Predicting Minnesota Wild Wins and Losses

Henry Smith

Introduction

Being the state of Hockey, and in light of the Wild's playoff run, I have decided to use what we have learned in ADM to predict our hockey team's wins and losses. Additionally, this topic is not only fitting because of the time of year and our location, but also the challenge that it brings. Hockey is widely considered to be one of the most random sports in popular media. Individual wins and losses are subject to much more luck than say a tennis match where the result is much more predictable. This made me wonder, is it possible to accurately predict wins and losses of the Wild's 2023 season based on previous seasons? In order to make this fair, I will not be able to use the statistics goals for and goals against at the same time as it would clearly defeat the purpose of this analysis. My goal is simple, to find game statistical indicators that help predict whether the Wild will win or lose a game. As a threshold, I will attempt to get an accuracy value greater than 95% in order to officially call this project a success.

To start let's look at the variables we will be working with

- PDO This variable represents the total save percentage plus the total shooting percentage
- GF Goals for the Wild
- oSZ% Offensive zone start percentage
- CF% Corsi For % at Even Strength

In order to understand these variables place in hockey and this project, it is right to explain their details a little more. The PDO is relatively straightforward. It is calculated by adding the save percentage (the amount of saves the Wild goalie makes divided by the total number of shots the opponent takes) to the shooting percentage (the amount of goals the Wild score divided by the total shots the Wild take). This should be a good indicator of wins/losses since it considers both offense and defense of the Wild which are both important in the result of a game. GF is exactly what it sounds like, but oZS% is a little more complicated. The offensive zone is the zone on the hockey rink that is past the blue line on the opposite end from the where the Wild's goalie is. Essentially it is the area that is close to the opponents goal. Likewise, the defensive zone is the ares that is close to where the Wild's goalie is. In hockey, sometimes the play stops because of fouls and the puck needs to be dropped somewhere where two players from each team can face off to win the puck. This is a face off. This statistic is calculated by taking the total offensive zone face offs divided by the offensive zone face offs plus the defensive zone face offs. This is a decent indicator of how often the puck was close to the other teams net. CF% has a confusing name. This is just the total shot plus blocks plus misses of your team divided by the shots plus blocks plus misses of your team plus the shots plus blocks plus misses of the opponent. A value greater than 50% means that the team was controlling the puck more often than not.

Data Gatering and Cleaning

I gathered my data from the past five seasons including this season from hockey-reference.com. Each season has one row per game, and stats for each game across the columns. Once I imported each separate season in

