Final Project Report

Introduction

In this project, my goal was to take my dataset, which was Korean Baseball Organization Pitching data, and look for how the variables in the data corresponded to my chosen target variable, which was win_loss_percentage. I chose this dataset because I was interested in finding out which pitching statistics coincide with the best winning percentage, as well as vice versa. The main problem I am trying to look into is which statistics are actually important to overall team success, and whether some of the statistics that teams and the public consider to be important to team success are actually important and vital to achieving a winning season. Some existing approaches I chose to use for my project include Linear Regression, Cross Validation, Elastic Net, and Decision trees.

Data collection and preparation

I got my dataset from Kaggle, and the dataset itself contains 34 variables. The data was collected from 20 years of KBO play, and had over 300 observations. There was some preprocessing and data cleaning needed before I could start coding, and I removed all NA's, and I removed a few variables that were not necessary and were not quantifiable, namely Team and ID.

```
> kbo_colnames<-colnames(kbo)
> kbo_colnames
 [1] "id"                 "year"               "team"
 [4] "average_age"        "runs_per_game"      "wins"
 [7] "losses"             "win_loss_percentage" "ERA"
[10] "run_average_9"      "games"              "games_started"
[13] "games_finished"     "complete_game"      "shutouts"
[16] "saves"              "innings_pitched"    "hits"
[19] "runs"               "earned_runs"        "home_runs"
[22] "walks"              "intentional_walks"  "strikeouts"
[25] "hit_batter"         "balks"              "wild_pitches"
[28] "batters_faced"      "WHIP"               "hits_9"
[31] "homeruns_9"         "walks_9"            "strikeouts_9"
[34] "strikeout_walk"
>
```

These are my variables before team and ID are removed.

Model Discussion/Methods

I used many different analysis methods and techniques to visualize the problem and give insight to what we are trying to find. I used scatter plots, feature plots, and matrices to help visualize the data. The methods below were used to go in depth and really study the data to get results.

Joseph Izzo
STA 4241

Linear Regression:

The first model I used was linear regression. I wanted to use linear regression to analyze the significance of my variables with win_loss_percentage as my target variable. I thought linear regression would be one of the most straightforward and best ways to visualize and interpret the usability and importance of the variables in the set. It allowed me to look at which variables were significant in predicting winning percentage, and that was my ultimate goal of this project.

Elastic Net:

The second model I used was Elastic Net. I used Elastic Net because I wanted to use another regression model that avoided some of the shortcomings associated with lasso and ridge regression, as it is good with regularization of models. It would help me to find the best overall model and a subset of variables that would make the best model possible. Also the hyperparameters of elastic net allows you to change as you see fit and pick the best option.

Decision Trees:

The third model/method I used was decision trees. Using decision trees allowed me to visualize a breakdown of what would cause different outcomes of my target variable, which is winning percentage. I wanted to see how the model decided which variables, aside from wins and losses, were significant in predicting winning percentage and I wanted to see it mapped out. Decision trees are also good for prediction, and I wanted to get a good idea of how accurate this model would have been against the regression and elastic net models that were also run throughout my project.

