Final Project

STA3241.01 – April 27, 2020

Luiz Gustavo Fagundes Malpele, Cindy Nguyen, Isabel Zimmerman

Table of Contents

[Introduction 1](#_Toc38887556)

[Dataset 1](#_Toc38887557)

[Data Dictionary 2](#_Toc38887558)

[Exploratory Data Analysis 3](#_Toc38887559)

[Modeling 5](#_Toc38887560)

[Preprocessing 5](#_Toc38887561)

[Best Model 6](#_Toc38887562)

[Random Forest 6](#_Toc38887563)

[Model Comparison 9](#_Toc38887564)

[Other Techniques 10](#_Toc38887565)

[Simple Linear Regression 10](#_Toc38887566)

[PCA 12](#_Toc38887567)

[SVM 14](#_Toc38887568)

[LASSO 15](#_Toc38887569)

# Introduction

The aim of this report was to predict early and mid career pay for college graduates from data collected on independent variables such as: tuition costs, school enrollment size, percentage STEM of majors, percentage of minority students, etc. After a robust exploratory data analysis, various regression algorithms were created utilizing the libraries tidyverse, caret, DataExplorer, fastDummies, leaps, cowplot, and GGally. All of the R code can be found at the GitHub [here](https://github.com/luizmalpele/stats_learning_project/).

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Description** | **Data Type** | |
| name | Institution Name | factor | |
| state\_code | State Abbreviation | factor |
| make\_world\_better\_percent | Percent of alumni who think they are making the world a better place | integer | |
| room\_and\_board | Room and board in USD | integer | |
| ln\_room\_and\_board | Natural Log of Room and board in U$D | double | |
| early\_career\_pay | Estimated early career pay in USD | int | |
| ln\_early\_career\_pay | Natural log of estimated early career pay in USD | double | |
| mid\_career\_pay | Estimated mid career pay in USD | int | |
| ln\_mid\_career\_pay | Natural log of estimated mid career pay in USD | double | |
| total\_enrollment | Total enrollment of students | double | |
| ln\_total\_enrollment | Natural Log of Total enrollment of students | double | |
| out\_of\_state\_tuition | Tuition for out-of-state residents in USD | integer | |
| ln\_out\_of\_state\_tuition | Natural Log of Tuition for out-of-state residents in USD | double | |
| in\_of\_state\_tuition | Tuition for in-of-state residents in USD | integer | |
| ln\_in\_of\_state\_tuition | Natural Log of Tuition for in-of-state residents in USD | double | |
| stem\_percent | Percent of student body in STEM | double | |
| private | Type: 0 for Public, 1 for Private | integer | |
| asian\_ratio | Percentage of Asian Students | double | |
| black\_ratio | Percentage of Black Students | double | |
| minority\_ratio | Percentage of all Minorities Combined | double | |
| hispanic\_ratio | Percentage of Hispanic Students | double | |
| women\_ratio | Percentage of Women Students | double | |
| tuition\_ratio | Out-of-State Tuition and In-State Tuition Ratio | double | |
|  |  |  | |

# Dataset

We began by importing processed data from TidyTuesday, which can also be found [here](https://github.com/luizmalpele/stats_learning_project/blob/master/data/data.Rmd).

