INST737
Team 2
Dr. Frías Martínez
11/28/23

Milestone 3

Additional Data Collection

Based on the reviews and suggestions from the critical feedback from our peers, we decided that the amount of data we had was sufficient to perform our tests, thus we did not add or include any new data. Our data currently contains League of Legends data from patches 13.11, 13.12, and 13.13. Depending on the methods and models used, we had to remove the categorical variables to provide results to the tests.

Question 1. SVMs

For the SVM model, we chose to utilize the variables Pick %, Ban % and Win Percent based off of previous tests in other milestones. These were the most influential variables when predicting score. As a result, this was our approach for our SVM models. For our first SVM, we chose to use the "vanilladot" kernel.

```
# We use the three main variables that have the most influence along with the vanilla dot kernel first
LoL_regression <- ksvm(Score~`Pick %`+ `Ban %` + `Win %`, data = LoL_train, kernel = "vanilladot")
```

After running the results of this model, these were our results. The main value we will highlight here is that the training error is 0.251077.

```
Linear (vanilla) kernel function.

Number of Support Vectors : 470

Objective Function Value : -160.8055
Training error : 0.251077
```

Now to compare these results, we utilized the “rbf” kernel to further improve our previous model.

```
# Opted for the RBF KERNAL to compare to the vanilla kernel to improve the model
LoL_regression_rbf <- ksvm(Score~`Pick %`+ `Ban %` + `Win %`, data = LoL_train, kernel = "rbf")
```

After running this model, we were able to improve our model’s training error to 0.138372, which was a 0.112705 decrease indicating an improvement in our SVM model.

```
Number of Support Vectors : 494

Objective Function Value : -135.9976
Training error : 0.138372
```

For our RMSE values, even though our training error decreased and model improved, our RMSE value increased by .294658. This indicates that the performance of the model has decreased.

```
> rmse(LoL_predictions, LoL_test$Score)
[1] 6.626107
> rmse(LoL_predictions_rbf, LoL_test$Score)
[1] 6.920765
```

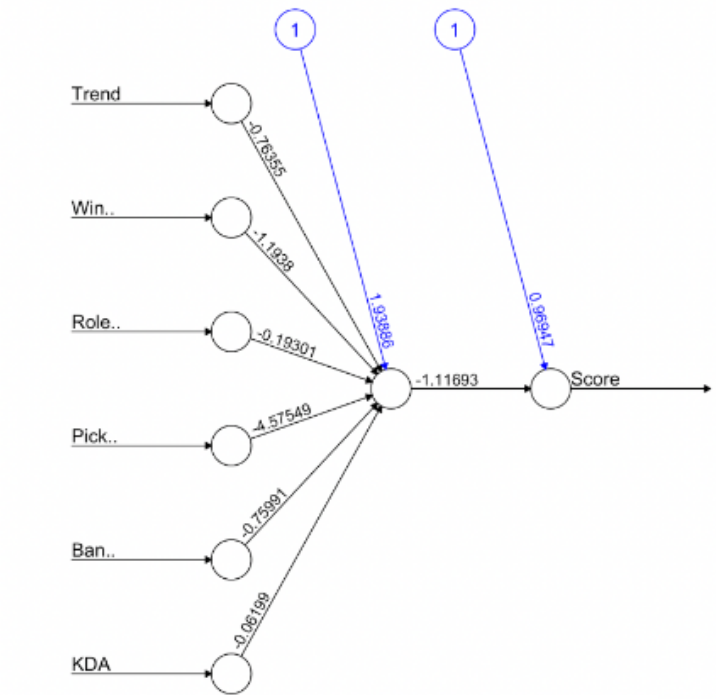
One possible reason for this increase in the RMSE values is that the RMSE value is better fitted for the linear “vanilladot” kernel rather than the “rbf” kernel. This is because the “rbf” kernel is more closely fitted to a gaussian curve which would lead to higher distance in the RMSE.

```
> cor(predicted_Score,LoL_test$Score)
          [,1]
[1,] 0.8741886
```