Results

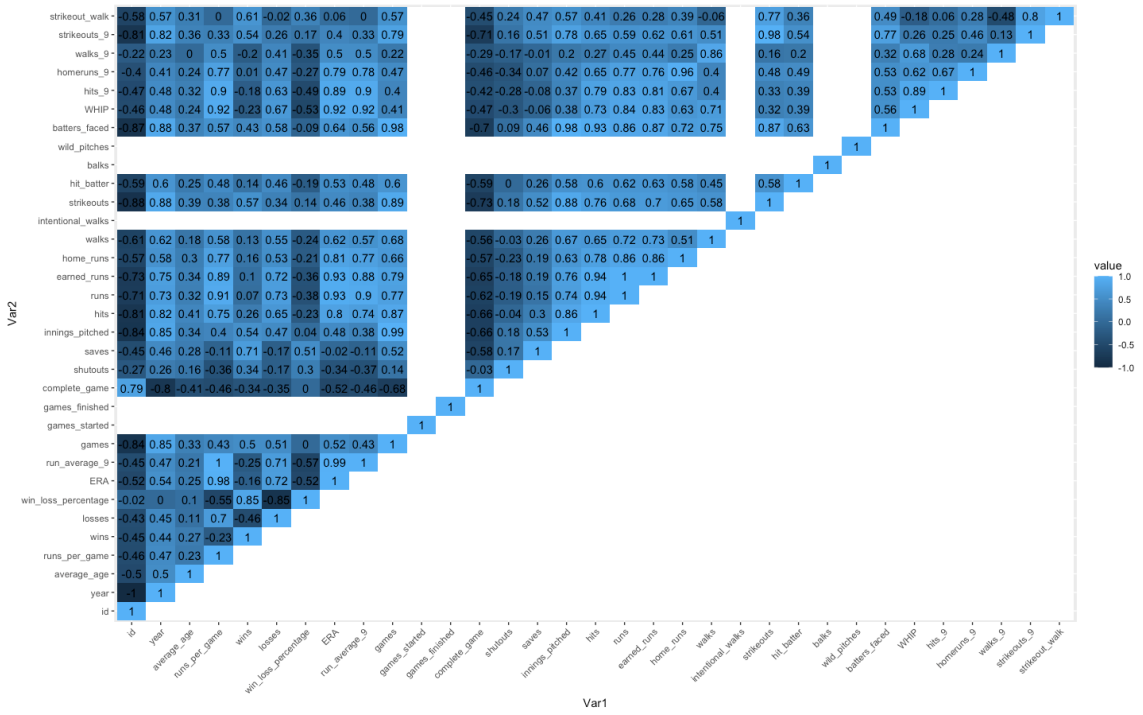


Figure 1. This is a correlation matrix of all 34 of my variables.

Linear Regression Results:

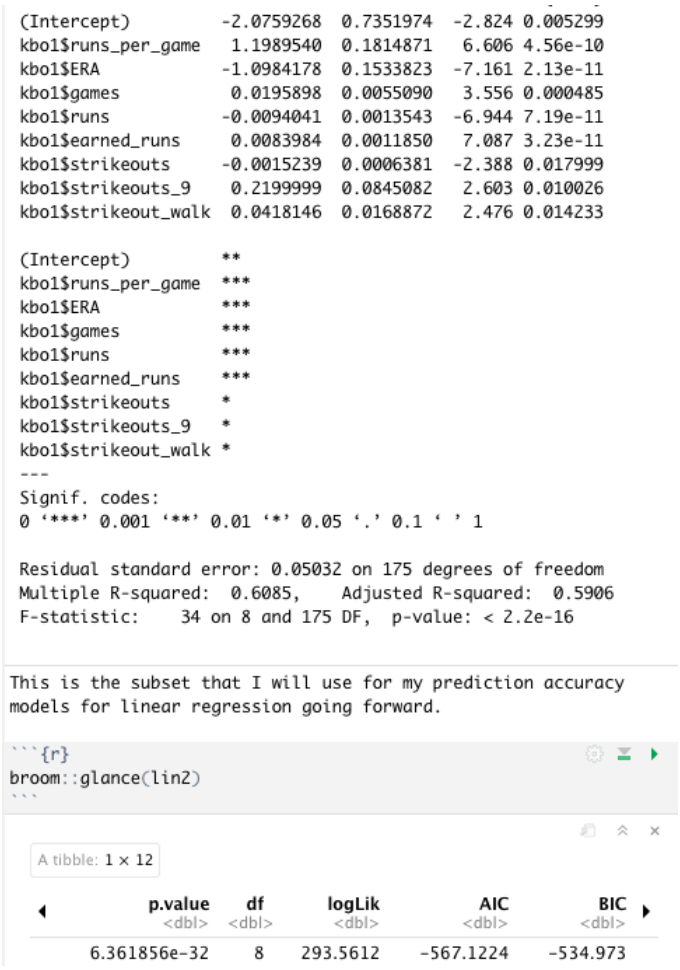
For my linear regression models, I started with one model with every variable aside from ID and team. I then fit a second model that had the significant variables from the first regression, except wins and losses, as I removed them from this model because of how highly correlated to winning percentage they are. I reported on the AIC, BIC, RMSE, and adjusted r squared.

```
Coefficients: (2 not defined because of singularities)
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       7.761e-01  2.842e-01   2.731  0.00746 **
year              5.636e-05  1.280e-04   0.440  0.66076
average_age       1.010e-04  3.444e-04   0.293  0.76999
runs_per_game    -1.714e-01  6.059e-02  -2.828  0.00565 **
wins              3.644e-03  1.601e-04  22.760  < 2e-16 ***
losses           -3.897e-03  1.659e-04 -23.483  < 2e-16 ***
ERA               1.265e-01  6.549e-02   1.931  0.05627 .
run_average_9     9.460e-04  7.375e-02   0.013  0.98979
games            -5.476e-03  2.079e-03  -2.634  0.00976 **
games_started            NA         NA      NA       NA
games_finished   -3.674e-05  1.797e-04  -0.205  0.83836
complete_game            NA         NA      NA       NA
shutouts          3.801e-05  1.371e-04   0.277  0.78214
saves            -7.925e-05  7.506e-05  -1.056  0.29355
innings_pitched   3.412e-04  2.722e-04   1.253  0.21304
hits              1.360e-04  1.737e-04   0.783  0.43529
runs              1.307e-03  4.870e-04   2.684  0.00850 **
earned_runs      -9.912e-04  4.851e-04  -2.043  0.04361 *
home_runs         6.918e-05  9.704e-05   0.713  0.47755
walks             2.504e-04  1.663e-04   1.506  0.13525
intentional_walks -4.947e-05  4.332e-05  -1.142  0.25621
strikeouts       -1.385e-04  6.840e-05  -2.024  0.04559 *
hit_batter       -4.559e-05  2.878e-05  -1.584  0.11629
balks             1.028e-05  1.761e-04   0.058  0.95358
wild_pitches     -2.643e-05  3.404e-05  -0.776  0.43928
batters_faced    -1.865e-05  2.068e-05  -0.902  0.36939
WHIP             -2.402e-01  1.828e-01  -1.314  0.19183
hits_9            1.065e-02  1.114e-02   0.956  0.34137
homeruns_9       -9.589e-03  1.256e-02  -0.763  0.44712
walks_9           2.300e-03  1.190e-02   0.193  0.84711
strikeouts_9      1.582e-02  9.002e-03   1.757  0.08187 .
strikeout_walk    1.409e-02  8.284e-03   1.701  0.09205 .
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.003694 on 101 degrees of freedom
Multiple R-squared:  0.9984,    Adjusted R-squared:  0.9979
```