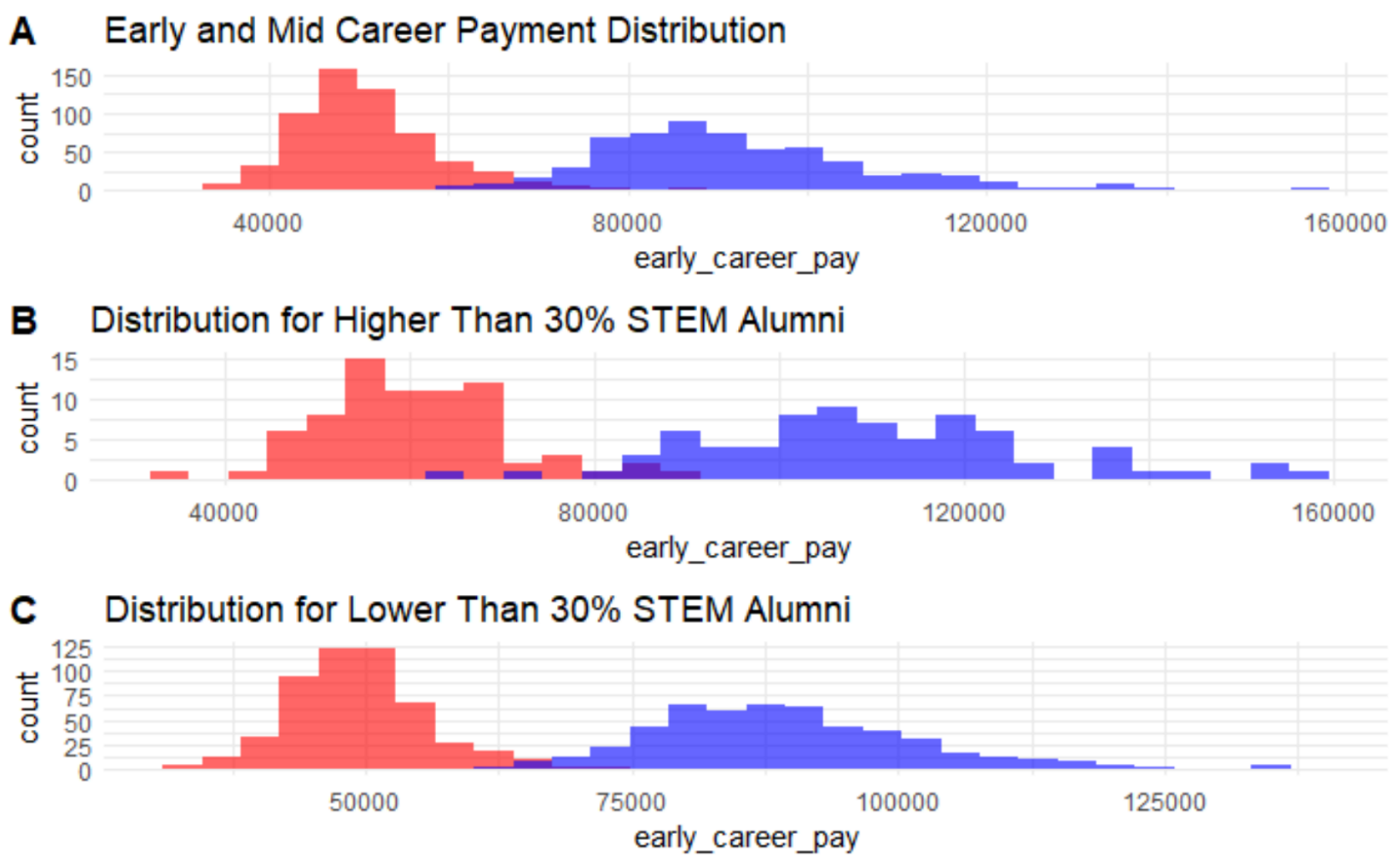
From this data, we transformed all minority variables into percentages of total enrollment and took the log of the following variables: early career pay, mid career pay, in state tuition, out of state tuition, room and board, and total enrollment. This was to create more normal distributions in the data and possibly remove heteroscedasticity. With the addition of these two features in the data, some of the models (particularly linear models) may improve in predictive power.

## Data Dictionary

## Exploratory Data Analysis

The first step was to use the DataExplorer package to automatically create an EDA. This report can be found [here](https://github.com/luizmalpele/stats_learning_project/blob/master/project/EDA_report.html). Using this process was preferred as it automatically created all the univariate distributions and correlation matricies for the variables. This way, we were able to focus on creating more complex explorations that were fine-tuned to the question we wanted to answer.