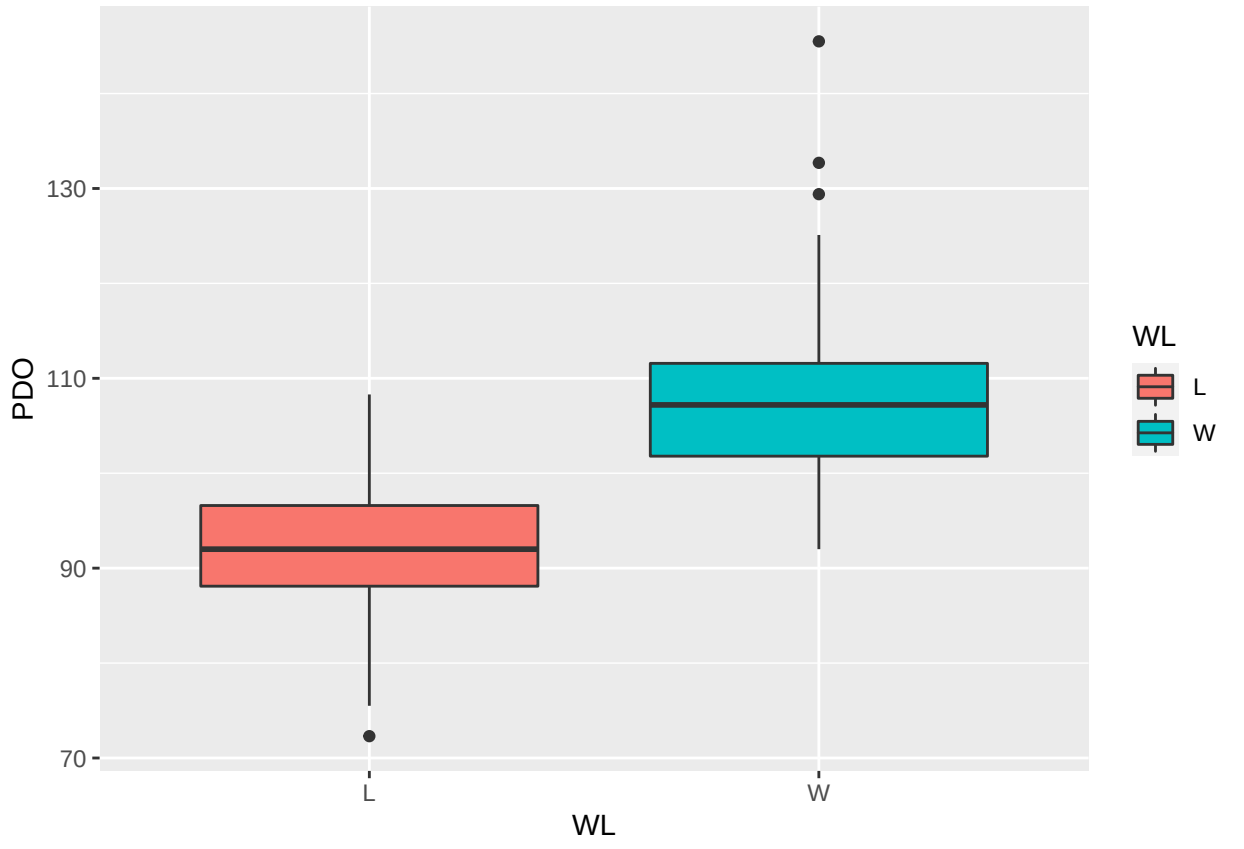
r while simultaneously changing the column types, I deleted useless rows and columns, and renamed them appropriately. Once I had done this for each season, I used this season (2023 season) as my testing data set since I am predicting wins and losses from this season, and combined all the previous 4 seasons to be my training dataset.

```
wild_test <- wild23[-(1),]
wild22 <- wild22[-(1),]
wild21 <- wild21[-(1),]
wild20 <- wild20[-(1),]
wild19 <- wild19[-(1),]


test <- rbind(wild22, wild21)
test2 <- rbind(wild20,test)
wild_train <- rbind(wild19,test2)
```

