Fifa ML Analysis

Juan Carlos Ferreyra

Simulation exercise

Data generation

Data Source: https://www.kaggle.com/datasets/nyagami/ea-sports-fc-25-database-ratings-and-stats/data

EA FC is the largest football game available on the market today. With data of over 16,000 players, this data can serve several purposes for models. In this case, creating a classification model for the players will be the goal. To define some terminology that will be used, the following abbreviations are explained: Position(The position on the field of a football player), CM (central-midfielder, a position known as the "jack of all trades"), CAM (Central-Attacking-Midfielder, a position focused on pushing up the field, attempting long balls and long shots, and dominating in the attack), OM (Out-Midfielder, position that is originally labeled as LM or RM (Left or Right Middle), but for the purpose of the project is aggregated into one position), and CDM (Central-Defensive-Midfielder, focused on not going past the midfield line and on recovering possession.) A player has dozens of statistics, but the main ones are the following: PAC (Pace, or how fast a player is on the pitch), DRI (Dribbling, or how well a player controls the ball), PAS (Passing, the players ability to pass the ball), PHY (Physical, how strong a player is and how much resistance they have on field), DEF (Defending, how well a player is defensively) and SHO (Shooting, or how well a player can throw precise and strong shots).To focus on a more accurate CEF model, we will only take these positions and statistics into consideration, where the goal is to classify a player's position based on these statistics. The amount of midfielders in EA FC is around 6,000 players.

Data Cleansing:

Not every metric is required, and to safe computational power and ensure more precise results, we will only keep the aforementioned columns.Data integrity has been verified through Kaggle.

```
## # A tibble: 6 x 8
##   Name                          PAC   SHO   PAS   DRI   DEF   PHY Position
##   <chr>                       <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>
## 1 "Ã"scar Trejo"                 64    75    77    80    70    77 CAM
## 2 "Ã\u0081lvaro Cuello"          61    57    64    62    39    60 CAM
## 3 "Å\u0081ukasz WolsztyÅ„ski"    61    60    60    62    40    62 CAM
## 4 "Aaron Malouda"                74    54    57    67    32    44 CAM
## 5 "Aaron Molinas"                73    57    72    76    51    63 CAM
## 6 "AbdÃ¼lkadir Ã-mÃ¼r"           82    65    70    77    53    56 CAM
```

```
## # A tibble: 1,864 x 8
##    Name                     PAC   SHO   PAS   DRI   DEF   PHY Position
##    <chr>                  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>
##  1 "Ã"scar Aranda"           77    61    62    72    27    55 OM
##  2 "Ã"scar Clemente"         67    56    67    63    49    47 OM
##  3 "Ã"scar CortÃ©s"          78    66    66    72    48    54 OM
##  4 "Ã"scar Perea"            82    57    59    74    27    62 OM
##  5 "Ã"scar Plano"            82    70    69    72    56    76 OM
##  6 "Ã‰lie Youan"             86    67    62    75    40    62 OM
##  7 "Ã\u0081lex Baena"        74    75    80    78    65    67 OM
##  8 "Ã\u0081lex Berenguer"    85    75    75    83    64    68 OM
##  9 "Ã\u0081lex Bermejo"      73    69    65    72    53    66 OM
## 10 "Ã\u0081lex Cardero"      72    64    65    68    53    46 OM
## # i 1,854 more rows
```

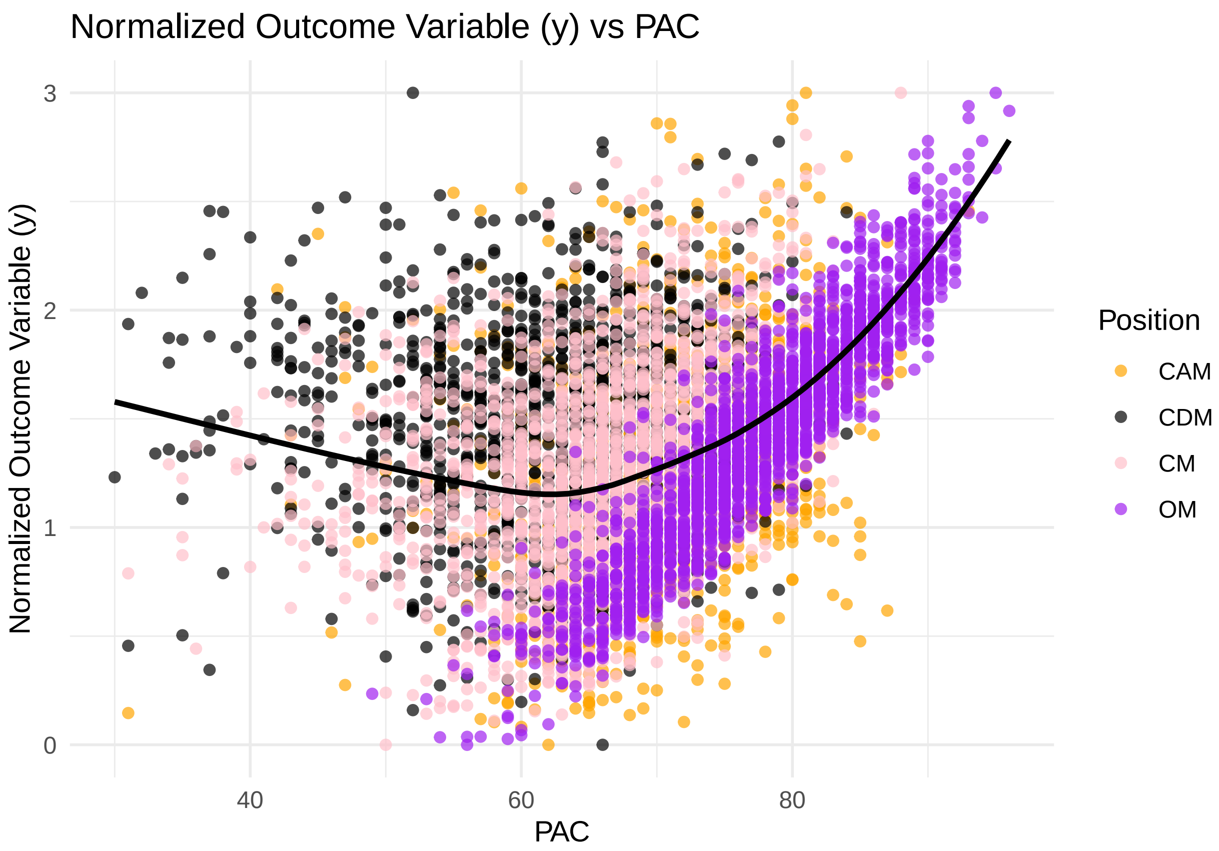
CEF

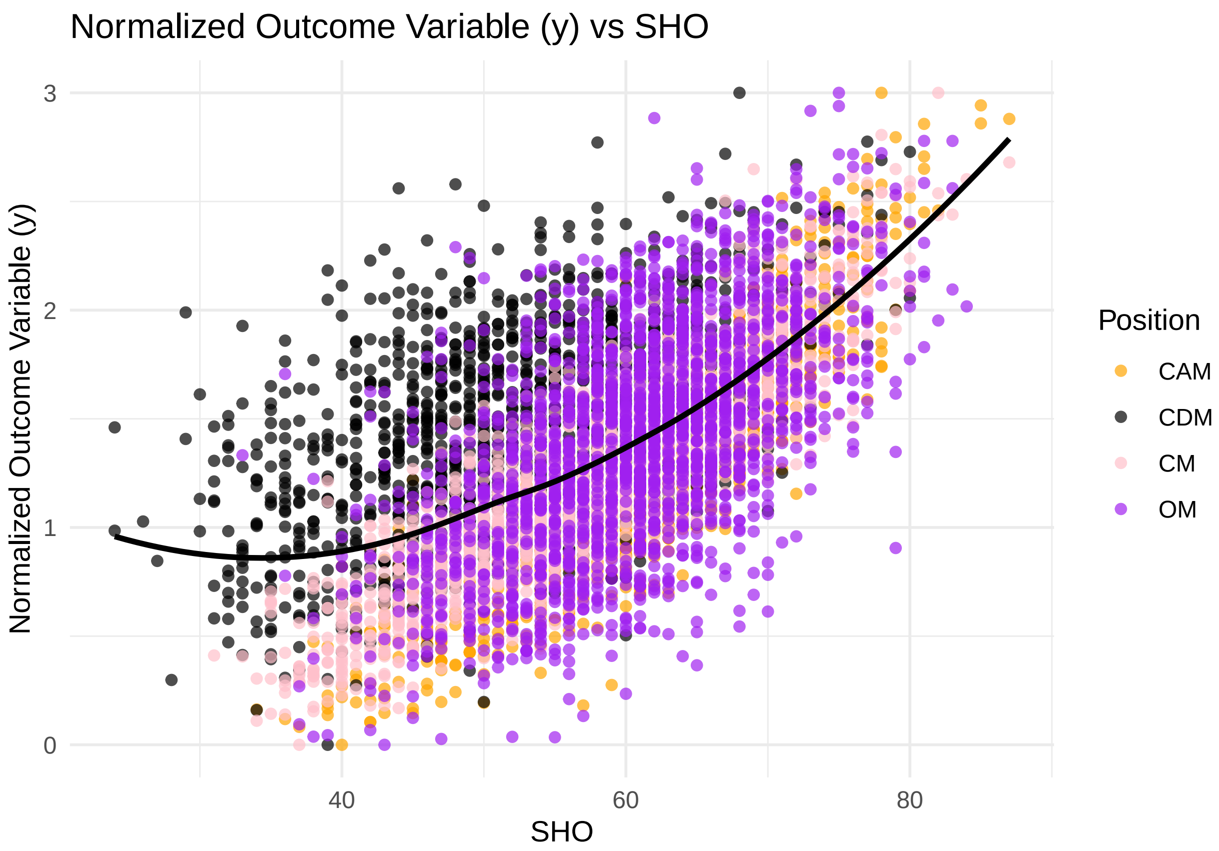
A player's stats in EA FC are not only calculated by ability, but also by international recognition (as a player on a lower and less competitive division may have amazing stats, but the lack of competitiveness makes it so that the stats are relative to the level of the league). This CEF puts weights on different positions based on the importance of the statistic on the mentioned positions. For example, for CAM, passing, shooting, and dribbling, are the most important metrics to consider, while for other positions like CM, a "jack of all trades" approach is taken as it is more balanced. This overall simulates the functional relationship between a player's metrics (predictors) and their position (outcome). The outcome varibale (y) needs to be consistent with real-world domain knowledge.Again, the weights of each metric is based on football knowledge on what managers prioritize for each position. The weighted contributions are used to avoid a single metric dominating others unless it is intentional.

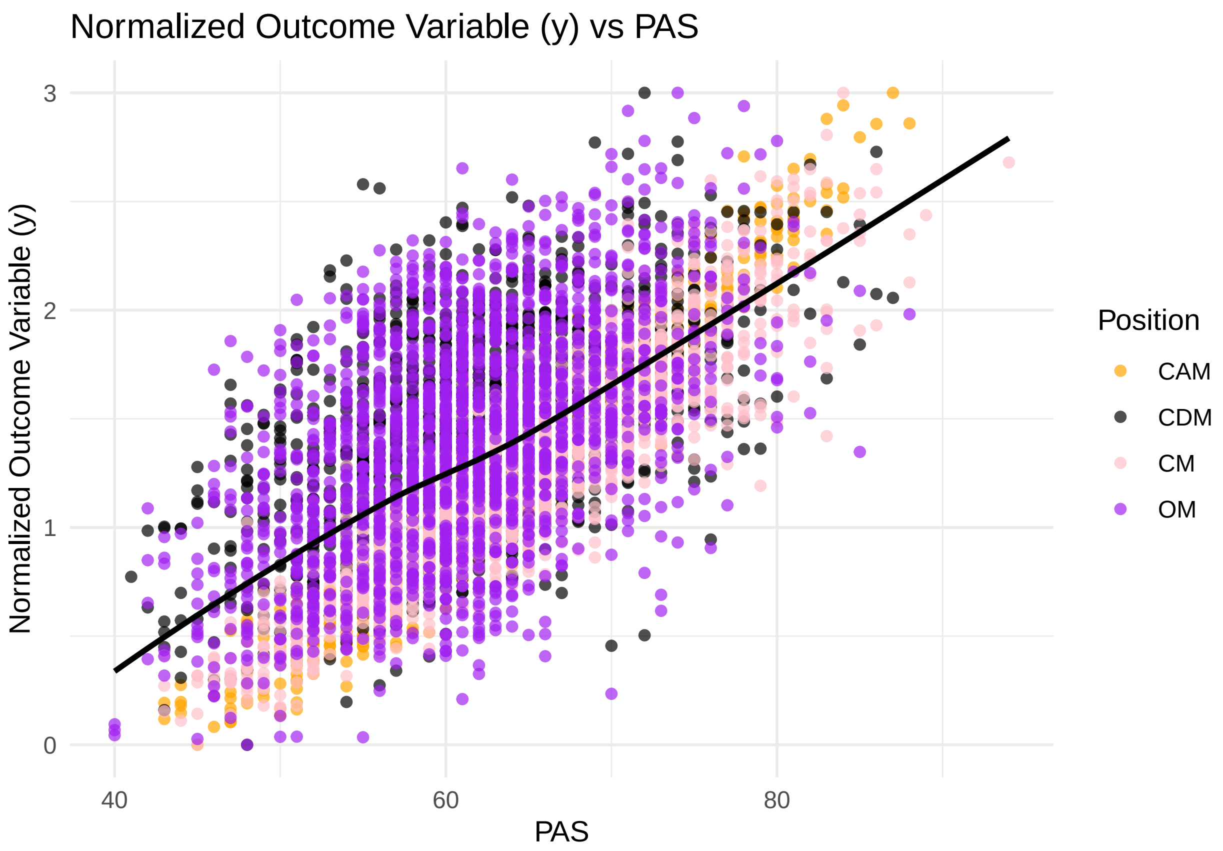
Because of the large variability in a player performance, we add noise, with chosing a standard deviation of 2 to keep outputs stable. We scale the y to a range of 1 to 4 to have comparability in the visualization.