Question 2. Neural Networks

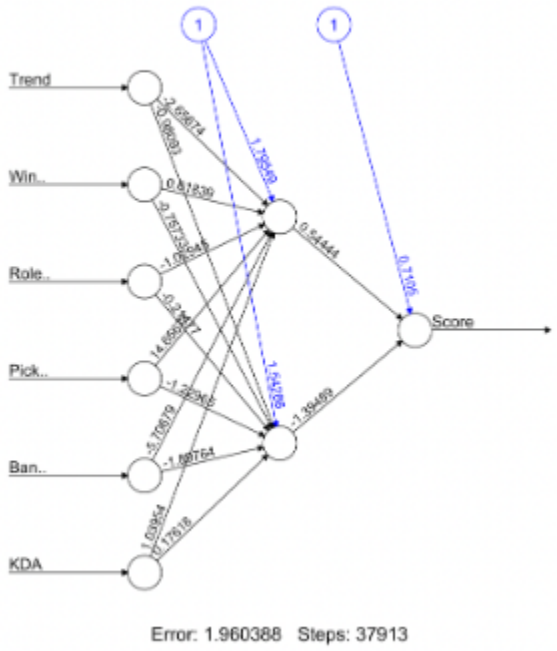
For our neural network models, we based them off our research question revolving around the variable “Score”. For our first neural network, we first tested it without any hidden layers to see the performance/correlation. After examining the correlation of our predicted Score and actual values with this first model, we had a correlation of 0.8741886. We will now test and compare this value with different models with additional hidden layers. It is important to note that there was 71 neurons in each neural network for every variable.

W/O Hidden Layers



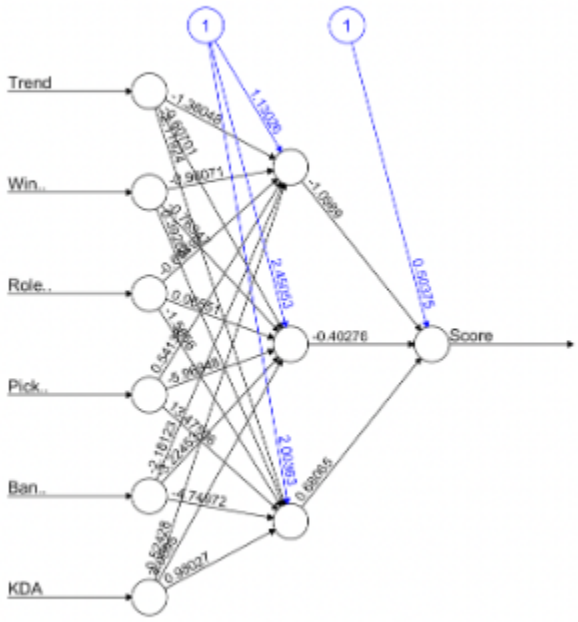
With implementation of 2 hidden layers within the model, we had a correlation of 0.8826897. This was .0085011 higher than our first model without any hidden layers.

With 2 Hidden Layers



To further test and find a higher correlation, we tested 3 hidden layers, which gave us a correlation of 0.8947565, which was 0.0205679 increase compared to our base neural network model.

With 3 Hidden Layers

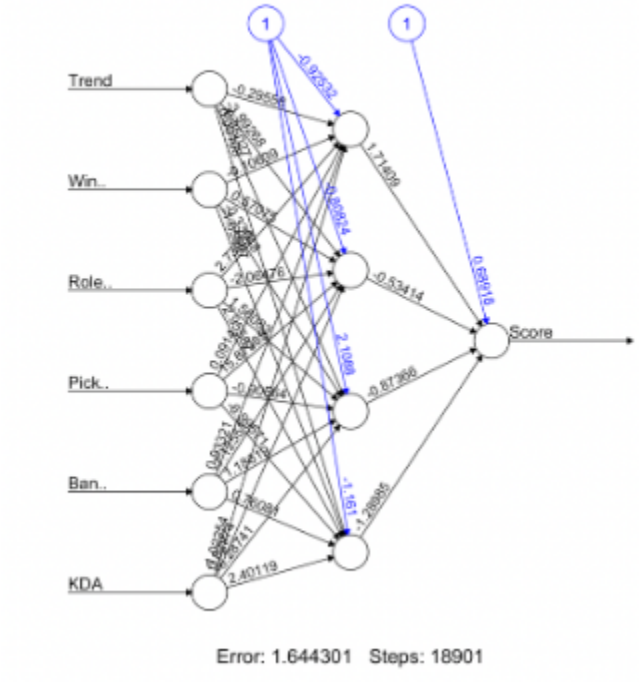



```
> cor(predicted_scorehidden2,LoL_test$Score)
          [,1]
[1,] 0.8947565
> cor(predicted_scorehidden3,LoL_test$Score)
          [,1]
[1,] 0.8735664
```

Finally, we increased our hidden layer value to 4, and had a correlation of 0.8735664. This was where we finally saw a decrease in correlation value likely due to the model overfitting to compensate for data points that appear as outliers.