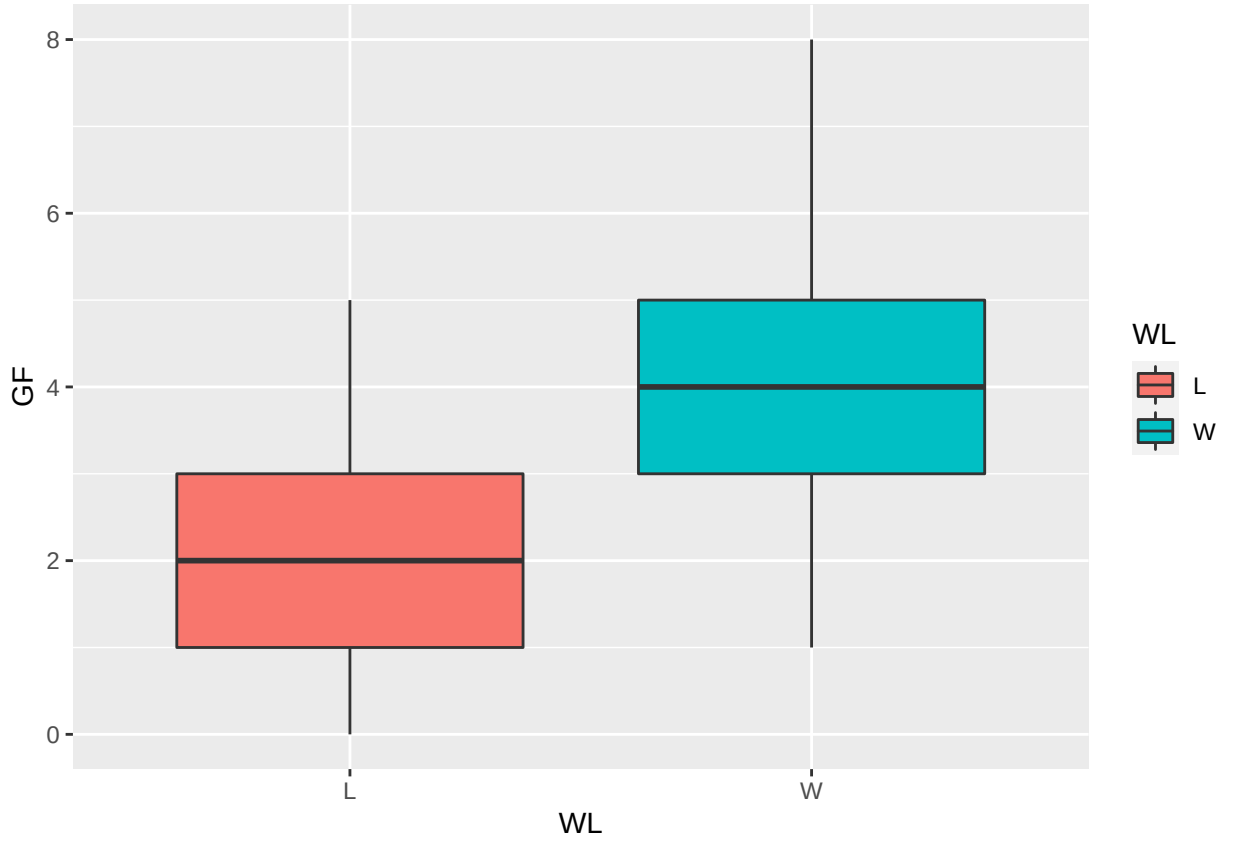
EDA

Since we have set up our data, it is now fitting to see what relationships exist within our data between statistics and wins and losses. This is a necessary step before modelling as it will indicate which variables are likely to have the most impact in prediction and will also save time in the modeling process.

```
ggplot(wild_train, aes(x=WL, y=PDO, fill=WL)) +
  geom_boxplot()
```



```
ggplot(wild_train, aes(x=WL, y=GF, fill=WL)) +
  geom_boxplot()
```



After trying many different versions of EDA across all the variables, I came to find that only these two variables, PDO and GF, separated wins from losses significantly. From the boxplots above, we can see that each of these variables is crucial in determining wins and losses for the Wild and most likely hockey teams in general. For PDO, it looks as though a value of around 100 and above will be a win, and anything below is more likely to be a loss. Between GF, goals of around 3 appear to determine a win or a loss. A game with over 3 goals is more likely to be a winning game and a game with under 3 goals is more likely to be a losing game. I now have some good indicators of wins and losses with distinct thresholds for separating the two categories. First, lets plug just these variables into a logistic model to see how accurate they can be.

Models

```
logit_model2 <- logistic_reg() %>%
  set_engine("glm") %>%
  set_mode("classification")

wild_recipe2 <-
  recipe(WL ~ PDO+GF, data=wild_train)

logit_wflow2 <- workflow() %>%
  add_recipe(wild_recipe2) %>%
```