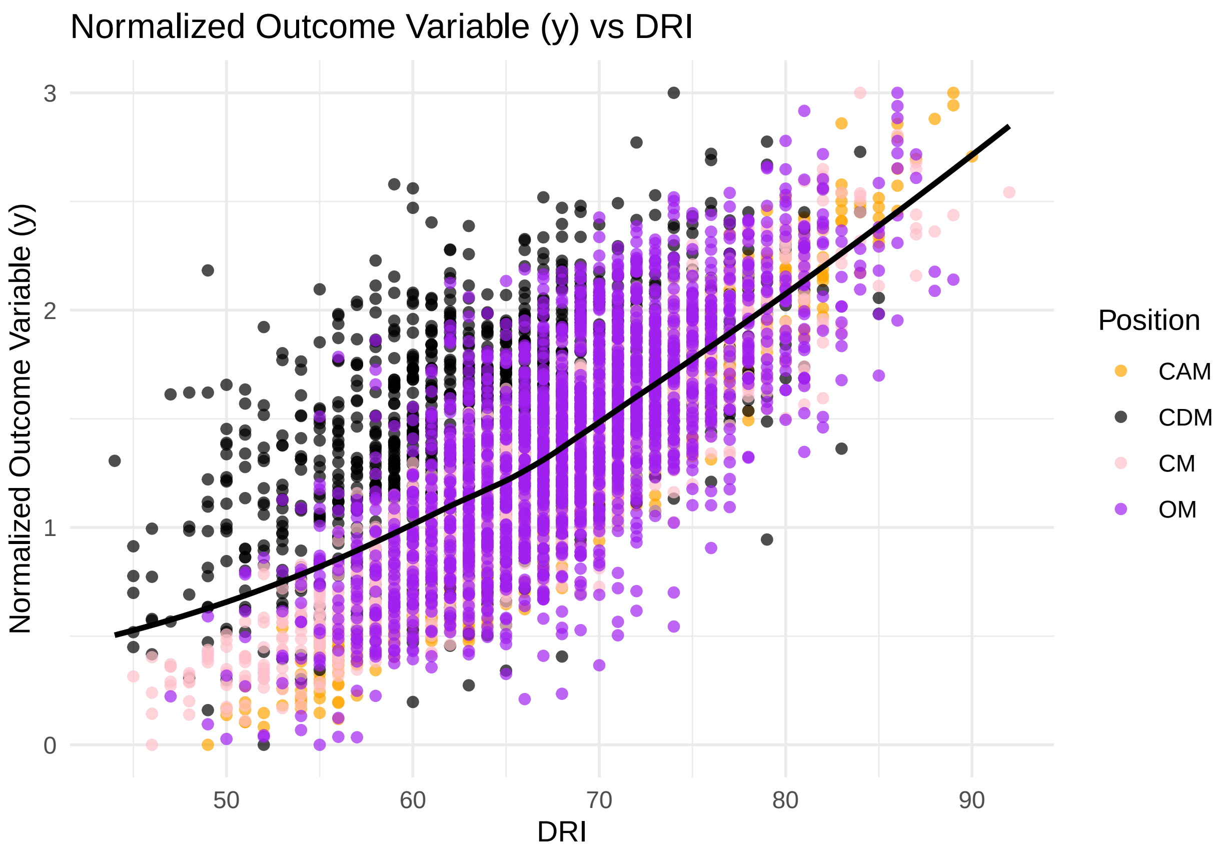
We split the data into training and test data, to later test different models for accuracy, we do a 80/20 split. We also remove the position column for the test data as this will be the outcome variable.

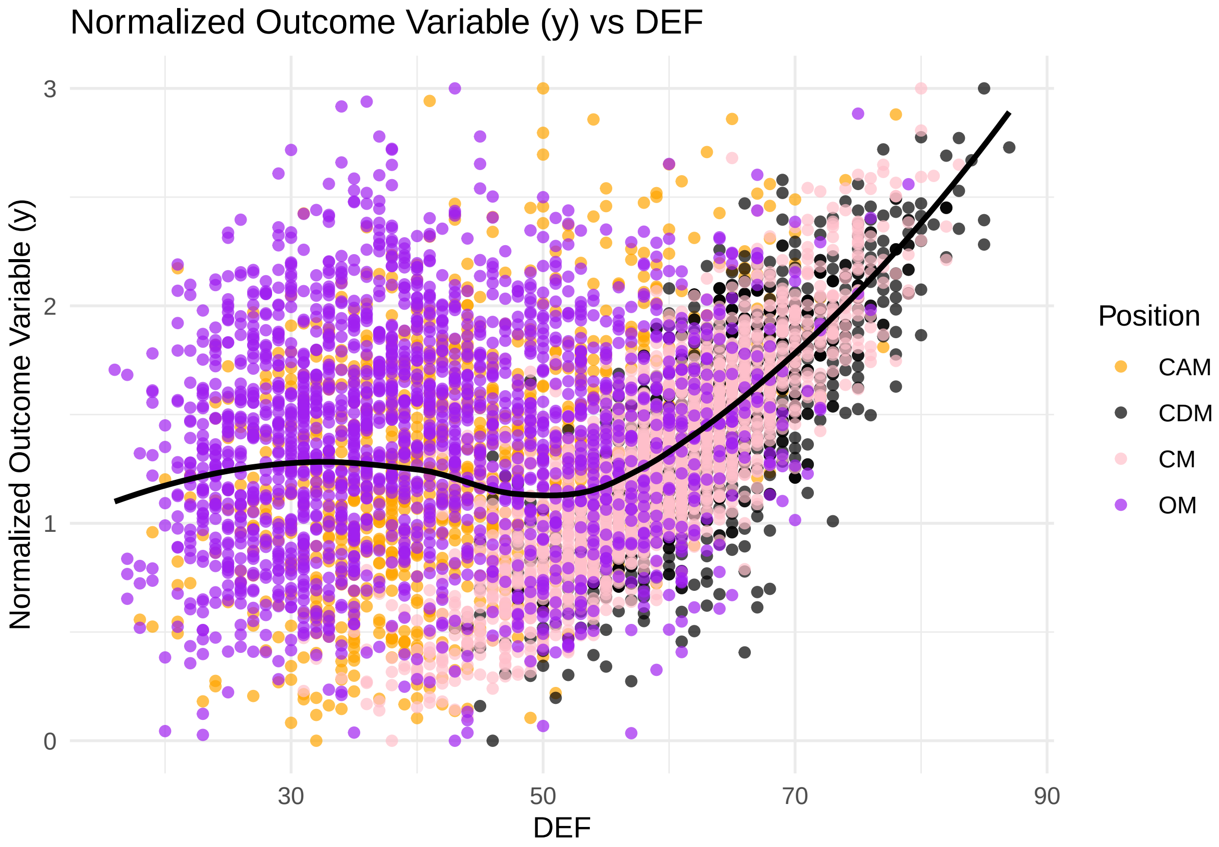
We visualize the outcome variable against each numeric predictor, each position is assigned a different color in order to have a contrast in the visualization.

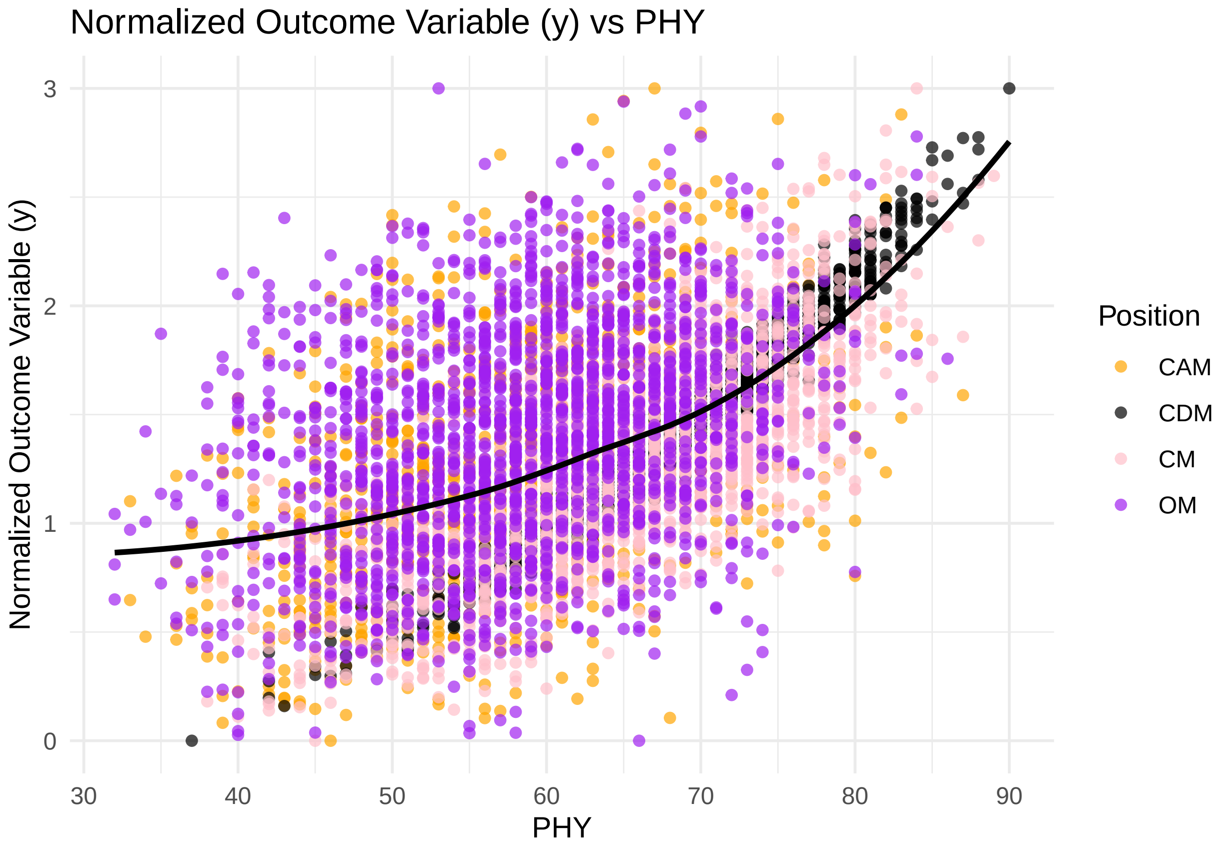

Normalized Outcome Variable (y) vs PAC
Normalized Outcome Variable (y)
3
2
1
0
40
60
80
PAC
Position
CAM
CDM
CM
OM


Normalized Outcome Variable (y) vs SHO
Normalized Outcome Variable (y)
3
2
1
0
40
60
80
SHO
Position
CAM
CDM
CM
OM