This resulted in 3 hidden layers giving us the highest correlation value, and the model we will be further exploring.

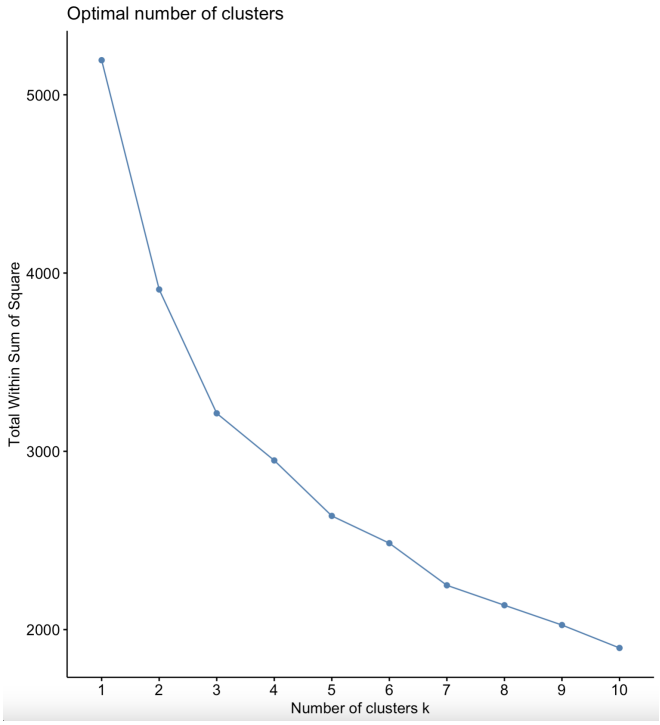
With 4 Hidden Layers

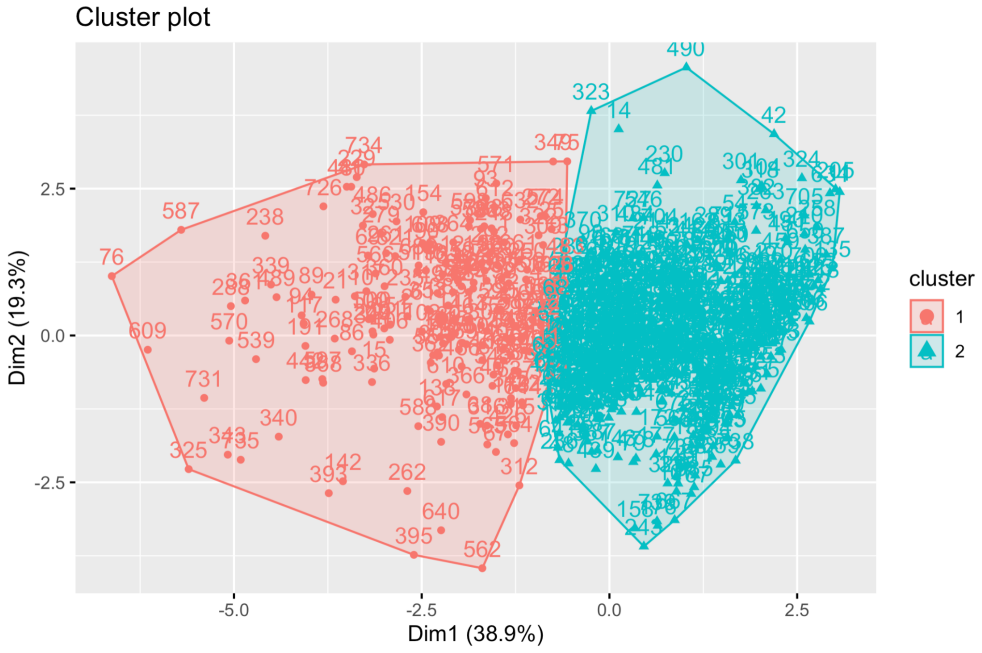


INST737
Team 2
Dr. Frías Martínez
11/28/23

Question 3. Clustering

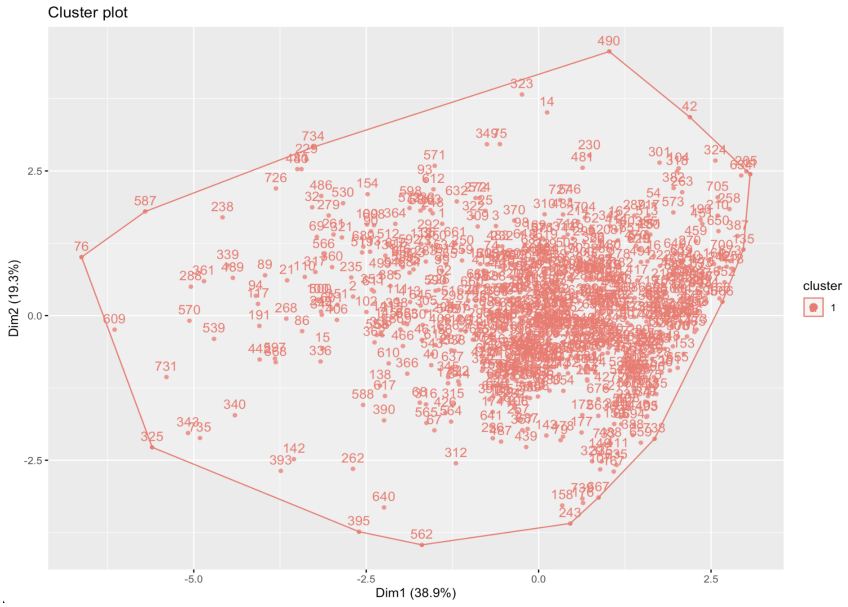
For our k-means model, cluster 1 in red, has 555 points contained in the cluster and for cluster 2, there are 188. After analyzing the elbow graph, we believe that 2 clusters is optimal for this model and our data. Since the elbow graph can have some subjectivity with interpretation, we tested other cluster sizes as well to see if there would be a better amount of clusters. With extensive testing, there was too much overlap when the clusters exceeded 2 which we considered as not optimal.