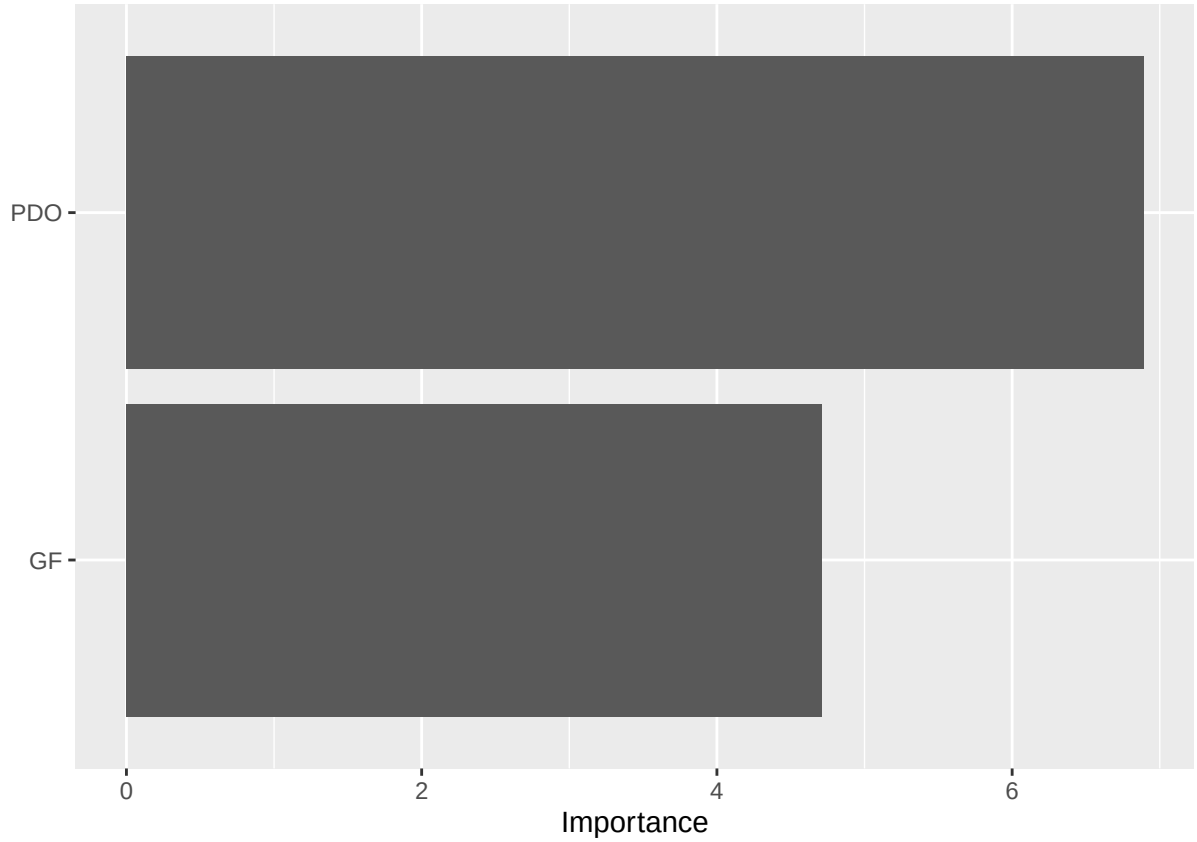
```
  add_model(logit_model2)

logit_fit2 <- fit(logit_wflow2, wild_train)

augment(logit_fit2, wild_test) %>%
  accuracy(truth = WL, estimate = .pred_class)
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.852
```

```
augment(logit_fit2, wild_test) %>%
  conf_mat(truth = WL, estimate = .pred_class)
```

```
##           Truth
## Prediction  L  W
##          L 30  7
##          W  5 39
```

```
extract_fit_parsnip(logit_fit2) %>%
  vip()
```



