotion_last_5years+salary+Work_accident,data=HR.train)
summary(lm.fit)
lm.pred=predict(lm.fit,HR.test)
summary(lm.pred)

```
mean((lm.pred-HR.test$time_spend_company)^2) #test MSE
sqrt(mean((lm.pred-HR.test$time_spend_company)^2)) #RMSE
#our prediction is 1.41446 away from the test value.
```

##Fit a ridge regression model on the training set, with lambda chosen by cross validation. Report the test error obtained.

```{r}
train.mat<-model.matrix(time_spend_company~satisfaction_level+last_evaluation+number_project+average_montly_hours +promotion_last_5years+salary+Work_accident,data=HR.train)
test.mat<-model.matrix(time_spend_company~satisfaction_level+last_evaluation+number_project+average_montly_hours +promotion_last_5years+salary+Work_accident,data=HR.test)
grid<-10^seq(10,-2,length=100)
ridge.fit<-glmnet(train.mat,HR.train$time_spend_company,alpha=0,lambda=grid)
cv.out<-cv.glmnet(train.mat,HR.train$time_spend_company,alpha=0)
bestlam<-cv.out$lambda.min
bestlam
ridge.pred<-predict(ridge.fit,s=bestlam,newx=test.mat)
mean((ridge.pred-HR.test$time_spend_company)^2)
sqrt(mean((ridge.pred-HR.test$time_spend_company)^2))
```

##Fit a lasso regression model on the training set, with lambda chosen by cross validation. Report the test error obtained.

```{r}
train.mat<-model.matrix(time_spend_company~satisfaction_level+last_evaluation+number_project+average_montly_hours +promotion_last_5years+salary+Work_accident,data=HR.train)

test.mat<-model.matrix(time_spend_company~satisfaction_level+last_evaluation+number_project+average_montly_hours +promotion_last_5years+salary+Work_accident,data=HR.test)
```

```
grid<-10^seq(10,-5,length=100)

lasso.fit<-glmnet(train.mat,HR.train$time_spend_company,alpha=1,lambda=grid)
#use cross validation to figure out which lambda value gives smallest MSE.
cv.out<-cv.glmnet(train.mat,HR.train$time_spend_company,alpha=1)
bestlam<-cv.out$lambda.min
plot(cv.out$lambda)
bestlam

lasso.pred<-predict(lasso.fit,s=bestlam,newx=test.mat)
mean((lasso.pred-HR.test$time_spend_company)^2)
sqrt(mean((lasso.pred-HR.test$time_spend_company)^2)) #RMSE
```

##fit a PCR model on the training set, with M chosen by cross validation. Report the test error obtained, along with the value of M selected by cross-validation

```{r}
set.seed(1)
pcr.fit=pcr(time_spend_company~satisfaction_level+last_evaluation+number_project+average_
montly_hours
+promotion_last_5years+salary+Work_accident,data=HR.train,scale=TRUE,validation="CV")
validationplot(pcr.fit,val.type="MSEP")
#the lowest cv error occurs when M=6
pcr.pred=predict(pcr.fit,HR.test,ncomp=6)
mean((pcr.pred-HR.test$time_spend_company)^2)
sqrt(mean((pcr.pred-HR.test$time_spend_company)^2))
```

##fit a PLS model on the training set with M chosen by cross validation. Report the test error obtained, along with the value of M selected by cross validation

```{r}
```

```
set.seed(1)
pls.fit=plsr(time_spend_company~satisfaction_level+last_evaluation+number_project+average_
montly_hours
+promotion_last_5years+salary+Work_accident,data=HR.train,scale=TRUE,validation="CV")
validationplot(pls.fit,val.type="MSEP")
#the lowest cv error occurs when M=4
pls.pred=predict(pls.fit,HR.test,ncomp=4)
mean((pls.pred-HR.test$time_spend_company)^2)
sqrt(mean((pls.pred-HR.test$time_spend_company)^2))
```

##fit a model on the training set using decision trees

```{r}
set.seed(1)
library(tree)
attach(HR)
tree.HR=tree(time_spend_company~satisfaction_level+last_evaluation+number_project+averag
e_montly_hours+promotion_last_5years+salary+Work_accident,subset=train)
plot(tree.HR)
text(tree.HR, pretty=0)
summary(tree.HR)
cv.HR=cv.tree(tree.HR)
plot(cv.HR$size,cv.HR$dev,type='b')
prune.HR=prune.tree(tree.HR,best=6)
plot(prune.HR)
text(prune.HR,pretty=0)
yhat=predict(tree.HR,newdata=HR[-train,])
HR.test=HR[-train,"time_spend_company"]
plot(yhat,HR.test)
abline(0,1)
mean((yhat-HR.test)^2)
```

```
sqrt(mean((yhat-HR.test)^2))
```

##Bagging.

```{r}
set.seed(1)
bag.HR = randomForest(time_spend_company~., data=HR.train, mtry = 9, ntree = 500,
importance=TRUE)
bag.pred = predict(bag.HR, HR.test)
MSE=mean((HR.test$time_spend_company - bag.pred)^2)
RMSE=sqrt(MSE)
RMSE

importance(bag.HR)
```