Figure 2. This is my first linear regression model. We can see from this output that the adjusted r squared is extremely high, and I believe that is due to the inclusion of wins and losses. The AIC and BIC of this model was –1067.79 and –978.659 respectively. All of these results were done using the 70/30 train test split that I used. The RMSE was equal to 0.1064045.

My next model has wins and losses removed.



```
(Intercept)         -2.0759268  0.7351974  -2.824 0.005299
kbo1$runs_per_game   1.1989540  0.1814871   6.606 4.56e-10
kbo1$ERA            -1.0984178  0.1533823  -7.161 2.13e-11
kbo1$games           0.0195898  0.0055090   3.556 0.000485
kbo1$runs           -0.0094041  0.0013543  -6.944 7.19e-11
kbo1$earned_runs     0.0083984  0.0011850   7.087 3.23e-11
kbo1$strikeouts     -0.0015239  0.0006381  -2.388 0.017999
kbo1$strikeouts_9    0.2199999  0.0845082   2.603 0.010026
kbo1$strikeout_walk  0.0418146  0.0168872   2.476 0.014233

(Intercept)         **
kbo1$runs_per_game  ***
kbo1$ERA            ***
kbo1$games          ***
kbo1$runs           ***
kbo1$earned_runs    ***
kbo1$strikeouts     *
kbo1$strikeouts_9   *
kbo1$strikeout_walk *
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.05032 on 175 degrees of freedom
Multiple R-squared:  0.6085,    Adjusted R-squared:  0.5906
F-statistic:    34 on 8 and 175 DF,  p-value: < 2.2e-16
```

This is the subset that I will use for my prediction accuracy models for linear regression going forward.

```{r}
broom::glance(lin2)
```

A tibble: 1 × 12

p.value	df	logLik	AIC	BIC
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
6.361856e-32	8	293.5612	-567.1224	-534.973

Figure 3. This is my second regression model. This model has all my significant variables from the first regression model put together, and I omitted wins and losses. We can see that the adjusted r squared is 0.5906, and the AIC and BIC are equal to –567.1224 and –534.973 respectively. The RMSE is equal to 0.0954948. This RMSE was actually lower than the first regression model.