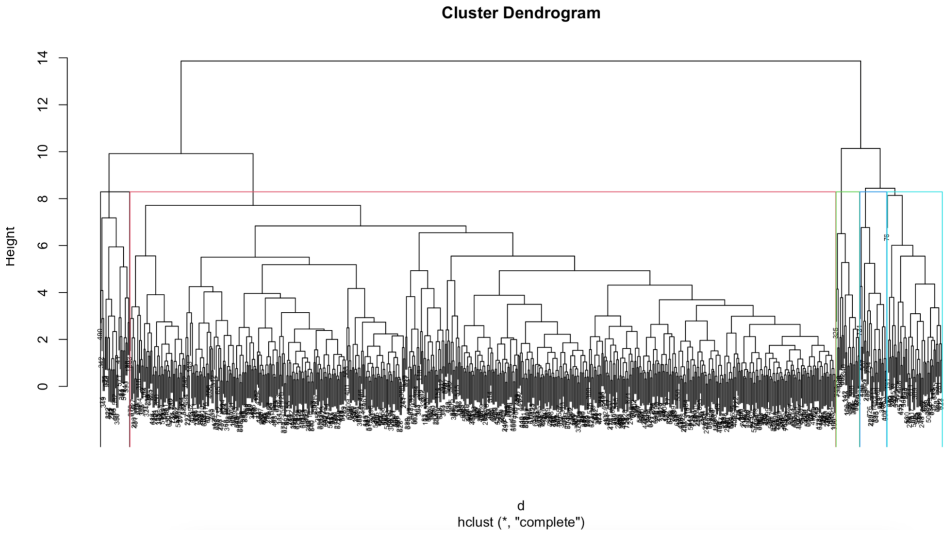


INST737
Team 2
Dr. Frías Martínez
11/28/23

With the Density based clustering model, our data had only one cluster which had all 743 values and also had the same seed value as a result. We can support this result given that most of the values are within a similar range. Having removed all categorical variables, a majority of the data points are within the 0-1 scale given that 3 of the variables included are percent based variables in a decimal form. Something else we can note is there are no outliers based on the cluster created, but given the distance away from the main cluster of points, this is something that could be considered.