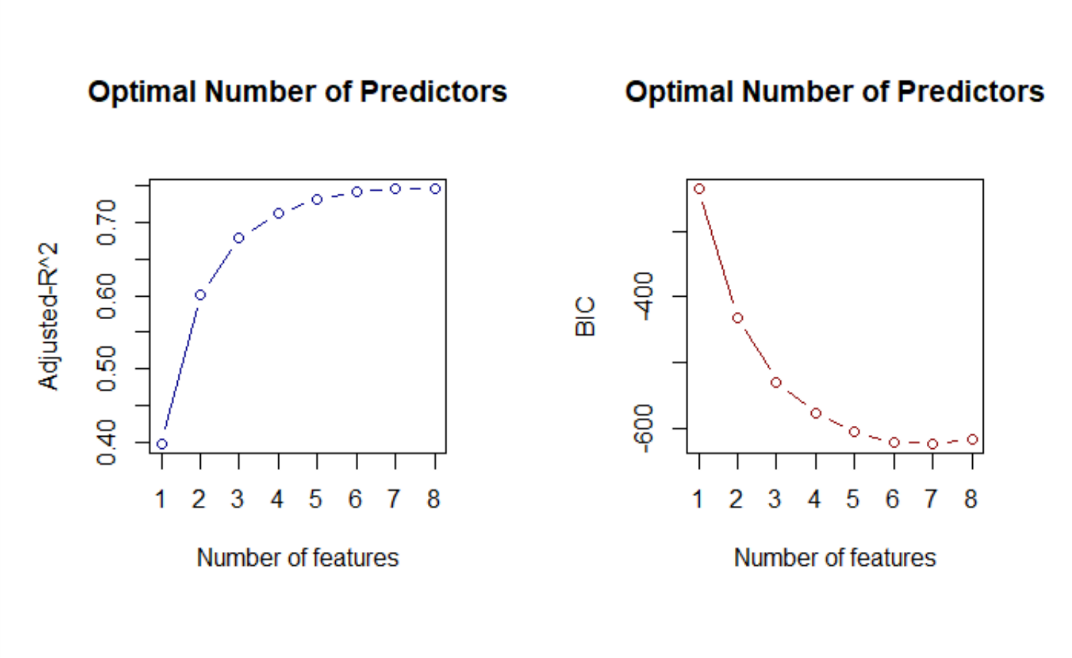
#create\_report(college\_dataset)

The first look into the data was to see how the distribution of pay shifted from early to mid-career. We could tell that the distribution became wider and right-skewed for mid-career pay and was higher on average; the mean early pay was $51,000 whereas the mid-career pay average was $92,000.

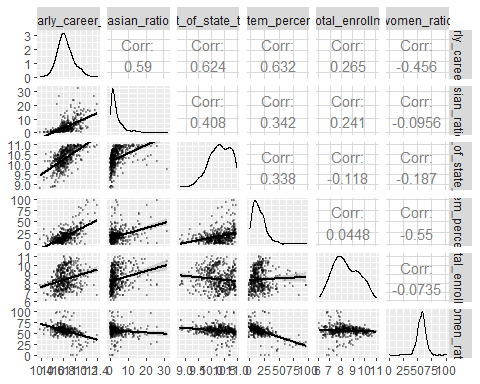
Then, to understand more thoroughly the impact of variables we thought would be higly significant in our regression models. The first variable we chose to explore was *stem\_percent* as STEM majors tend to have higher paid jobs both right out of college and over time. In Figure A, it is observed that both early and mid career pay has relatively normal distributions. However, when observing schools with higher than 30% STEM majors in Figure B, there is no longer a normal distribution; both early and mid career pay are observed to be proportionally hight, but do note that the sample size is much smaller for this visualization. Finally, we see that the less than 30% STEM majors has a relatively similar distribution as the school totals; that is, this distribution is fairly normal.

# Modeling

## Variables Selection



Based on the EDA, BIC, Mallows’ CP, and the , the models will be tested on the following predictors: *ln\_early\_career\_pay, asian\_ratio, ln\_out\_of\_state\_tuition, stem\_percent, ln\_total\_enrollment,* and *women\_ratio*.. More than this will result in overfitting.