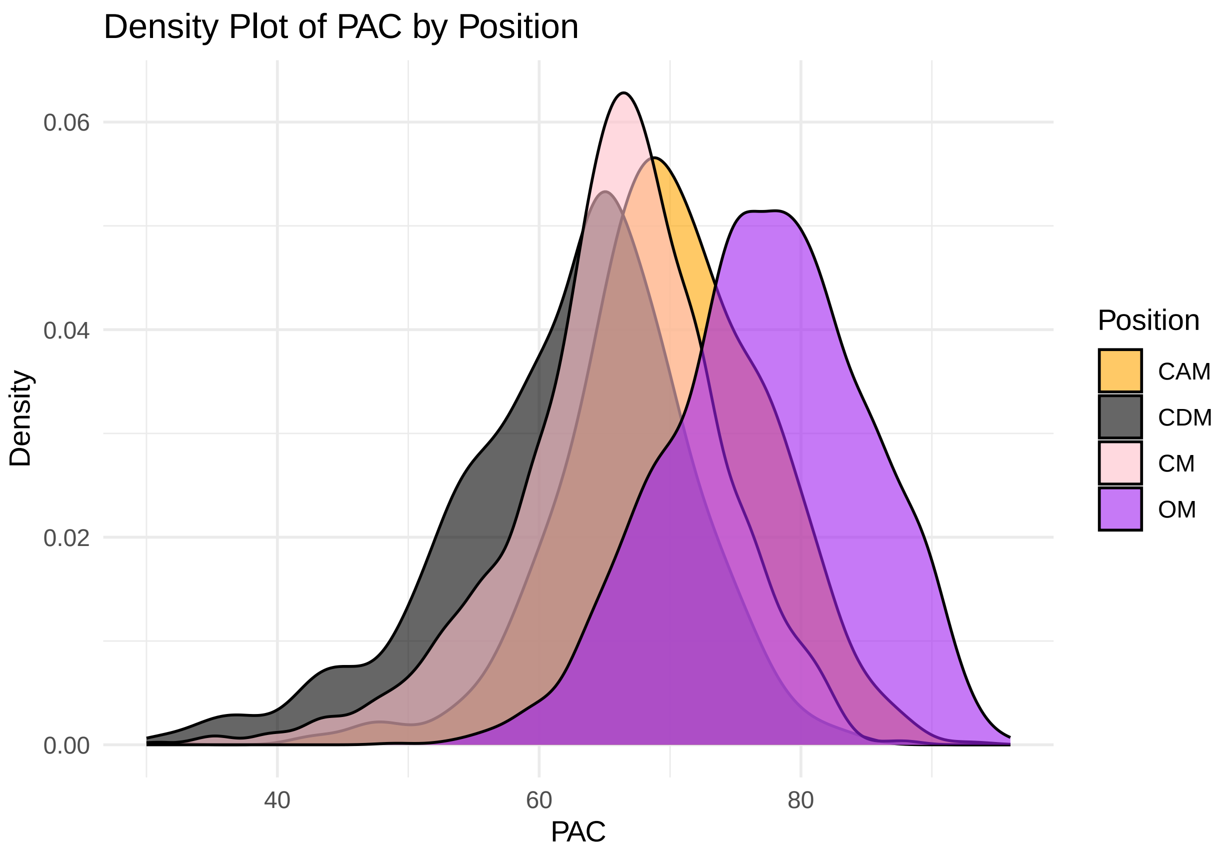

Normalized Outcome Variable (y) vs PAS
Normalized Outcome Variable (y)
3
2
1
0
40
60
80
PAS
Position
CAM
CDM
CM
OM

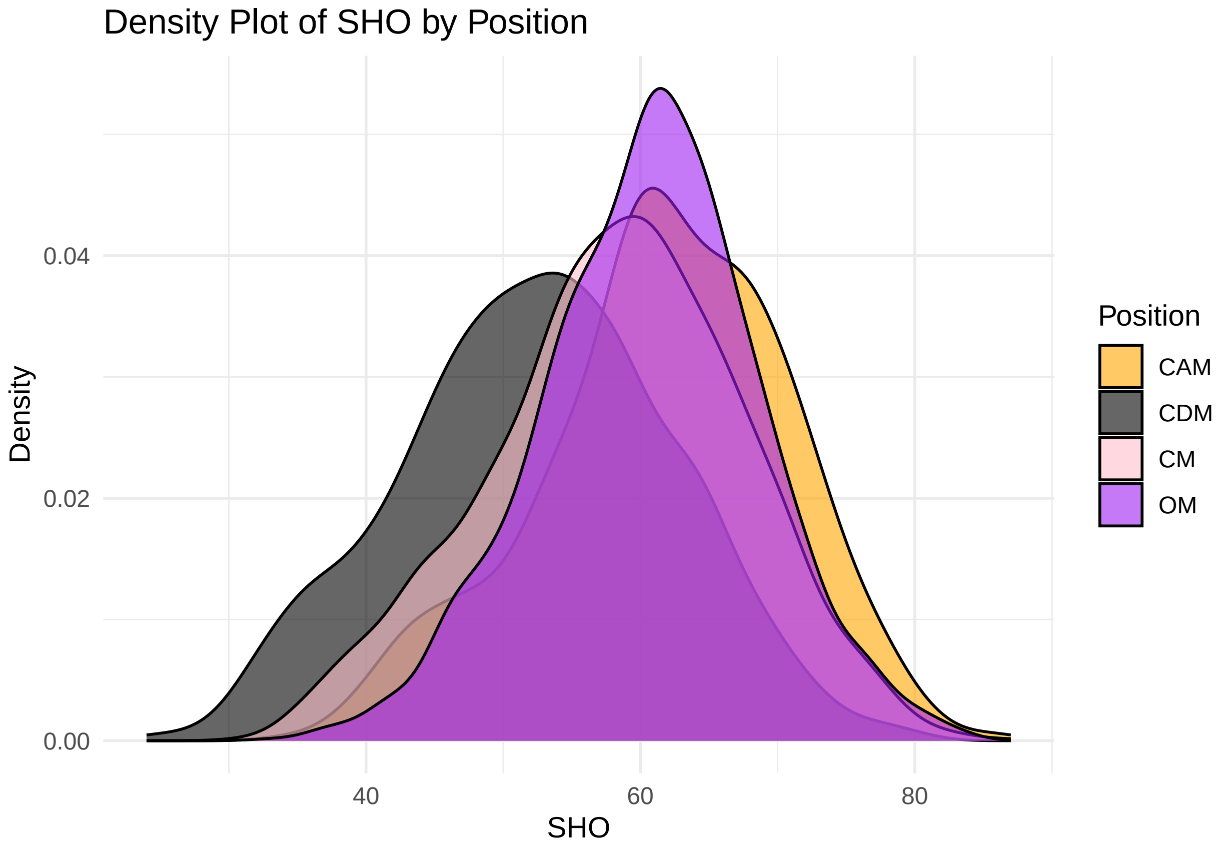

Normalized Outcome Variable (y) vs DRI
Normalized Outcome Variable (y)
0
1
2
3
50
60
70
80
90
DRI
Position
CAM
CDM
CM
OM

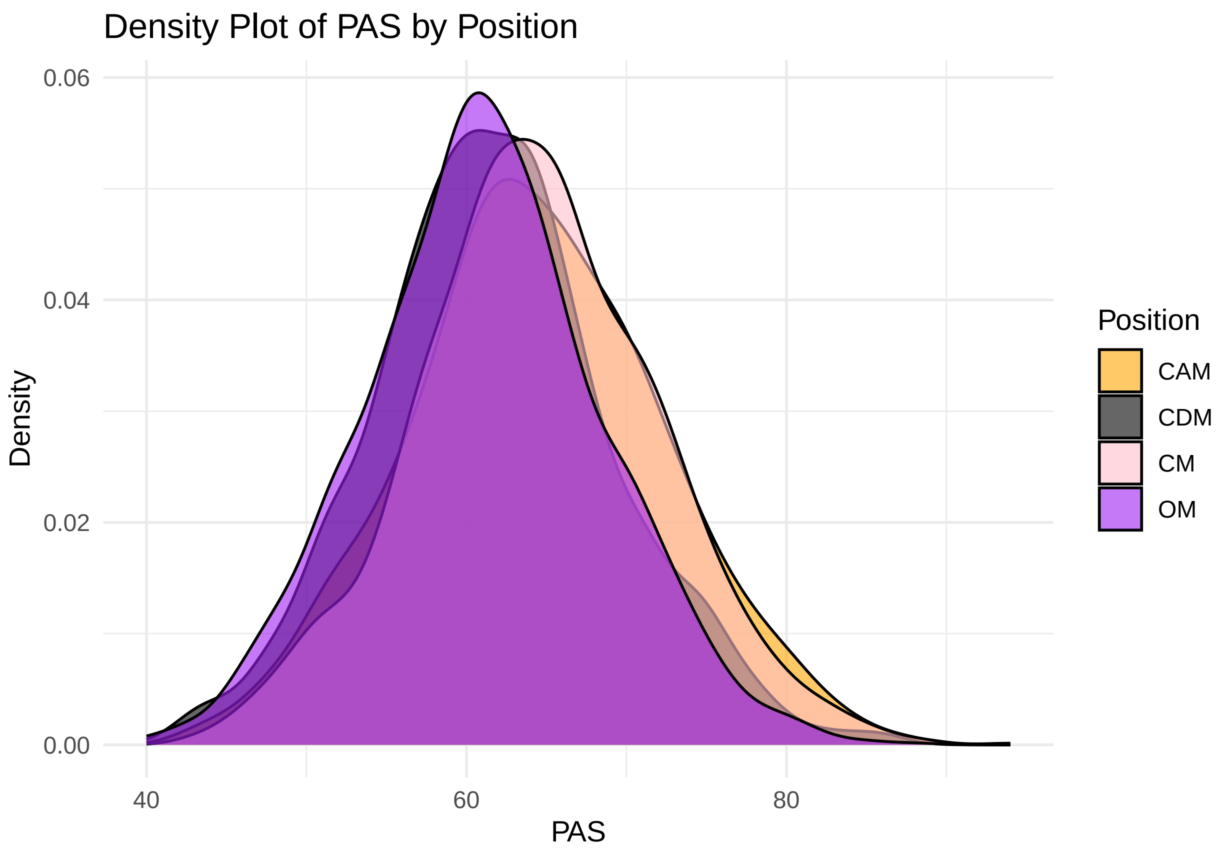

Normalized Outcome Variable (y) vs DEF
Normalized Outcome Variable (y)
3
2
1
0
30
50
70
90
DEF
Position
CAM
CDM
CM
OM