After we removed the classifier variables from our data, the model was able to determine that there were five major clusters based on the roles having similar combinations of our independent variables as seen in our table function. Our dendrogram shows these five clusters and their spacing. We assume that the model determined these five clusters from our dataset, reflecting the five main roles that all champions in League of Legends can fall within, shown in our "Role" column as Top, Mid, Jungle, Support, and ADC.




```
cl_members
  1   2   3   4   5
 49 623  26  24  21
```

INST737
Team 2
Dr. Frías Martínez
11/28/23

Question 4. Comparative Analysis

For our comparative analysis, we compared a neural network, random forest and SVM model. We ran these 3 models and had both graphs and a summary to better understand the comparison.

```
# train the Neural Network
set.seed(7)
modelNN <- train(Score~., data = Lol_final, method ="neuralnet", trControl = control)

# Train the RF model
set.seed(7)
modelRF <- train(Score~., data = Lol_final, method ="rf", trControl = control, verbose = FALSE)
# train the SVM model
set.seed(7)
modelSVM <- train(Score~., data = Lol_final, method ="svmRadial", trControl = control)

#collect resamples
results <- resamples(list(NN = modelNN, RF = modelRF, SVM = modelSVM))
```

Regarding the mean absolute error (MAE), found on the screenshot above, the Neural Network received an average of 10.447, while in comparison, the Random Forest received an average of 3.368, and the Support Vector Machine received an average of 3.838. This indicates that the Neural Network model did not predict as well as the Random Forest model or the Support Vector Machine.

Regarding the root-mean-square deviation (RMSE), found on the screenshot above, the Neural Network received an average of 13.988, while in comparison, the Random Forest received an average of 4.760, and the Support Vector Machine received an average of 5.840. This indicates that the Neural Network model did not predict as well as the Random Forest model or the Support Vector Machine. In addition, the Neural Network had the highest range in values, indicating that this

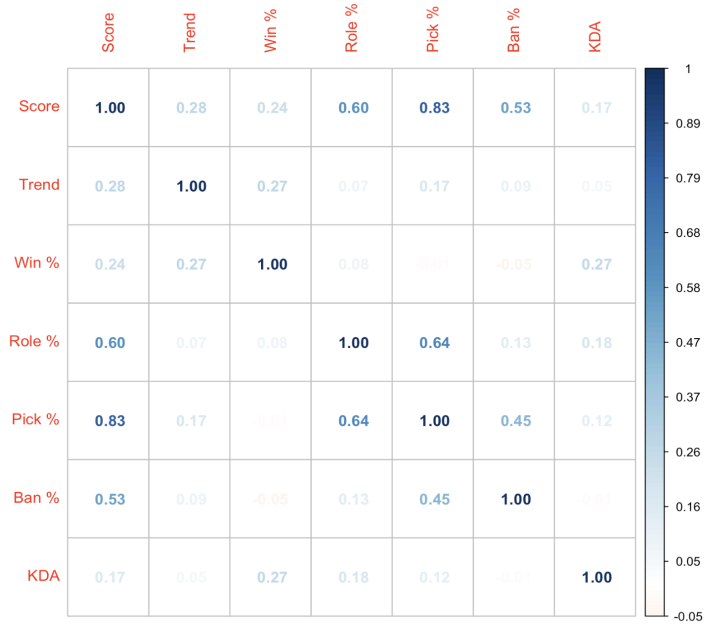
```
Call:
summary.resamples(object = results)

Models: NN, RF, SVM
Number of resamples: 30

MAE
        Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
NN  10.325373 10.386283 10.447193 10.447193 10.508104 10.569014   28
RF   2.724767  3.146771  3.282141  3.368117  3.584759  4.142755    0
SVM  3.206884  3.526090  3.741140  3.838147  4.126164  5.003703    0

RMSE
        Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
NN  13.782350 13.885274 13.988198 13.988198 14.091122 14.194046   28
RF   3.748860  4.324111  4.645988  4.760578  5.185825  6.235736    0
SVM  4.797855  5.314718  5.797670  5.840603  6.271315  7.447842    0

Rsquared
         Min.    1st Qu.     Median       Mean    3rd Qu.       Max. NA's
NN  0.03400723 0.04141444 0.04882166 0.04882166 0.05622887 0.06363608   28
RF  0.80303116 0.87597228 0.88301892 0.88734732 0.90842871 0.93744110    0
SVM 0.76551640 0.79825466 0.82762515 0.83233342 0.86247930 0.92571829    0
```

model was not a good fit for our dataset.