We then formed a correlation matrix between these variables; if variables are highly correlated, they can cause standard errors of models to be unreliable and cause poor models in general. From this, we found that the variables were, at most, about 60% correlated. This was not worrisome in itself, but certainly something to keep in mind when evaluating results.

For testing purposes, we created a train control variable in order to establish that each model would be tested with 10-fold cross-validation. This is to ensure that the models are not overfitting in the training phase, and it gives feedback on how well the model is performing. We also split the data so that 80% of aribitrary but specific data is used to train, and the other 20% is used to test the model’s performance. This is also done to avoid overfitting, and it is preferable to perform the final model selection with an out of sample criterion.

## Best Model

### Random Forest

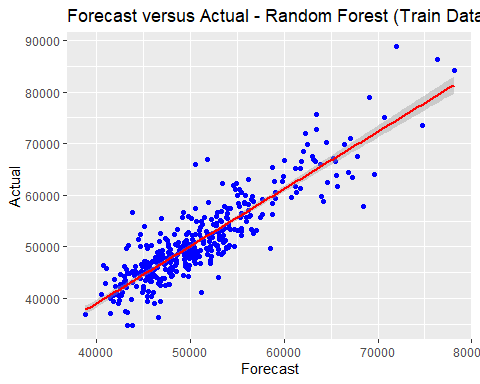
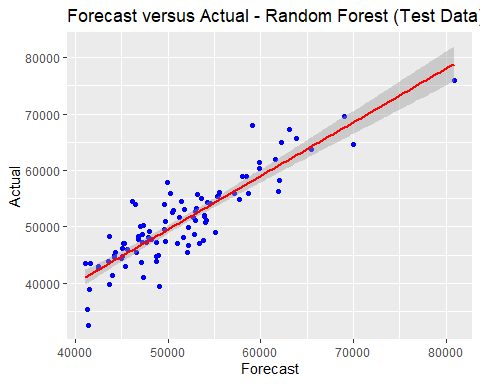
We chose this because

oob <- trainControl(method = "oob")  
cv\_5 <- trainControl(method = "cv", number = 5)  
rf\_grid <- expand.grid(mtry = 1:10)  
  
set.seed(825)  
rf\_model <- train(ln\_early\_career\_pay ~ ., data = train\_data,  
 method = "rf",  
 trControl = oob,  
 verbose = FALSE,  
 tuneGrid = rf\_grid)

## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:  
## reset to within valid range  
  
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:  
## reset to within valid range  
  
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:  
## reset to within valid range  
  
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:  
## reset to within valid range  
  
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid mtry:  
## reset to within valid range

# print results  
rf\_model

## Random Forest   
##   
## 391 samples  
## 5 predictor  
##   
## No pre-processing  
## Resampling results across tuning parameters:  
##   
## mtry RMSE Rsquared   
## 1 0.07381380 0.7566485  
## 2 0.07268202 0.7640539  
## 3 0.07326102 0.7602798  
## 4 0.07393627 0.7558403  
## 5 0.07389827 0.7560913  
## 6 0.07456265 0.7516859  
## 7 0.07485997 0.7497016  
## 8 0.07472893 0.7505771  
## 9 0.07486140 0.7496920  
## 10 0.07432065 0.7532951  
##   
## RMSE was used to select the optimal model using the smallest value.  
## The final value used for the model was mtry = 2.



### 

### Model Comparison

For brevity, the model selection results are posted below. We chose to highlight the random forest model as it has the highest and the lowest in and out of sample RMSE. In order to see our analysis of the other models, see below in the *Other Techniques* section. Furthermore, the random forest model can be displayed as a decision tree, and it is easy to interpret by people out of the Data Science field since it mirrors the human decision-making process.