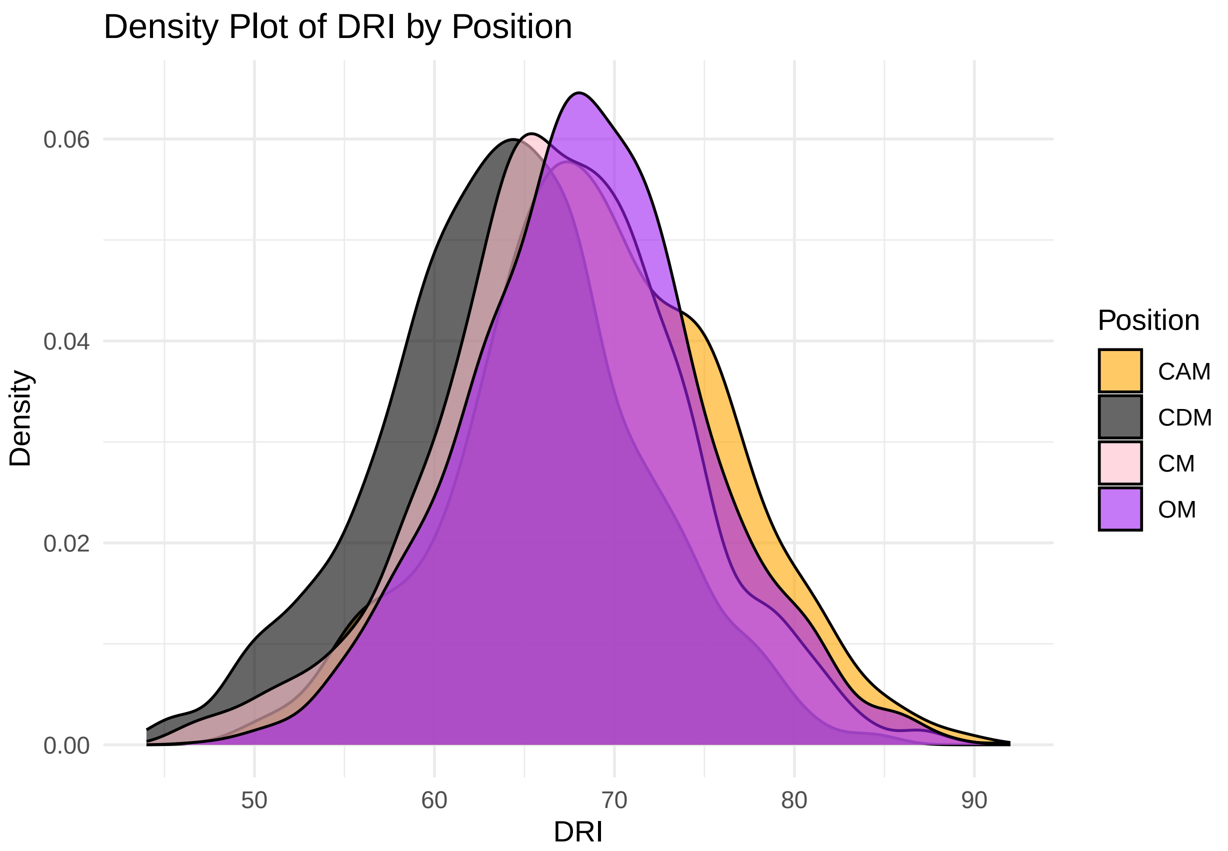


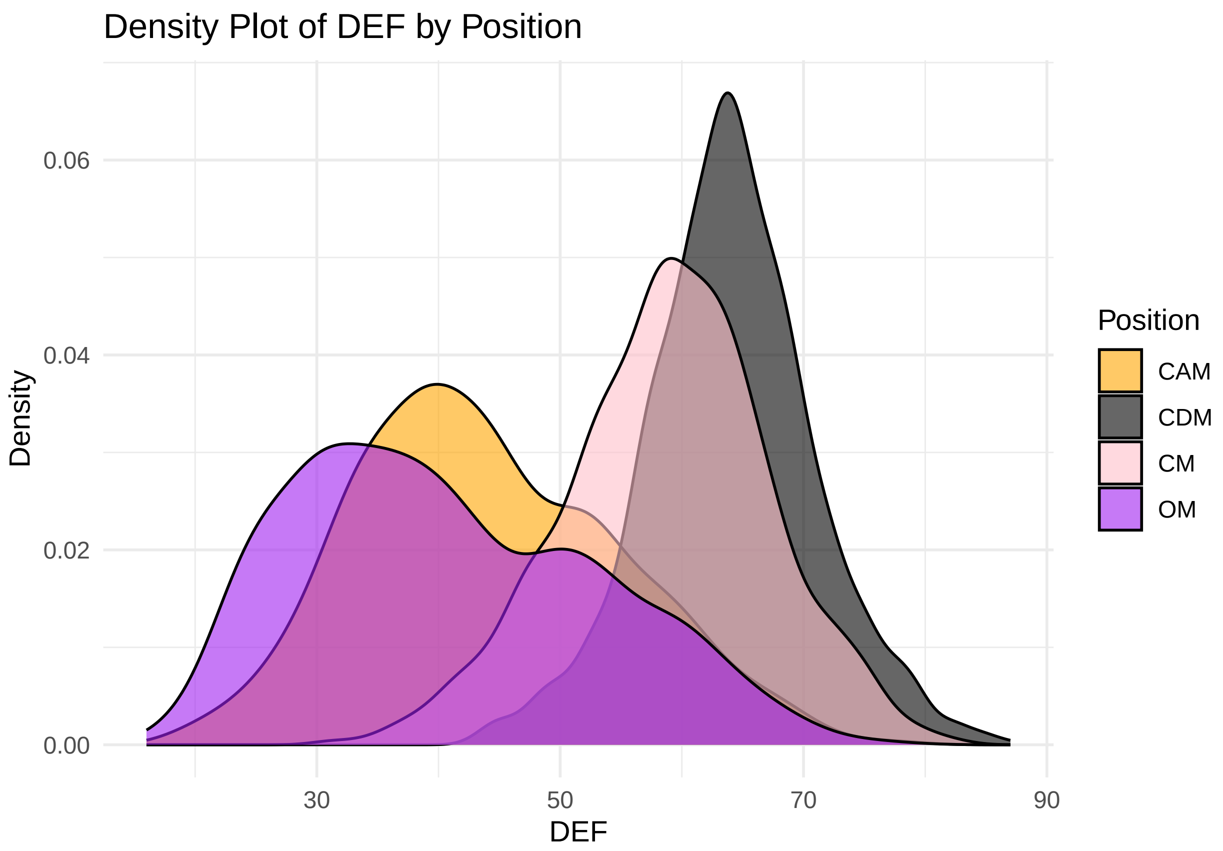
The following plots are representing the normalized outcome variable vs a certain feature (each plot represents a different feature on the x-axis). Each dot would be a player, and the color would represent the position of said player. This allows us to visualize the importance of each feature for each position.One clear example is in the PAC (Pace) graphic, were we can see how OM cluster towards the right, because of the importance of an outfield midfielder to be faster. This trend is followed in the dribbling graphic, as this is another important metric for outfield midfielders. The height of each point is how important the weight is relative to the players position.

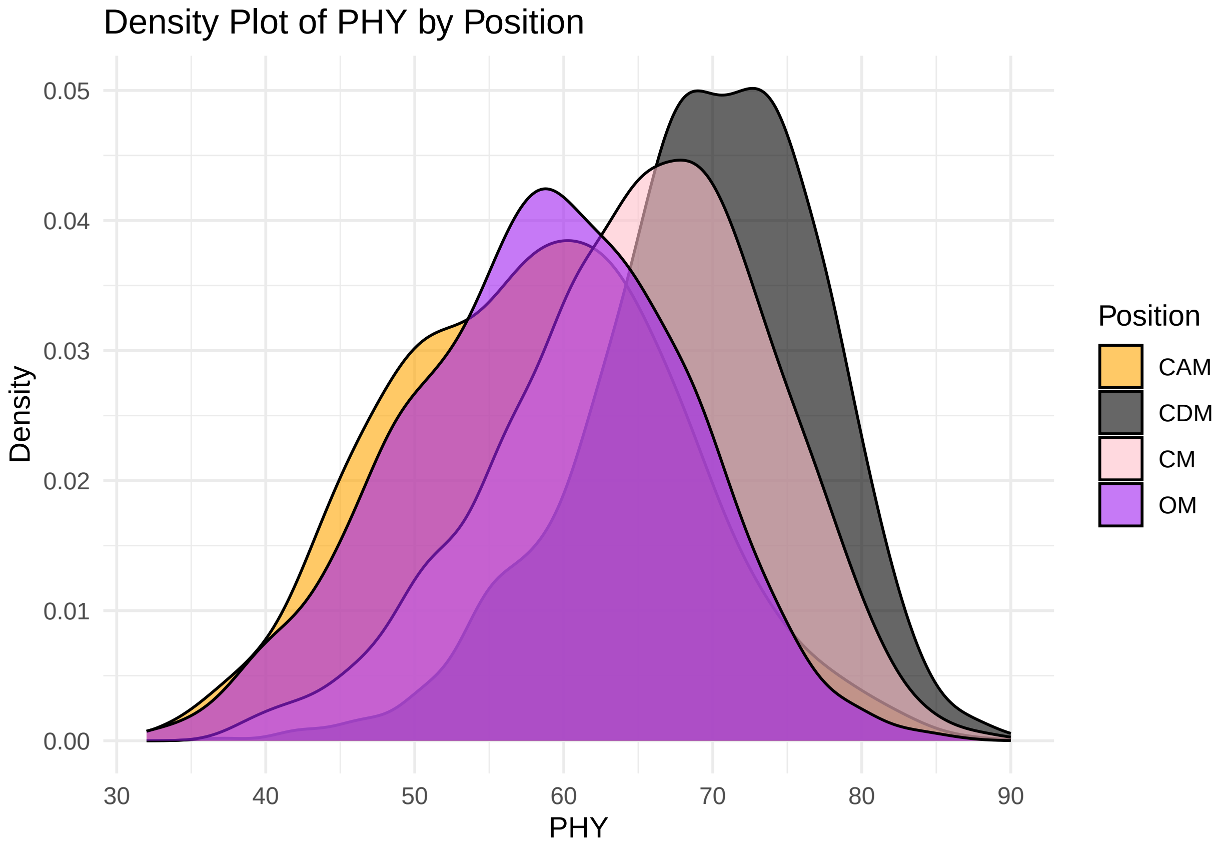

Density Plot of PAC by Position
Density
0.06
0.04
0.02
0.00
40
60
80
PAC
Position
CAM
CDM
CM
OM


Density Plot of SHO by Position
Density
0.04
0.02
0.00
40
60
80
SHO
Position
CAM
CDM
CM
OM


Density Plot of PAS by Position
Density
0.06
0.04
0.02
0.00
40
60
80
PAS
Position
CAM
CDM
CM
OM


Density Plot of DRI by Position
Density
0.06
0.04
0.02
0.00
50
60
70
80
90
DRI
Position
CAM
CDM
CM
OM


Density Plot of DEF by Position
Density
0.06
0.04
0.02
0.00
30
50
70
90
DEF
Position
CAM
CDM
CM
OM



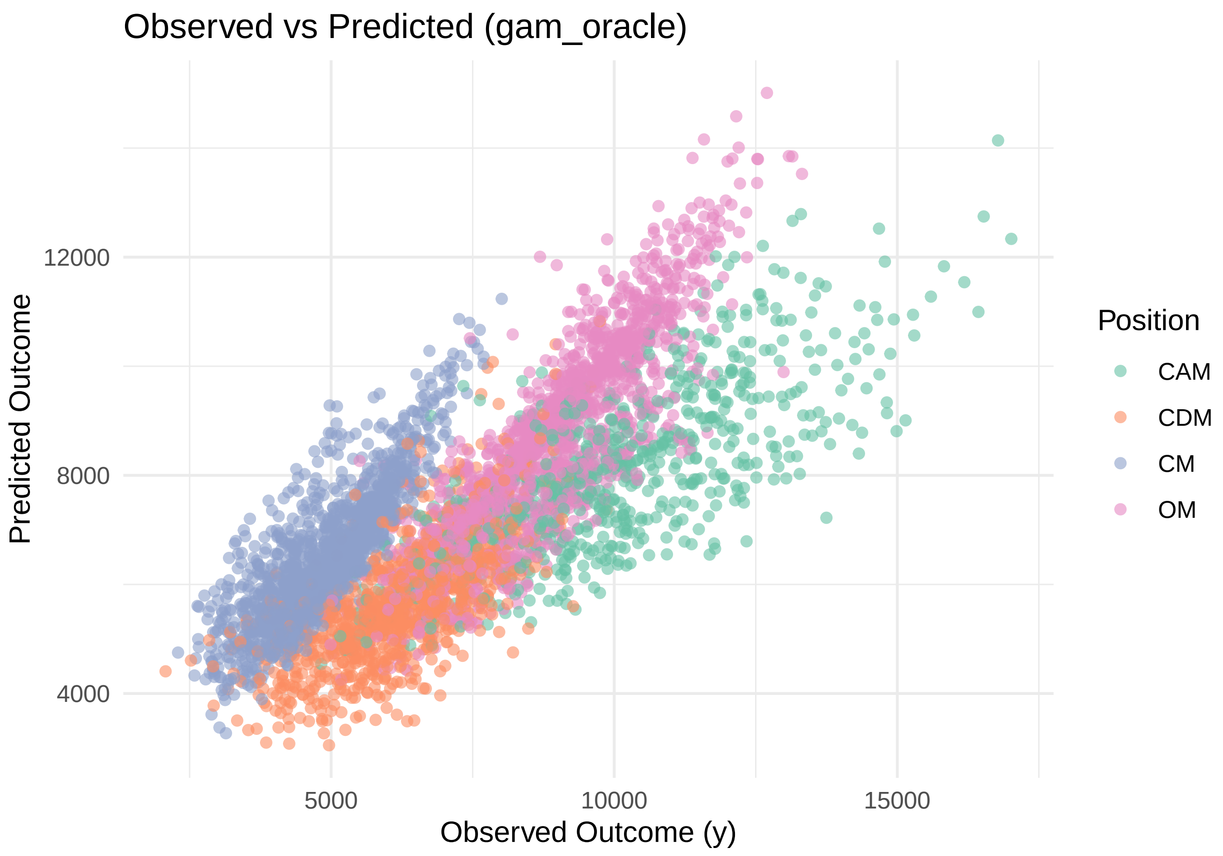
With the following density plots, we can see the range of where players are falling on in regards to the specific predictor variable.The values are most concentrated at a different rate depending on the feature. For example, for the defense feature graph, we see how CDM has a high density, because of the high importance that the defense metric has on the position.

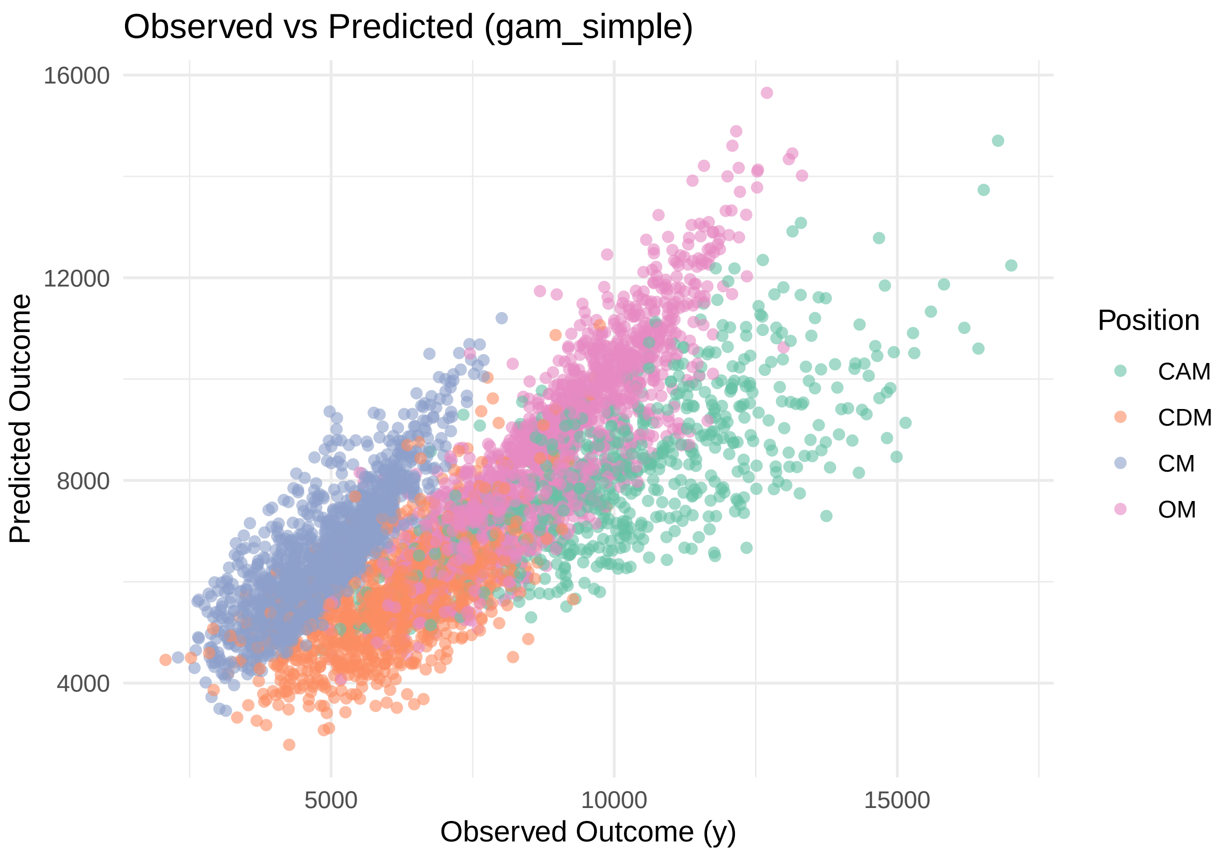
Additive models

GAM Oracle considers interaction terms like the interaction between pace and passing, and the interaction between shooting and dribbling. This is because the interaction between these terms can be different for an OM than for a CAM.