Regarding the R squared values found on the screenshot above, the Neural Network received an average of 0.0488, while in comparison, the Random Forest received an average of 0.8873, and the Support Vector Machine received an average of 0.8323. This indicates that the Neural Network model did not predict as well as the Random Forest model or the Support Vector Machine. In addition, the Random Forest and Support Vector Machine models are a good fit, as their R squared value sits closer to 1 indicating that they have a stronger correlation.

Question 5. Feature Selection

Select three models from Milestones 2 or 3 (linear/logistic regression, Naïve Bayes, Decision Trees, SVM or Neural Networks) and apply three different types of feature selection: filter, wrapper or embedding, one per model. Report results focusing on improvements in performance (or lack of) due to feature selection.



For our first feature selection, we used a filter for our SVM. With the help of the correlation graph, we decided to remove the variable, "Trend" from our model. For a clear comparison, we kept the kernels the same.

```
# Now lets make default model
model1 <- ksvm(log(Score)~`Pick %`+ `Ban %` + `Win %` + Trend, data = train, kernel = "rbf")

model1

# Kept kernel the same and removed trend
model2 <- ksvm(log(Score)~`Pick %`+  `Ban %`+ `Win %`, data = train, kernel = "rbf")

model2
```

From model one, these were our results:

```
> model1
Support Vector Machine object of class "ksvm"

SV type: eps-svr  (regression)
 parameter : epsilon = 0.1  cost C = 1

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  0.583420326415571

Number of Support Vectors : 440

Objective Function Value : -126.7206
Training error : 0.126765
```

With a training error of a 0.12675, we will try to improve our model and make it less complex with the removal of "Trend"

In model 2, with the removal of variable "Trend" our training error went up 0.033531. Even though this went up, by removing a variable that does not have a large influence on the model, we have made this a less complex model.

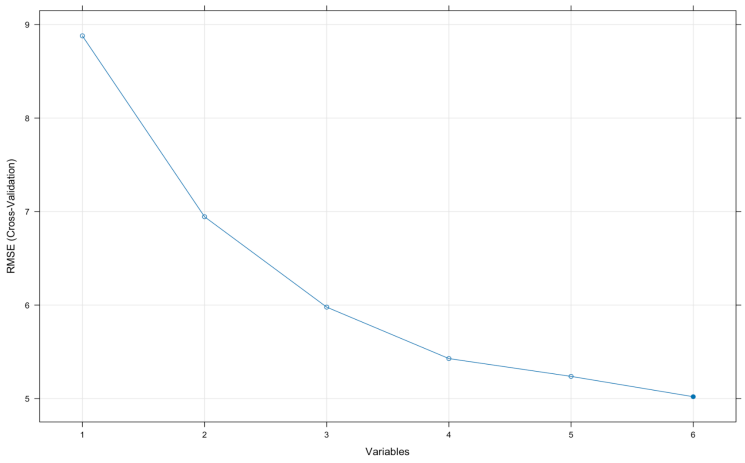
```
> model2
Support Vector Machine object of class "ksvm"

SV type: eps-svr  (regression)
 parameter : epsilon = 0.1  cost C = 1

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  0.853523514998443

Number of Support Vectors : 433

Objective Function Value : -130.296
Training error : 0.160296
```

For the wrapper feature selection method we used the linear regression model. We used the stepwise algorithm and chose the "forward" direction. Here is the output which outlines our intercept and coefficients of each variable.

```
Call:
lm(formula = Score ~ `Pick %` + `Win %` + `Ban %` + `Role %` +
    Trend, data = trainData)

Coefficients:
(Intercept)     `Pick %`      `Win %`      `Ban %`     `Role %`        Trend
   -61.5414     245.9994     188.3321      51.9263       7.5239       0.1917
```

As we can see from our output, the inclusion of all variables creates the best model.