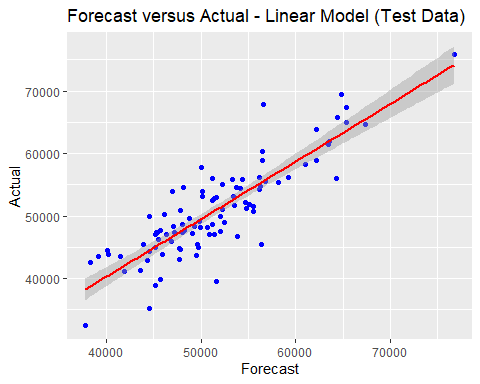
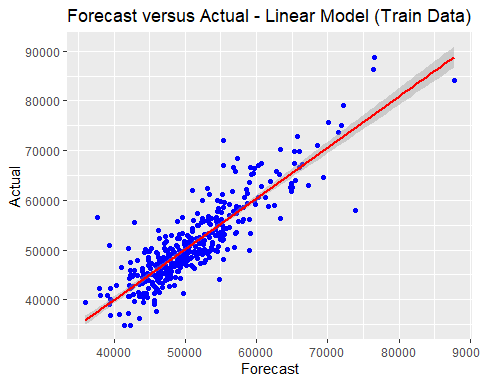
|  |  |  |  |
| --- | --- | --- | --- |
| Predictive Model | R-Squared | In Sample RMSE | Out of Sample RMSE |
| *Ordinary Least Squares* | 0.7382 | 0.07665 | 0.08076483 |
| *Ordinary Least Squares-glmnet* | 0.07715067 | 0.7468707 | 0.08027886 |
| *Random Forest* | 0.7640539 | 0.07268202 | 0.0727503 |
| *Principal Component Analysis* | 0.7417742 | 0.07682088 | 0.08076483 |
| *Support Vector Machine* | 0.7367368 | 0.07692738 | 0.07923666 |

## Other Techniques

### Ordinary Least Squares

earlypay\_lm <- lm(ln\_early\_career\_pay ~ .,   
 data = train\_data)  
summary(earlypay\_lm)

##   
## Call:  
## lm(formula = ln\_early\_career\_pay ~ ., data = train\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.24361 -0.04826 -0.00374 0.04307 0.40764   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 9.1334438 0.1200992 76.049 < 2e-16 \*\*\*  
## asian\_ratio 0.0091614 0.0010750 8.522 3.60e-16 \*\*\*  
## ln\_out\_of\_state\_tuition 0.1425788 0.0102636 13.892 < 2e-16 \*\*\*  
## stem\_percent 0.0029324 0.0003481 8.423 7.37e-16 \*\*\*  
## ln\_total\_enrollment 0.0308594 0.0038493 8.017 1.31e-14 \*\*\*  
## women\_ratio -0.0020555 0.0003963 -5.187 3.46e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.07665 on 385 degrees of freedom  
## Multiple R-squared: 0.7416, Adjusted R-squared: 0.7382   
## F-statistic: 221 on 5 and 385 DF, p-value: < 2.2e-16

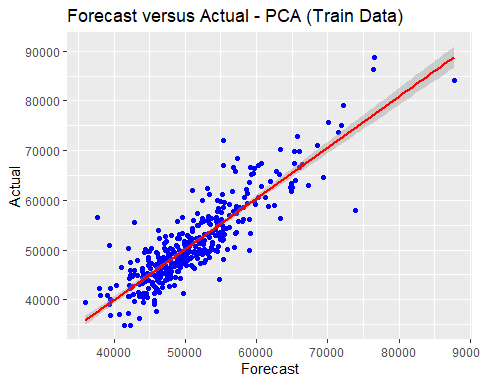
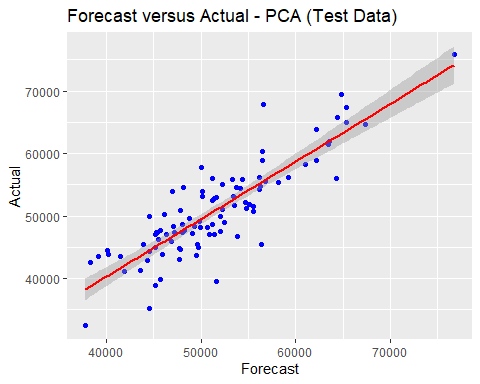
This **Ordinary Least Squares** linear model is focused on the variable, *ln\_early\_career\_pay*, and is being tested with seven other variables that were previously selected by the previous methods. The Adjusted- is 0.7487 and all predictors are statistically significant to the analysis.