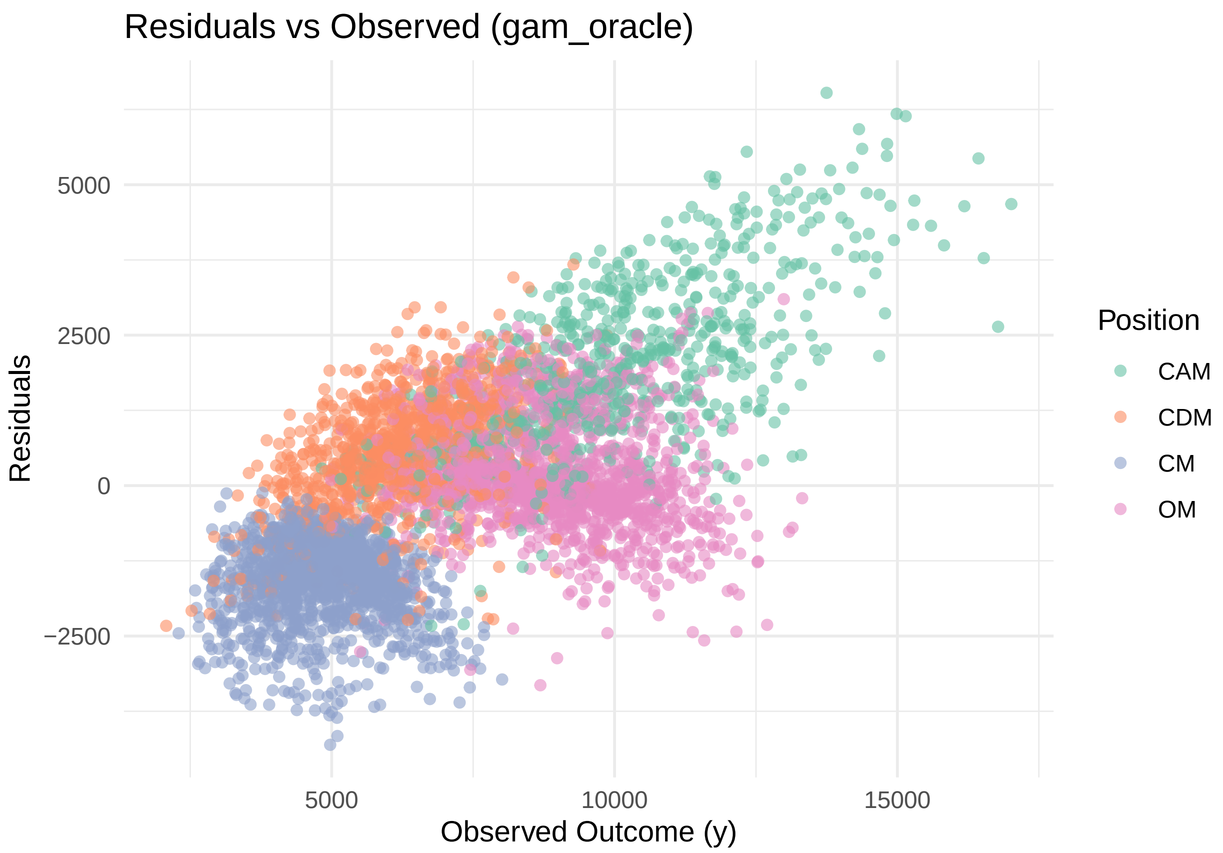
GAM Simple considers individual terms, using the smooth function on each term without considering the potential interaction between them. As we are dealing with potential non linear data, we smooth out each individual term.

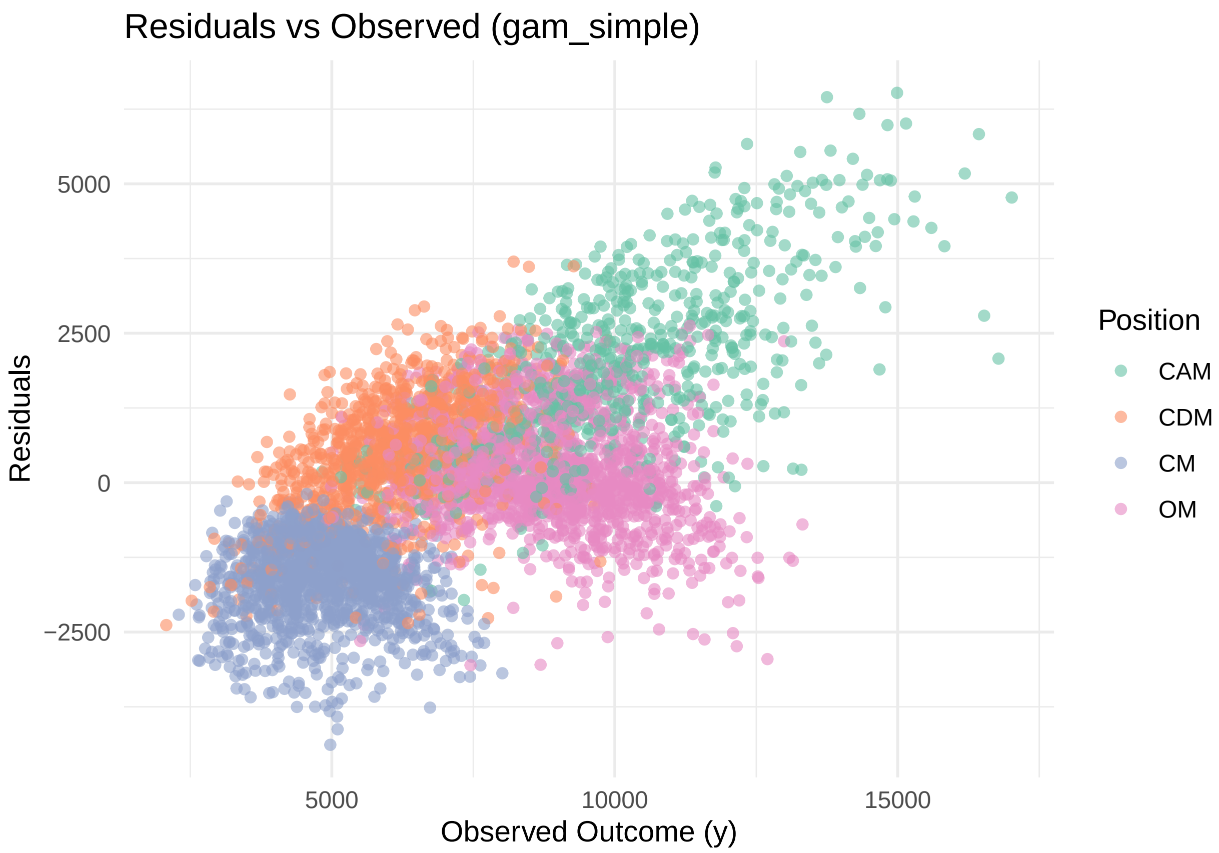
We then predict with the GAM Oracle and GAM Simple models approaches, visually comparing them to further understand their effectiveness in the context of EA FC. With a smaller residual, we can infer a better prediction, as it would consider the distance between the actual value and the predicted value.


Observed vs Predicted (gam_oracle)
Predicted Outcome
12000
8000
4000
5000
10000
15000
Observed Outcome (y)
Position
● CAM
● CDM
● CM
● OM

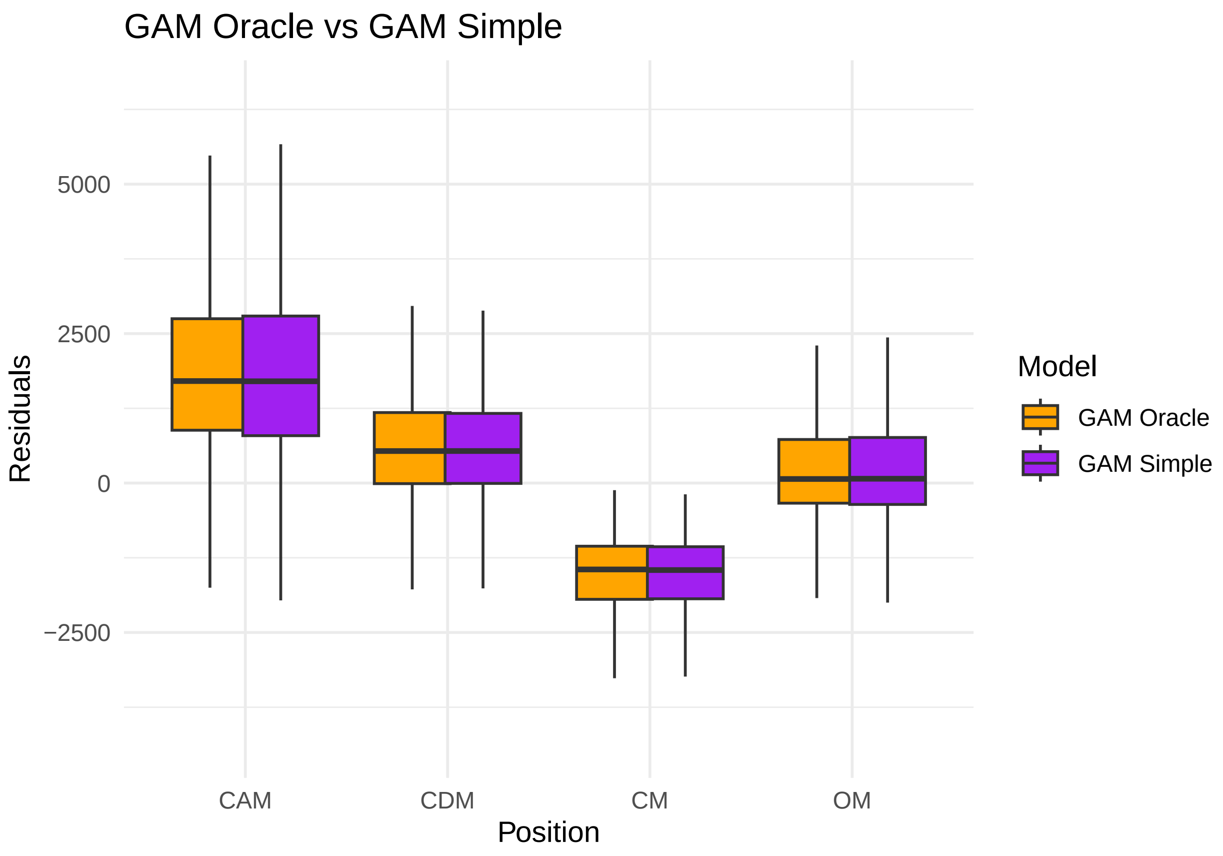


While observing a clear separation between clusters, this graphic is not sufficient to evaluate which model might be more effective. There is a strong positive correlation between observed outcomes and predicted outcomes, indicating that the model captured the trend well. We can also see how OM has a higher variability than other positions, which makes sense in the context of real-word OM being extremely versatile in their position. Because of the way the CEF assigns the weights, the clusters get spread out throughout the graph. As the CEF utilizes squared terms, the values that are higher get significantly amplified because of the non-linear relationship.


Residuals vs Observed (gam_oracle)
Residuals
5000
2500
0
−2500
5000
10000
15000
Observed Outcome (y)
Position
CAM
CDM
CM
OM



In both plots, residuals are distributed accross an observed outcome range, however, the spread of the residuals in the GAM oracle model appears to be slightly more concentrated than the GAM simple.The residuals show hetoskedasticity, which was predictable because of the non-linear relationship in the data, where the GAM oracle is able to approach better through the interaction terms being considered. The GAM Oracle slightly captures the relationship better, however, another visual would be ideal to further explore this evaluation.