For our third feature selection, we attempted the hybrid option with RFE. Given that all our data is continuous in nature, we were able to get a graph that shows cross validation based on the RMSE values. Based on the graph, 6 variables allowed for the lowest RMSE value.



When we run the predictors function, we get the 6 variables that creates the best results:

```
> predictors(results)
[1] "Pick %" "Win %"  "KDA"    "Ban %"  "Role %" "Trend"
```

Question 6. Extra Credit

Discuss potential ethical issues in the research questions you have proposed for your project. As covered in Class 10 (Ethical Data Science part), delve into concerns around data collection, variable manipulation and model evaluation.

While the League of Legends data may not have been collected from a primary source, it is important to trace the data and take a look into how the data has been collected, manipulated, and

compiled into the Kaggle resource that we had received it as. One point that we wanted to touch on was how this specific data collection and usage method would potentially not require an IRB or an Institutional Review Board, as the data is being collected second hand and from a site where players knowingly opted in to having their data collected. Our project also does not require hands-on testing with users and all data involved maintains participant anonymity. If our study was to go further into a higher level of research, it would be ethically responsible to reach out to the host of the site and ask for permission to use the data. An example where an IRB may be needed would be if we were to study a collegiate esports team's in-game performance. The IRB would be able to verify that we received informed consent from the participants, maintain that we are providing the best welfare for the participants, and that it follows all of the university specific standards.

In terms of demographic considerations, the demographic makeup of League players has shifted over time, possibly due to the release of more general entertainment content, such as the popularization of the Netflix series *Arcane* which is based in the game universe of League of Legends. Before this era, most League players tended to be male, and thus now with a more gender inclusive audience, the data collected from League of Legends would have changed, due to the addition of the new players. This brings about the interesting topic of how variables may or may not have been manipulated to reveal certain aspects about the general community's play style and/or behaviors, which could be creating a bias, as certain groups of people may have been left out.

Another area to note within data collection is that players are functionally segmented into regions globally by Riot Games for reasons relating to server capacity and language barriers. It is important to consider when collecting data which regions are represented. Sometimes, it is intentional for a researcher to work within a specific demographic region or make distinctions between demographic regions such as comparing stats between say North American Players and European Players. These comparisons can be useful when analyzing how players within different regions develop in game strategies. Therefore, it is important to be mindful of how different races and ethnicities are discussed and what conclusions are drawn from your models when analyzing individual player performance between demographic regions.

INST737
Team 2
Dr. Frías Martínez
11/28/23

Contributions:

Report:

Question 1:

- Additional Data Collection: Paris
- SVMs
    - Formatting: Paris
    - SVM values and explanation: Saran

Question 2:

- Neural Networks
    - Formatting: Paris
    - Hidden layers values and explanation: Saran

Question 3:

- Clustering
    - Formatting: Paris
    - Cluster values and explanation: Saran
    - Dendrogram values and explanation: Michaela

Question 4:

- Comparative Analysis
    - Formatting: Paris
    - Summary and explanation : Paris, Michaela

Question 5:

- Feature Selection
    - Formatting: Paris
    - Selection values and explanation: Saran

Question 6:

- Extra Credit
    - Written explanations: Michaela, Paris

- Contributions Section: Paris

INST737
Team 2
Dr. Frías Martínez
11/28/23

Code:

- R Setup: Saran
- Directory Routing: Saran
- Reading dataset into R: Saran
- SVM kernel choices and results: Saran
    - Soundboard for troubleshooting: Michaela, Paris
- Neural Networks
    - Code Setup: Saran
    - Visualizations : Saran
    - Hidden layers: Saran
- Clustering: Saran
    - Soundboard for interpretation: Paris, Michaela
- MAE, RMSE, Rsquared summaries: Saran
- Feature selection: Saran
    - Soundboard for feature selection decision making: Michaela, Paris
- Readme: Saran

Presentation:

Presentation Slides:

- Slide setup : Paris
- Slide formatting: Michaela
- Talking:
    - Slides 3, 8, 10, 11, 13 : Paris
    - Slides 2, 5, 6, 7 : Saran
    - Slides 1,4, 9, 14 : Michaela