The Root Mean Square Error of the out sample prediction was calculated by utilizing the testing set of the mean of the following difference squared: , also know as RMSE, the result was 0.005810831. This is another linear graph that shows a comparison of the Actual and Forecast values, but only the test set or 20% of the data was used. Again, the data is mostly surrounded around the (50,000 , 50,000) mark.

### PCA

glm\_pca\_model <- train(ln\_early\_career\_pay ~ . ,   
 data = train\_data,   
 method = "glm",   
 preProcess = "pca",   
 trControl = train\_control)  
glm\_pca\_model

## Generalized Linear Model   
##   
## 391 samples  
## 5 predictor  
##   
## Pre-processing: principal component signal extraction (5), centered (5),  
## scaled (5)   
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 352, 352, 351, 352, 351, 352, ...   
## Resampling results:  
##   
## RMSE Rsquared MAE   
## 0.07753735 0.7365653 0.0580205

Principal Component Analysis, or PCA, is a type of linear transformation that allows you to visualize the overall format of the dataset. In a way, PCA “tilts” the dataset to be one dimensional. This will depend on the number of variables and will help to understand what variables are similar to each other and which are different. We utilized PCA to reduce the dimensionality of our dataset to make it easier to work with. In the linear model above, we have 391 samples with 8 predictors. The *Rsquared* value of 0.7469513 tells us that the model that we are running is fitting the actual data by 74.7%. It is ideal for *RMSE* values to be as small as possible, or as close to zero on a zero to one scale. The *RMSE* is 0.075609.

The PCA model shows the Forecast data being tested against Actual data. This training data is very similar to the training model for the Linear Model. This could mean that PCA simplified the dataset to the point of it being

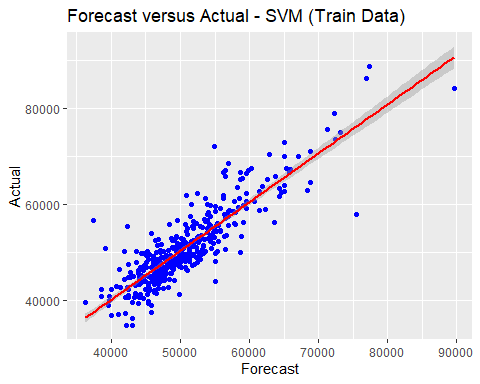
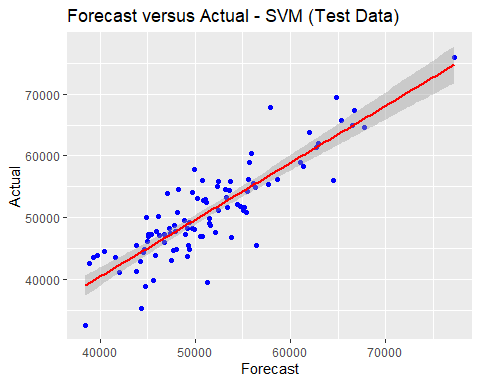
### Support Vector Machine

Preprocessing

Model

# model 1:  
svm\_model\_1 <- train(ln\_early\_career\_pay ~ .,  
 data = train\_data,   
 method = "svmLinear",  
 tuneGrid = tGrid,   
 trControl = tr\_control,   
 metric = "RMSE",  
 preProcess = c("center", "scale")  
)  
svm\_model\_1