Logistic classification has gave me a 85% accuracy rate. A promising start with only two variables if my goal is to reach 95% accuracy. This mdel missclassified 2 more losses than wins. Furthermore, PDO was shown to

be the more important variable in this setting. I don't see this as a surprise since PDO had a greater divide between wins and losses in its boxplot than GF did. I suspect that PDO will remain the most important variable for the following models and GF to be second. However, even though there is some good predicting already, I am worried about how effective the other variables will be in predicting wins and losses because their eda plots were not convincing. Let's use a lasso model to determine what the best model is.

```
#lasso model

wild_model <-
  logistic_reg(mixture = 1, penalty=tune()) %>%
  set_mode("classification") %>%
  set_engine("glmnet")

wild_recipe <-
  recipe(formula = WL ~ ., data = wild_train) %>%
  step_zv(home) %>%
  step_normalize(all_predictors())

wild_wf <- workflow() %>%
  add_recipe(wild_recipe) %>%
  add_model(wild_model)

set.seed(1234)
wild_fold_5 <- vfold_cv(wild_train, v = 5)

penalty_grid <-
  grid_regular(penalty(range = c(-2, 5)), levels = 20)

tune_res <- tune_grid(
  wild_wf,
  resamples = wild_fold_5,
  grid = penalty_grid
)

best_penalty <- select_best(tune_res, metric = "accuracy")

wild_final_wf <- finalize_workflow(wild_wf, best_penalty)
wild_final_fit <- fit(wild_final_wf, data = wild_train)

augment(wild_final_fit, new_data = wild_test) %>%
  conf_mat(truth = WL, estimate = .pred_class)
```

```
##           Truth
## Prediction  L  W
##          L 31  6
##          W  4 40
```

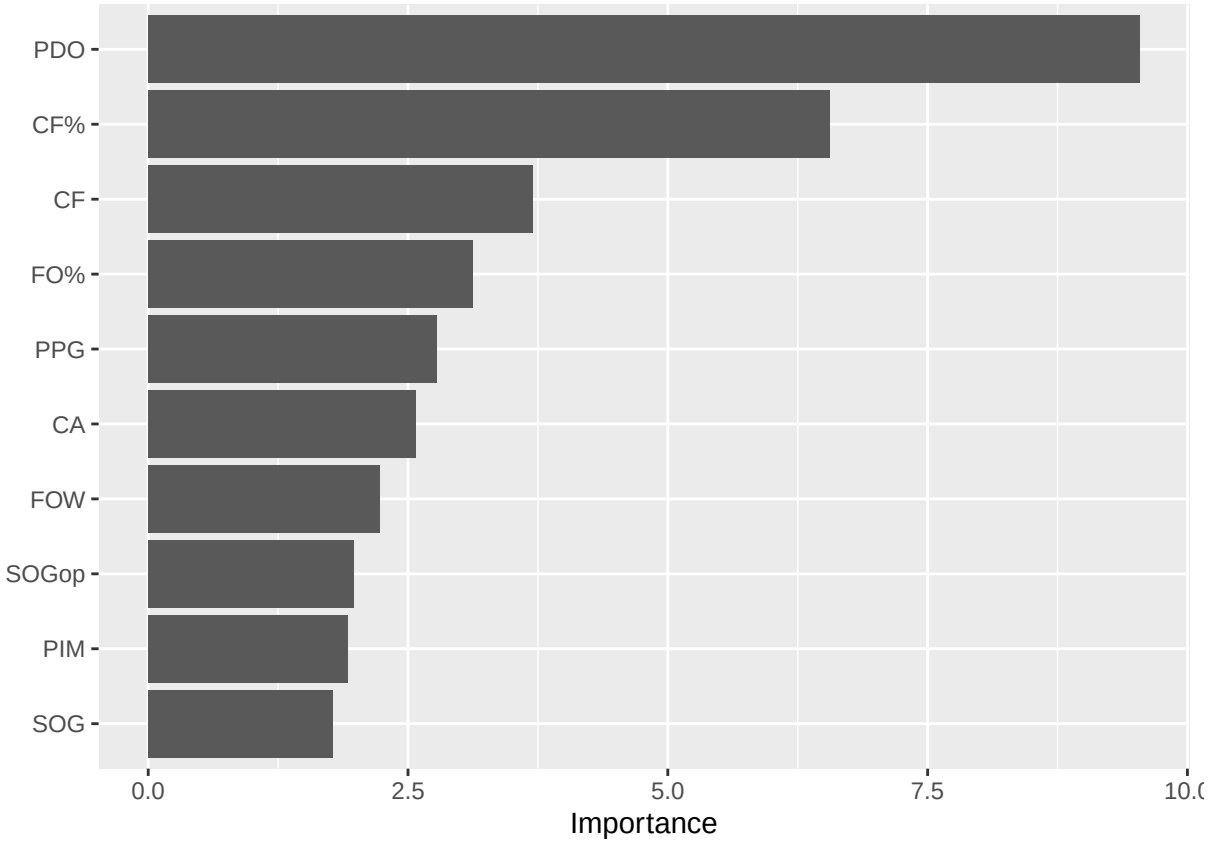
```
augment(wild_final_fit, new_data = wild_test) %>%
  accuracy(truth = WL, estimate = .pred_class)
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.877
```