Elastic Net Results:

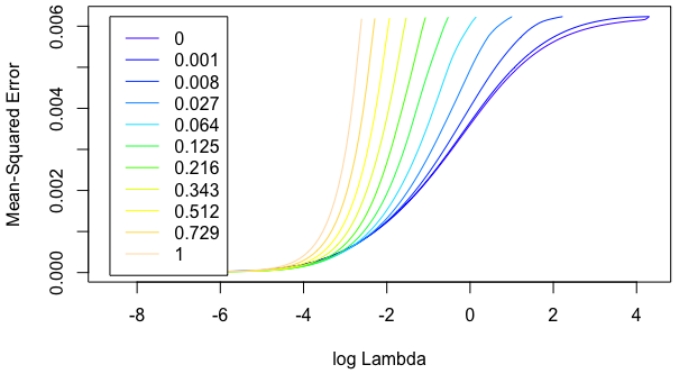


Figure 4. This is the elastic net plot. The MSE values are small, but that is expected with my target variable.

```
glmnet

131 samples
 31 predictor

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 103, 105, 106, 105, 105
Resampling results across tuning parameters:

  alpha  lambda       RMSE         Rsquared   MAE
  0.10   0.000151846  0.004744153  0.9967512  0.003782998
  0.10   0.001518460  0.004808041  0.9966989  0.003889458
  0.10   0.015184601  0.010751605  0.9909012  0.008557146
  0.55   0.000151846  0.004678965  0.9968451  0.003699492
  0.55   0.001518460  0.004709596  0.9970620  0.003727933
  0.55   0.015184601  0.012889561  0.9970552  0.010336294
  1.00   0.000151846  0.004536085  0.9970964  0.003584010
  1.00   0.001518460  0.004797511  0.9970549  0.003802466
  1.00   0.015184601  0.016700068  0.9970492  0.013608401

RMSE was used to select the optimal model using
 the smallest value.
The final values used for the model were alpha = 1
 and lambda = 0.000151846.
```

Figure 5. This is the output from my elastic net function, we can see that the r squared values are all quite high, similar to what we see in figure 2. The lambda used was 0.0001, and the RMSE value is very small, and is actually smaller than the regression models at a value of 0.00453. I used a get best result

function in r studio, which can be seen in the appendix, and I got that RMSE value. Using the test set, the RMSE and r squared values were very similar, with the training set being barely better, but not enough to make a differentiation. So far, elastic net appears to be the best model. We will see how decision trees compare to both elastic net and regression.

Decision Trees Results:

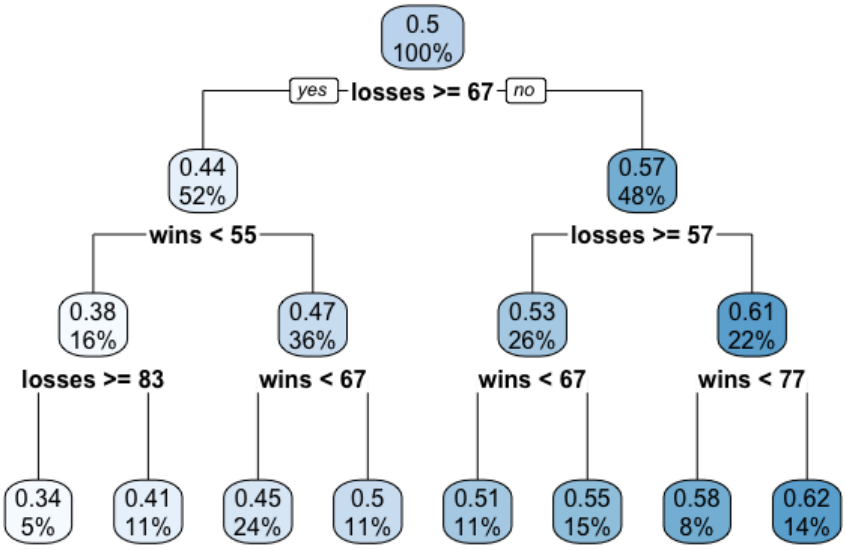


Figure 6. This is my first tree, this one has wins and losses included, and as we see here, those are the only 2 variables used in the model.

```
Regression tree:
tree(formula = win_loss_percentage ~ ., data = train)
Variables actually used in tree construction:
[1] "losses" "wins"
Number of terminal nodes:  8
Residual mean deviance:  0.0003446 = 0.04239 / 123
Distribution of residuals:
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-0.071220 -0.010950  0.002296  0.000000  0.009500  0.042930
```

Figure 7. This is the summary of the first tree, we can see that the residual mean is 0.0003.

The pruned decision tree was the exact same as the unpruned tree, which I thought was interesting. I believe that is because of the inclusion of wins and losses in the model.

```r
kbo_prune_pred = predict(kbo_tree_prune, newdata = train)
rmse(kbo_prune_pred, train$win_loss_percentage)
```

```
[1] 0.02000417
```

Above is the training RMSE.

Test RMSE:

```r
kbo_prune_test_pred = predict(kbo_tree_prune, newdata = test)
rmse(kbo_prune_test_pred, test$win_loss_percentage)
```

```
[1] 0.02523025
```

This is out test RMSE.

Figure 8. We can see in the picture above that there is not much difference between the train RMSE and the test RMSE for the pruned tree. The RMSE is still not as low as elastic net. Also, using CV I could tell that 8 nodes was the best size for the tree.

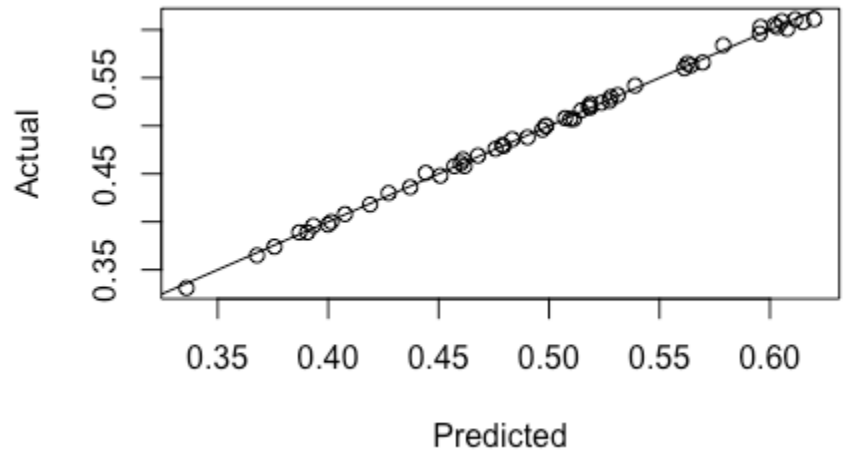


Figure 9. We can see on the above plot that I plotted the predicted vs actual values plot using the predict function and a linear regression model, and we can see above that the points fit the actual values very well.

Discussion of Data Analysis:

After seeing all of the plots, summaries, and statistics above we can get some good interpretations now. All of the models seemed to return very similar adjusted r squared values, as most of them were in the 0.9-0.98 range. When it comes to the RMSE, elastic net seemed to

minimize the error the best as it had a RMSE of 0.004, compared to linear regression which had a RMSE of 0.09, and decision trees which had a RMSE of 0.02. The test and training RMSE's as well as the AIC, BIC, and adjusted r squared were all very similar to one another. The training RMSE on the decision tree regression was slightly lower than the testing RMSE. Overall, the most important predictors of winning percentage I found throughout this project were ERA, wins, losses, strikeouts per walk, runs, runs per game, earned runs, strikeouts, and strikeouts per nine innings. Overall, the elastic net model appears to be the most well suited to reduce error in predicting winning percentage.

References

https://www.kaggle.com/code/natnif/korean-baseball-pitching-data-analysis

https://daviddalpiaz.github.io/r4sl/modeling-basics-in-r.html#visualization-for-regression