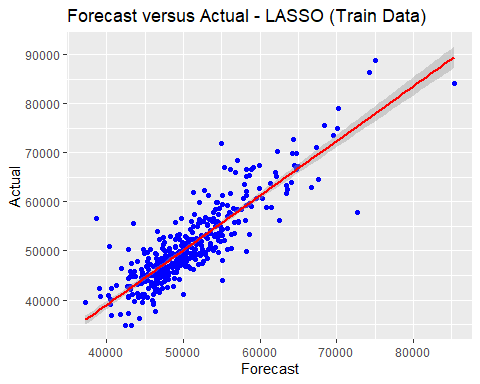
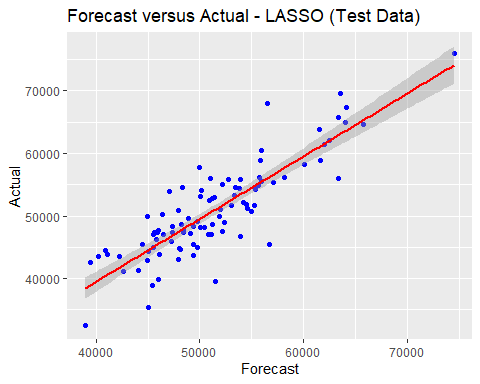
## Support Vector Machines with Linear Kernel   
##   
## 391 samples  
## 5 predictor  
##   
## Pre-processing: centered (5), scaled (5)   
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 352, 352, 352, 351, 353, 351, ...   
## Resampling results across tuning parameters:  
##   
## C RMSE Rsquared MAE   
## 0.01 0.07774523 0.7364486 0.05804287  
## 0.05 0.07692738 0.7367368 0.05746174  
## 0.10 0.07698578 0.7367897 0.05749150  
## 0.25 0.07695550 0.7367911 0.05742800  
## 0.50 0.07699114 0.7366774 0.05745745  
## 0.75 0.07701765 0.7364923 0.05746669  
## 1.00 0.07698763 0.7366353 0.05746556  
## 1.25 0.07704349 0.7365145 0.05749835  
## 1.50 0.07705455 0.7364748 0.05751070  
## 1.75 0.07705137 0.7364952 0.05750378  
## 2.00 0.07705310 0.7365205 0.05751559  
## 5.00 0.07707319 0.7363771 0.05752366  
##   
## RMSE was used to select the optimal model using the smallest value.  
## The final value used for the model was C = 0.05.



### Ordinary Least Squares with Glmnet (LASSO/Ridge)

set.seed(981)  
#10 fold CV  
train\_control <- trainControl(method = "cv", number = 10)  
#Grid  
grid <- seq(-2,10,length=100)  
  
lasso\_model <- train(ln\_early\_career\_pay ~ .,  
 data = train\_data,   
 method = "glmnet",   
 trControl = train\_control,  
 metric = "Rsquared",  
 tune\_Grid = expand.grid(alpha = 1, lambda = grid))  
lasso\_model

## glmnet   
##   
## 391 samples  
## 5 predictor  
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 353, 351, 353, 351, 353, 352, ...   
## Resampling results across tuning parameters:  
##   
## alpha lambda RMSE Rsquared MAE   
## 0.10 0.0001917911 0.07651221 0.7463793 0.05778304  
## 0.10 0.0019179111 0.07650888 0.7464208 0.05779062  
## 0.10 0.0191791107 0.07715067 0.7468707 0.05841034  
## 0.55 0.0001917911 0.07652939 0.7462609 0.05779879  
## 0.55 0.0019179111 0.07657131 0.7461840 0.05786946  
## 0.55 0.0191791107 0.07956271 0.7420766 0.06032027  
## 1.00 0.0001917911 0.07653376 0.7462049 0.05780614  
## 1.00 0.0019179111 0.07665041 0.7459095 0.05795561  
## 1.00 0.0191791107 0.08373134 0.7296577 0.06410784  
##   
## Rsquared was used to select the optimal model using the largest value.  
## The final values used for the model were alpha = 0.1 and lambda = 0.01917911.

This **Ordinary Least Squares with LASSO penalization** linear model contains the seven variables previously used in the linear model tested against the *ln\_early\_career\_pay*.The best LASSO model has a and . The highest is 0.7527662. LASSO increases the variance explained for the predictive model, but it also has a small penalty increasing the bias.

When the *test set* was used for an out of sample prediction, it is clear that the regression line for the Forecast versus Actual values presents a bettet result when compared to the simple OLS model. Observation fall closer to the line and the Out of Sample RMSE is 0.62077, which does not represent a significant increase in bias, when compared to gain on explanatory power when the LASSO penalization was used.