Even though almost 88% is an improvement from my previous model, this improvement is not as big as I had liked to hope. I am not sure that I will find a model that will reach 95% accuracy since I have all used important variables already, and it is unlikely that tweaking my model a little will yield an increase in accuracy of around 10%.

Similar to the logistic model though, this model missclassified two more losses than wins. There does exist some consistency between the models then. Is the importance between the variables the same as the logistic model?

```
extract_fit_parsnip(wild_final_fit) %>%
  vip()
```



Very interesting. As I had expected, PDO remained the most important variable of all the predictors. However, GF is not the second most important variable, and it not even on the plot! Initially, I had thought this was a fluke so I tested another lasso model while manually inputting many variables including GF to see if there was something wrong. My test turned out the same results. CF% should thus be noted as an important predictor of predicting wins and losses.

For completeness, let's check if a forest model will do any better than either of the two above, and what variables it names as important.

```
#forest
ranger_recipe <-
  recipe(formula = WL ~ ., data = wild_train)

ranger_spec <-
  rand_forest(trees = 1000, mtry=4) %>%
```

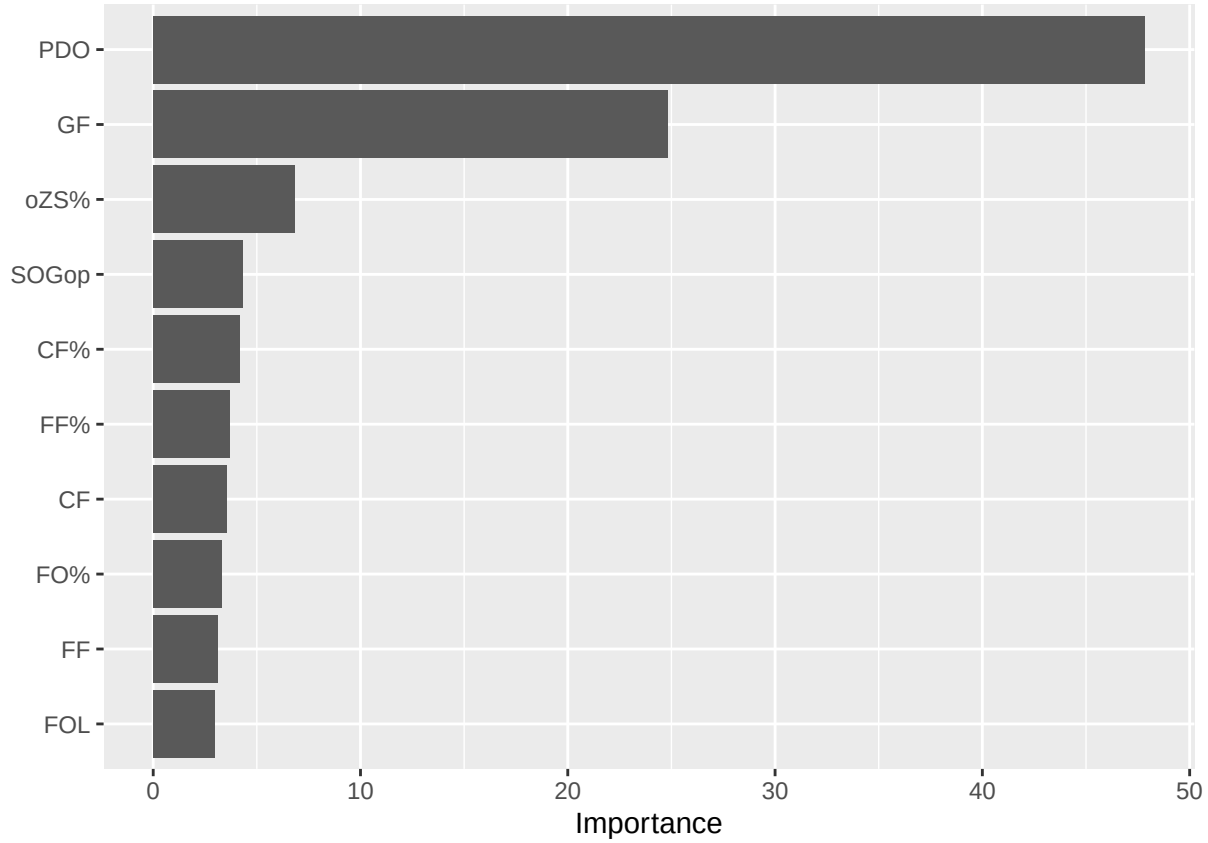
```
  set_mode("classification") %>%
  set_engine("ranger",importance = "impurity")

ranger_workflow <-
  workflow() %>%
  add_recipe(ranger_recipe) %>%
  add_model(ranger_spec)

wild_forest_model <- fit(ranger_workflow, wild_train)

augment(wild_forest_model, wild_test) %>%
  accuracy(truth=WL, estimate= .pred_class)
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.864
```

```
augment(wild_forest_model, wild_test) %>%
  conf_mat(truth=WL, estimate= .pred_class)
```

```
##           Truth
## Prediction  L  W
##          L 30  6
##          W  5 40
```

```
extract_fit_parsnip(wild_forest_model) %>%
  vip()
```