With presenting the residuals in boxplots, we are able to better see the difference in residuals. Accross the fours positions, we can see that the spread of residuals for CAM is larger, which can help determine decisions later on. For both CM and OM, we can see how GAM Oracle has slightly better performance, as they are closer to 0 when compared to a GAM Simple.

```
## [1] "GAM Oracle: MAE = 1202.12, RMSE = 1520.69"
```

```
## [1] "GAM Simple: MAE = 1209.44, RMSE = 1530.58"
```

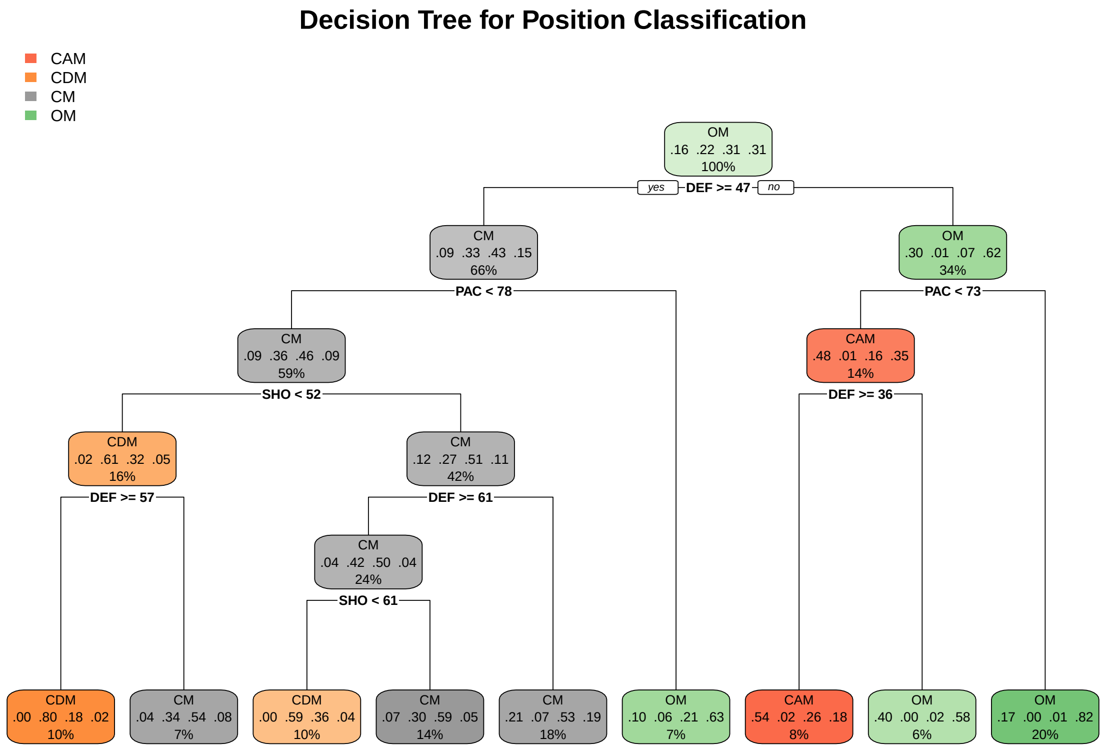
By utilizing metrics, we can more precisely determine not only which model is most effective, but also by how much. As evidenced in the lower Mean Absolute Error, and the lower Root Mean Squared Error, we can see how GAM Oracle ultimately performs better in both categories that can summarize the model performance numerically. This can conclude that while the GAM Oracle is better, the slight difference may indicate that the simple additive terms in GAM Simple are sufficient for most variance in the data.

Tree based models

For the decision tree, we decide to use a 10-fold for cross-validation, as the data set is relatively small, and so computational power will not be the biggest concern.

We calculate RMSE and MAE to ensure that while the complexity parameter may be reasonable, that the improvements on RMSE and MAE are worth the potential risk of over fitting.

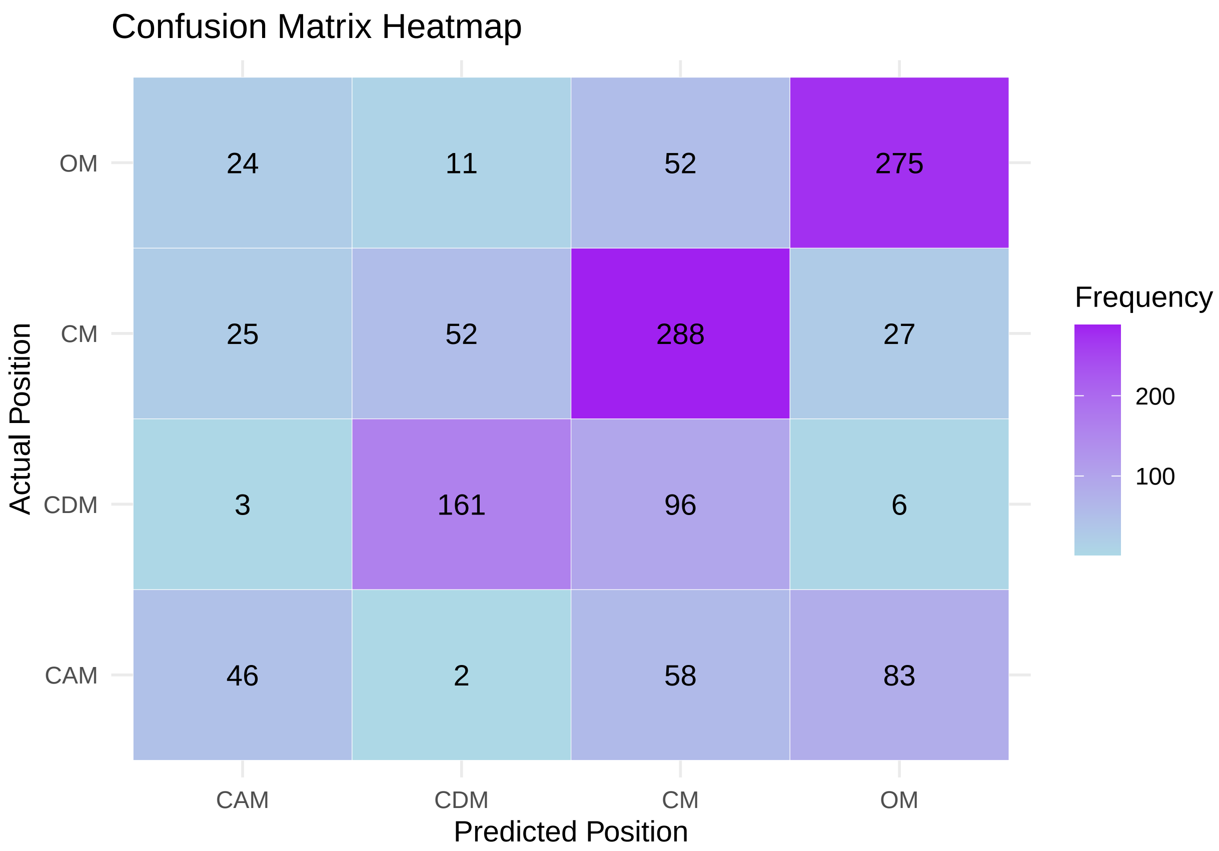
```
##           cp
## 1 0.01289742
```



```
## [1] "Decision Tree RMSE: 1.15"
```

```
## [1] "Decision Tree MAE: 0.62"
```

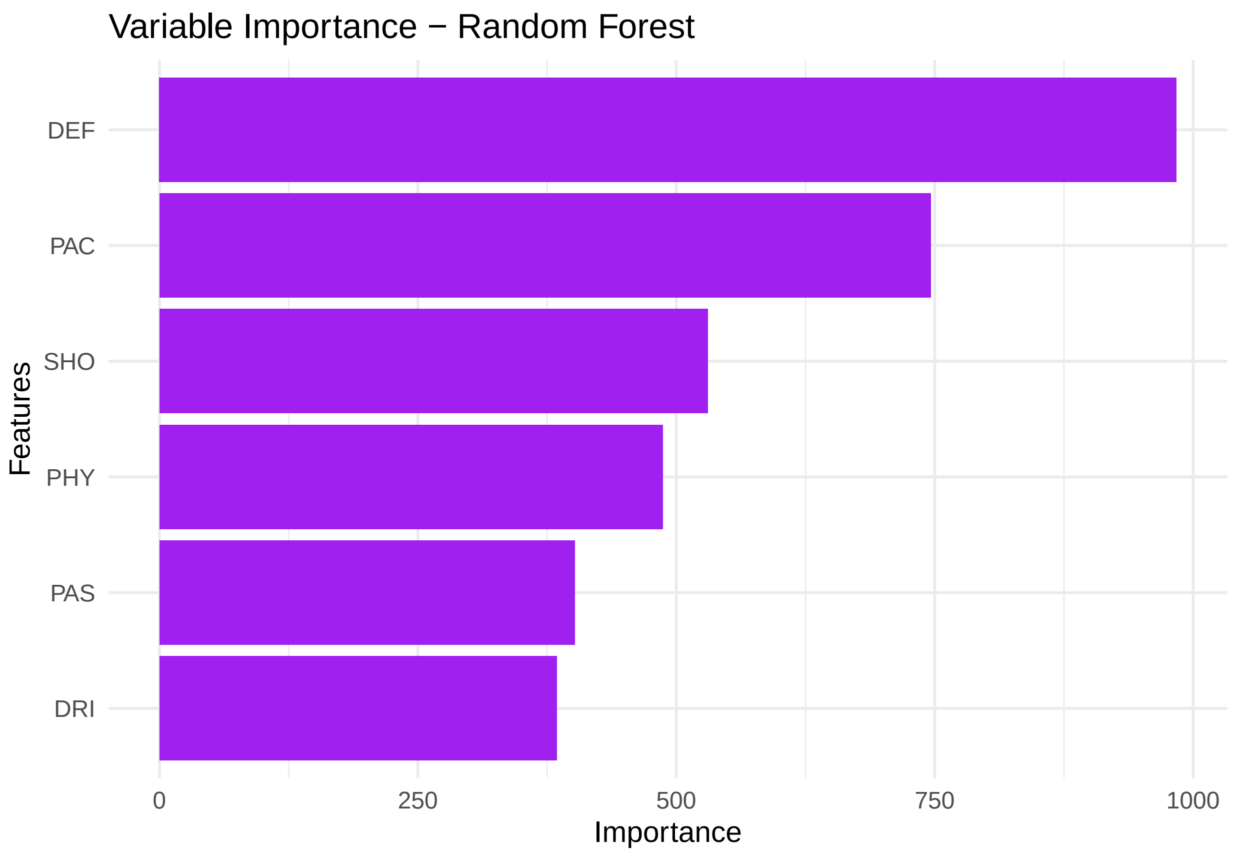
When experimenting with different tuneLenght, we can evaluate that 7 provides a sufficiently low complexity parameter that aligns with the size of our data set. Additionally, the risk of over fitting is reduced because of the mentioned size. As the test RMSE and MAE slightly reduce, we can confidently conclude that a tuneLenght of 7 sets a balance on model complexity and performance. Evaluating it further, it is a suitable choice for our dataset considering it can minimize errors while avoiding over fitting.



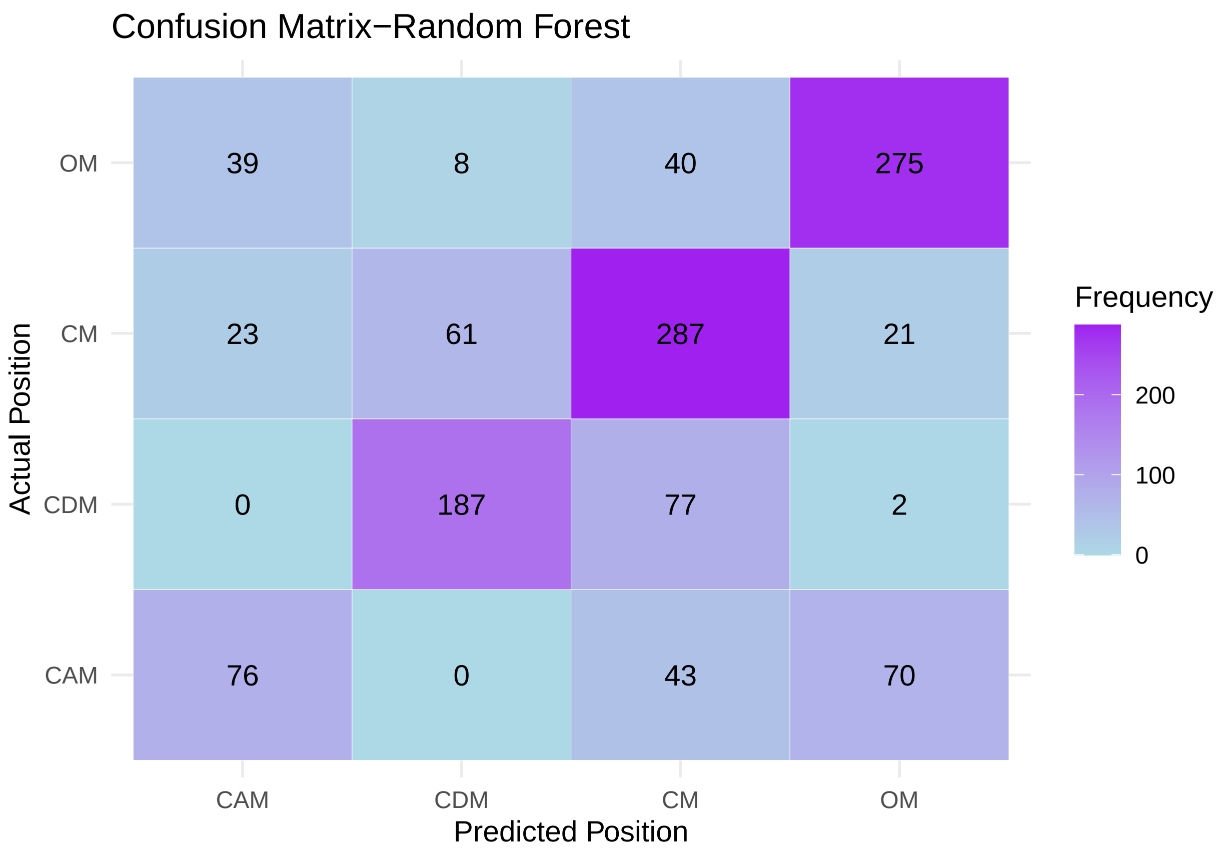
A confusion matrix allows us to have a different perspective on the results of the single tree. Firstly, as mentioned, we aggregated Left midfielder with Right midfielder, as OM, leading to OM having the higher frequency. Additionally, this highlights classifications across all points, with this also leading to OM having the highest misclassification rate. The next most confused class is the central midfielder, which is also logical as their rounded skillset makes it harder to predict as their feautres may overlap with CAM or CDM. The logical relationship between the midfield positions is reflected with the misclassifications, as features often overlapp between the positions.

```
## note: only 5 unique complexity parameters in default grid. Truncating the grid to 5 .
```