Not surprisingly, the forest model performed similarly to the other models at an 86% accuracy rate. There was one more loss that was predicted to be a win in this model, which is the difference in accuracy between this model and the lasso model. Interestingly though, PDO and GF are the most important variables by far which fits with my earlier prediction that they will be the top two most important predictors in all of the models. oZS% comes next, and CF% is fifth most important so there are some minor things to say about those predictors as well.

Conclusion

From these models, I have found the most accurate model to be lasso model at 87.7%. I am curious as to why this was the model with the highest accuracy, but did not include GF as a very important predictor since it shows promise in eda, and is important in my other two models. I also have found that in all my models, it was a little harder to predict if the will win, and a little easier to predict if they will lose. This difference is minimal, so in order for it to be conclusive there needs to be more analysis.

Unfortunately, I was unable to find a model that predicted wins and losses with 95% accuracy. If this project was to be expanded on, I would attempt to create new variables from the ones that exist in order to discover different relationships between predictors and the results of a hockey game. Ultimately, it remains that hockey is a difficult thing to predict even when there are useful statistics at your hand. To fully comprehend hockey's prediction difficulty, similar projects such as this one would need to be conducted and compared with the results here.

Despite not reaching my goal, there are still some silver linings to this project. A conclusion that I can take is that efficiency is incredibly important (arguably the most important aspect of a game) and games are not one on purely offense or defense. In order for the Wild to win a game, they have to always consider how much they are putting into attacking their opponent vs defending. If there is a large imbalance, this will likely

result in a loss. Even though hockey is a very random sport there is some signal in the noise. Furthermore, keeping the puck in the offensive zone, and controlling the puck overall will help you win games.