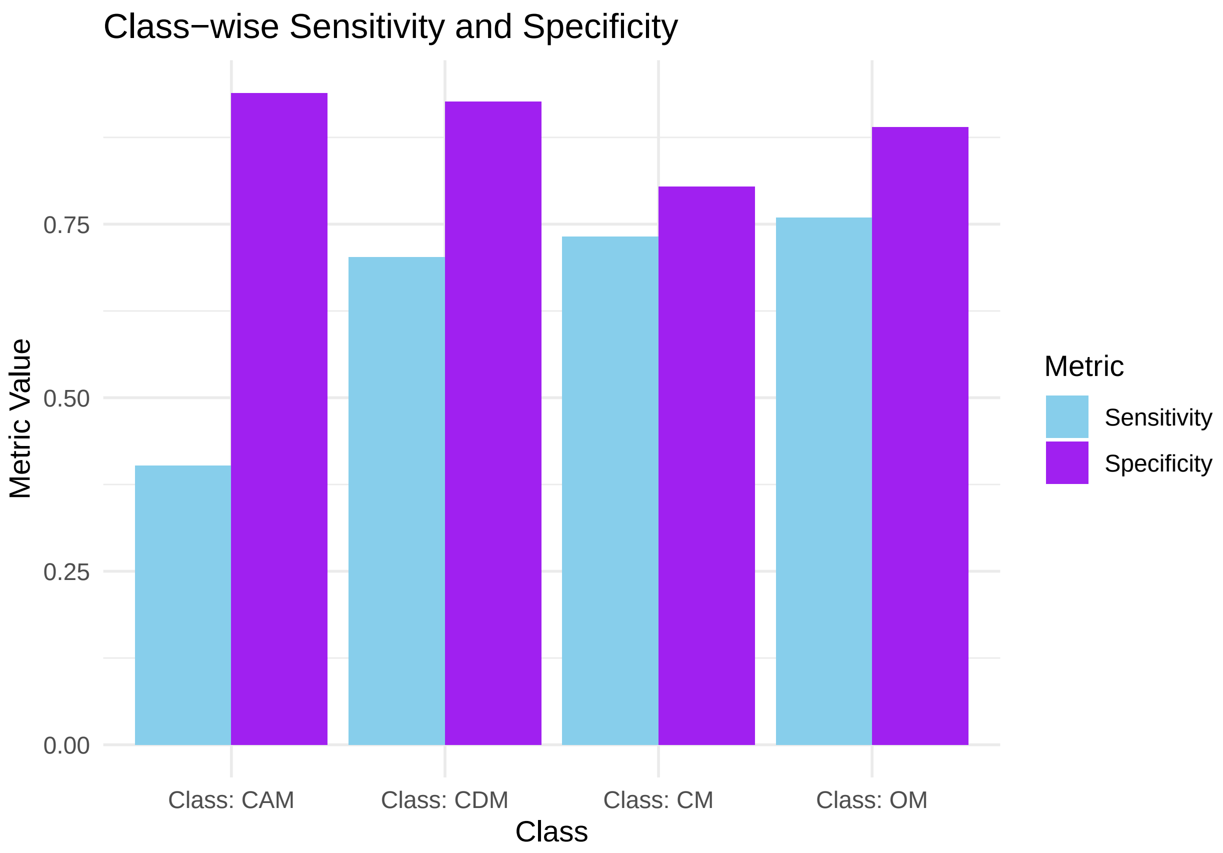
The VIP plot ranks the features based on the importance that they have in splitting the data. While affected by the CEF, this allows us to see the impact that each feature has, and most importantly their contribution towards a more accurate classification. This information will also help us understand the process for OOD later in the project. In this case, we see the impact that defense has, and I am assuming that this is mainly in the positions of CDM or OM, where in one a high defense indicates a higher probability of being classified as a CDM, with the opposite being true for OM (lower defense increasing chances of being classified as an OM).



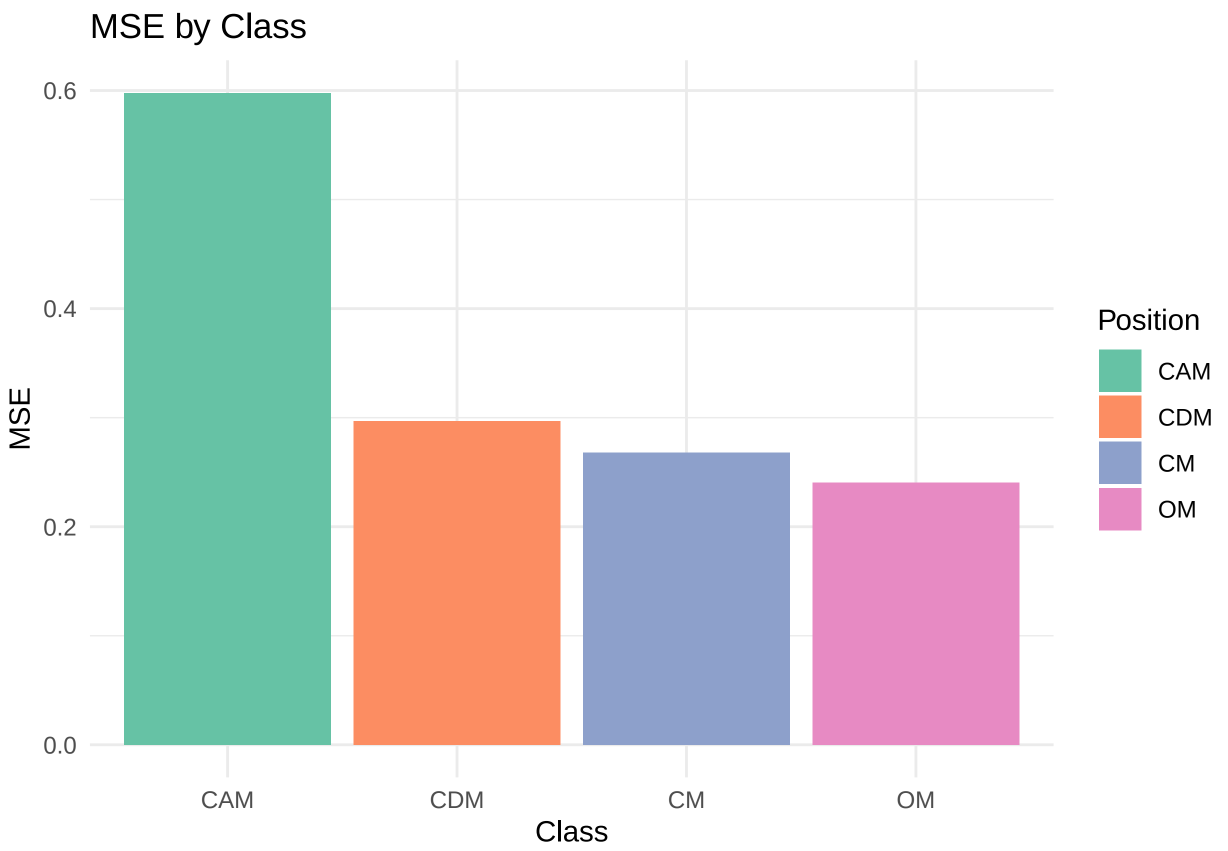
For the confusion matrix heat map, we can see the comparison of the predictions compared to the actual labels. We see a similar pattern that we saw in the simple tree, where OM was the most commonly miss classified. One potential limitation that we see again is the rate at which OM are being miss classified, which raises the question of whether the increased sample should be further tuned (in other words, if aggregating LM/RM into a OM position was the appropriate choice). In my opinion, it was an appropiate choice considering the size of the data set, but would have to consider doing a project with a larger data set to evaluate if there are differences amongst right and left footed outer midfielders.



The recall rate indicated the proportion of actual positive cases which were correctly identified by the model. We can see from this plot for example how the CAM position was the one which was least detected. The potential limitation with this is the CEF might have overestimated or underestimated the importance of certain features on a CAM player, leading it to have a smaller recall rate.In terms of specificity, the model correctly is able to identify the positions and not missclassifying with true negative cases. This is lower for CM, which makes sense because again, it is the position where most metrics might overlap.



One usage of the MSE or the mean squared error is to evaluate how well each model was detected. We can see how CAM was the class with the highest MSE, which is consistent with it also being the least sensitive class. Again, this calls to question why CAM is not being as accurately classified as other, and a potential change in the CEF might be needed to be implemented if this project is to be continued.



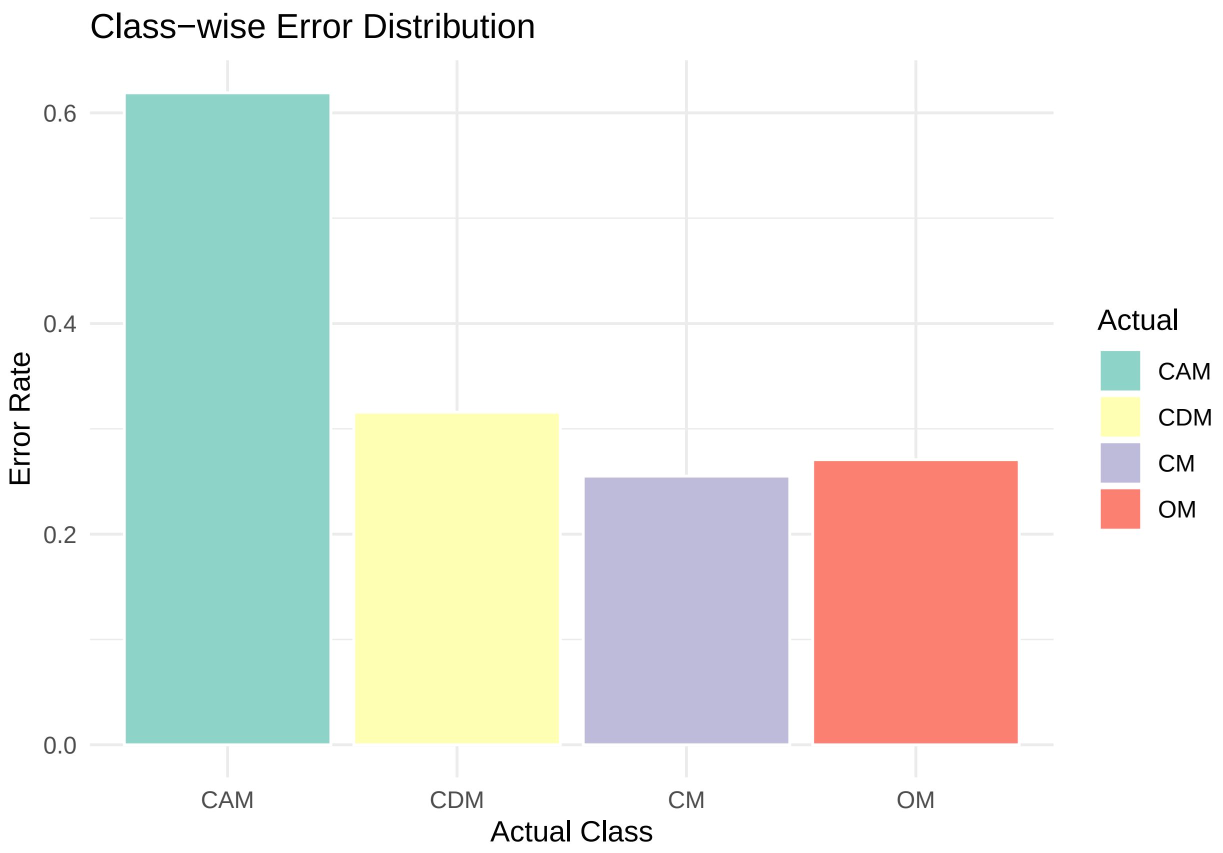
```
## [1] "Boosted Tree RMSE: 1.14"
```

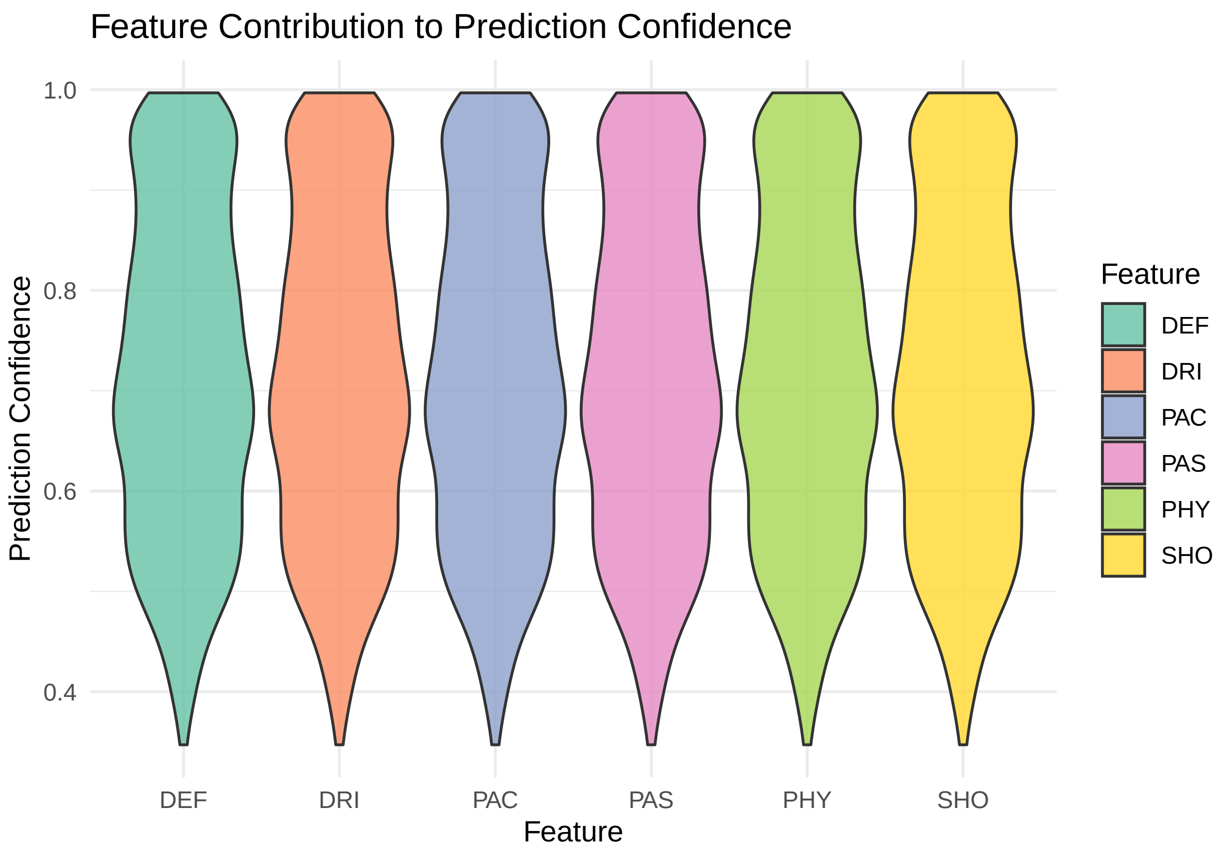
```
## [1] "Boosted Tree MAE: 0.59"
```

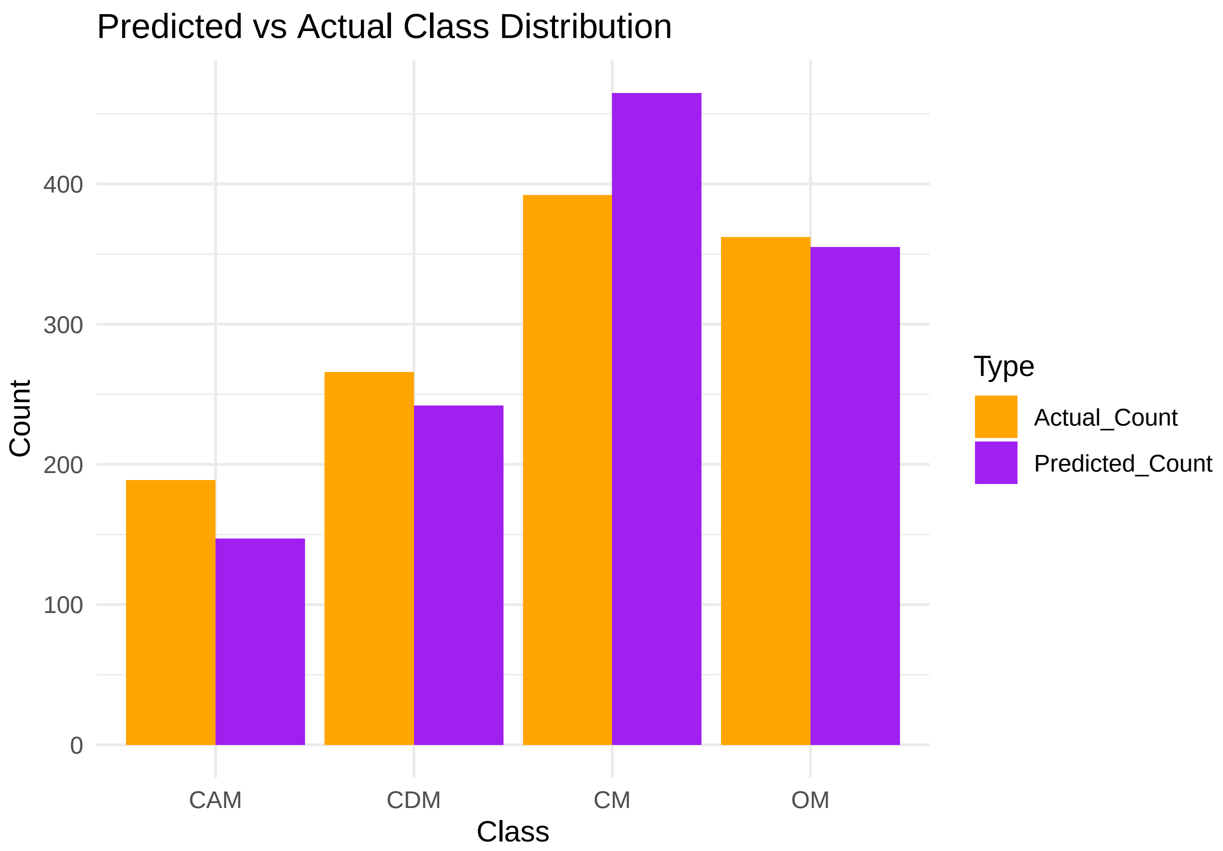
This visualization shows the error rate for each class.One reason why I decided to use this visualization is to compare with the random forest and see if CAM was outputting a high error rate because of some tuning in Random Forest or if it was more related to the CEF, and I can conclude that it most likely was the latter. This again furthers the idea that the appropriate parameters for CAM need to be re-evaluated.One thing that I can notice from the graph is that the error rate stayed consistent through the other Positions, which is a good indicator of the weakness lying mainly on the CAM position.

This violin plot ilustrates the distribution of prediction confidences for each feature. One thing I noticed the three trees had in common was consistency between the features, and so I wanted to see if the usage of each feature was homogenous, and equally distributed amongts all features.This was also to avoid overlooking an outlier that could make us believe that a position is being influenced heavily because of the CEF, when in reallity it was a feature that was not being observed closely enough.

This visual allows us to interpret the predicted vs actual data properly. The results from this visualization are interesting, as in two of the classes, the predicted was higher than the actual, while for the other two classes, the actual was higher than the predicted. The model shows a systematic bias towards CM and OM, which again might be beacuse of the largely overlapping metric requirements from CM with other positions.


Class−wise Error Distribution
Error Rate
0.0
0.2
0.4
0.6
CAM
CDM
CM
OM
Actual Class
Actual
CAM
CDM
CM
OM


Feature Contribution to Prediction Confidence
Prediction Confidence
1.0
0.8
0.6
0.4
DEF
DRI
PAC
PAS
PHY
SHO
Feature
Feature
DEF
DRI
PAC
PAS
PHY
SHO



ID generalisation

Comparisson of Tree Accuracy!

```
## [1] "Accuracy Comparisson for Tree-Based Models!:"
```

```
## [1] "1.Single Decission Tree Accuracy: 63.69%"
```

```
## [1] "2.Random Forest Accuracy: 68.16%"
```

```
## [1] "3.Boosted Tree Accuracy: 67.00%"
```

```
## [1] "Best Model: Random Forest"
```

While the Random Forest is the most accurate, this is a very close result between the Random Forest and the Boosted Tree!

OOD generalisation

Concept shift: For the concept shift, the idea is to change the conditional distribution of the outcome, without disturbing the distribution of the predictor values (or the metrics). In order to do this, we can set conditionals that will lead to a worse and better OOD accuracy.

Worse Concept Shift: For the worse concept shift, we would mutate the dataset and setting a condition. With the runif() function, we are selecting 25% of CM players who have a Dribbling higher than 75. The reason why this may not be accurate is because of the normal distribution curve. As mentioned, the difference between a 75 dribbling, an 80 dribbling, and a 85 dribbling, is not linear, as it is much harder to improve once you are at the top of the level (hence the reason why no players in EA FC have a 99 stat despite being consider the best of all time). Hence, since the average dribbling is around 75, we can see how we are proportionally taking more players than necessary with the CM and 75+ dribbling archetypes, incorrectly labeling in most cases as OM. For this reason, playing with this 75 dribbling shows how when setting 65 as the filter, the model does significantly worse, when selecting 75 as the filter, the model does slightly worse, and when selecting 85 as the filter, the model actually does slightly better. This is because the density of players with statistics are normally distributed, and it is much harder for a player to go in a year from a 85 dribbling to a 90 dribbling, than a player to go from a 55 dribbling to a 60 dribbling.It is important to also connect this to the variable importance, which can be futher explained when attempting to do a better accuracy OOD.

Better Concept Shift: For the better concept shift, we must directly reference the variable importance from the random forest. Firstly, as we understand from the weights assigned in our CEF, the tactical requirement for different position varies between positions. There have been an exaggerated amount of slow (low PAC) OM's in the past decade, where they are more recognized for their through ball passing and ability to quickly transition to a counter attack with a quick vision. However, a CDM that has a low defense statistic would essentially be a huge liability for a team, as usually the CDM's are the first line of defense for a team. For this reason, despite imitating the metrics in the last example, we can see that DEF has a greater effect on the accuracy, and it can make the model more accurate because a CM with such a high defensive stat would most likely than not be considered for the CDM position. It is here where we truly understand the importance of utilizing real-world knowledge in certain data-sets, as complex real-world problems usually include situations like these where the impact of each metric is much larger than it may seem in hindsight.

```
## [1] "Random Forest ID Accuracy: 68.16%"

## [1] "Random Forest Concept Shift Accuracy (Worse Scenario): 67.74%"

## [1] "Random Forest Concept Shift Accuracy (Better Scenario): 68.24%"
```

Covariate shift

For the covariate shift, the idea is to change the distribution of the predictor variables while keeping the outcome the same. By changing the distribution of the features, the idea is to alter the features themselves, to change the accuracy of the model.

Covariate Worse: For the covariate worse scenario, we've discussed throughout the project the importance of evaluating the impact of a feature on a class. In this case, we are going to assess the impact of certain features on the center-midfield (CM) position. One of the problems with misclassification of CMs arises from what we call the "jack of all trades" features. A CM is typically a well-rounded player, but the issue occurs when a CM is too good in certain areas.For example, consider the top 5% of CM players. Their shooting or pace is most likely better than players from lower divisions. This would mean that, despite a player from a lower division having a strong defense (relative to other players in that division), a highly rated CM from the top division of England will likely have a better overall defense than a lower-rated CDM, even if their defense is lower relative to their other statistics.This issue would be more pronounced with a larger dataset, where we could normalize and create a more realistic Comparative Effectiveness Factor (CEF). However, with a smaller dataset, this could result in niche exceptions that complicate training and reduce model accuracy.Let's explore this concept through covariate shift. If we increase the statistics of these lower-ranked players without standardizing them relative to other stats, we're likely to see a decrease in model accuracy. As mentioned, most players with 80+ in a given stat are exceptional in that area. By applying a filter of at least 75 in defense, pace, and shooting—and increasing each by 10 (which are key for CDM, OM, and CAM positions, respectively)—we create confusion for the model.Since high statistics in these fields would more likely classify players as CDM, OM, or CAM, a highly rated CM with overall high statistics across the board could confuse the model. This is because, while their defense, pace, and shooting may be high relative to their own position, they are still being classified based on their overall stats, which are higher in comparison to the other attributes on their card.

Covariate Better: As mentioned previously, when considering the influence of specific statistics on model accuracy, by analyzing the influenced position directly (CDM,OM,and CAM respectively), we are able to understand how the same process can make the model increase in accuracy contrary to what happened in the previous example. As mentioned, a highly rated CM can confuse the model, because them having a highly rated card can confuse the model. Another reason why we should consider introducing a metric to make the data be based on the other statistics (for example a percentage distribution) is to the example presented here. When considering the cards that are influenced by features, we see an increase in model accuracy when the same method is implemented. This is because we are aligning player attributes with the defining characteristics of their position. A CDM with a highly exceptional defense is less likely to be missclassified as a CM, OM, or CAM. This is because of the clear distinguished trait of being a highly defensive player and its effect on a player becoming a CDM. By focusing on the key attributes for each position we are able to reduce the overlap between positions. This analysis helped realize the focus that has to be made in CAM and CM, as the key distinguishing features for both are less clear cut, and so they need to be defined more appropiately.

```
## [1] "Random Forest ID Accuracy: 68.16%"
```

```
## [1] "Random Forest Covariate Shift Accuracy (Better Scenario): 69.73%"
```

```
## [1] "Random Forest Covariate Shift Accuracy (Worse Scenario): 66.25%"
```