College Football Game Attendance

Team 2: Akshay Gandhi, Amber Rehman, Zixiao Wang, Suyang Xu

University of Connecticut

OPIM 5604: Predictive Modeling

Professor Jennifer Eigo

April 21, 2021

**Table of Contents:**

[Executive Summary](#_iw7gdz4a7whk) 3

[Problem Statement](#_b6qvo4awoekw) 3

[Methodology](#_ab8xgoneiq2r) 4

[Sample](#_nv7al8819jxu) 4

[Explore](#_p8116admaery) 4

[Modify](#_3kpn719zemaw) 5

[Models](#_d6rymrofcrhc) 7

[Linear Regression Model](#_cq69ba8jq9ks) 7

[Decision Tree](#_9skmk2e91gsu) 8

[Bootstrap Forest](#_12luabn4osam) 8

[Boosted Tree Model](#_kfwmxs63djdf) 9

[Neural Network Model](#_wid64gape4e9) 9

[K Nearest Neighbor Model](#_2trea6qczjo) 10

[Naive Bayes Model](#_kbybeyqta4rx)  11

[Models Assessment](#_ilgmufd3i06) 11

[Result](#_ksa1hf8suwrd) 12

[Conclusion and Recommendations](#_d80t7zyy60eo) 13

[References](#_n7k5xt4utlqs) 14

Appendix 15

# **Executive Summary:**

Our project is about college football games and the dataset that we worked with contains 25 columns, essentially exploring the data consisting of the information about each game, such as date, time, team, opponent, current wins/losses, etc. This dataset gives insight into the specific information of each game and how it would affect the attendance of that particular game. With that, the target variable in this case is attendance. We carried out many models, such as a linear regression model, decision tree models, neural networks, KNN (K Nearest Neighbor), naïve bayes, and ensemble models. After conducting these models, we were able to determine which model was the best at predicting the most impactful variables on attendance to college football games. This information would be of value to colleges since it gives them the opportunity to improve upon their decision making in regards to their football games so that they can bring in more people in the future.

# **Problem Statement:**

College football has recently seen a decrease in attendance in the stands at games. This negative trend is quite worrisome to colleges due to the fact that college football brings in a great deal of revenue to schools. Teams make necessary adjustments to their stadiums in hopes for reeling in more people to their games. With that, it is vital to look at data patterns of college football games in order to determine which variables have the greatest influence on game attendance overall.

# **Methodology:**

From the dataset of college football games, we are going to build a predictive model to find out which variants have the strongest influence on attendance and if there are any patterns that can be seen by the different variables. Also, we will be analyzing if there is any difference in the variants of low attendance and high attendance games.

# **Sample:**

# When we first looked at the 6672 rows and 25 columns in the football game dataset, it was obvious that there was plenty of data for modeling. The first discovery was that some of the columns had far too many different values or levels that could be used as categorical variables so we needed to convert some of them based on the data available in that column. The dataset did not have any missing values but we had to check for outliers to determine if we needed to factor them into our modeling.

# **Explore:**

We examined the dataset and used the boxplot and distribution features to see if any categorical variables had missing values or outliers. Similarly, we used JMP's explore outliers and explore missing values features to see whether there were any missing values or outliers for continuous variables [figure 1 and figure 2]. There were no missing values or any major outliers which needed to be removed. Further, Multivariate analysis is done to understand the correlation between the columns [figure 3]. The attendance was strongly correlated with stadium capacity and fill rate. However, fill rate was excluded due to the fact that attendance is derived from fill rate and since attendance is our target variable we disregard it. Surprisingly, there was not much of an impact on attendance due to the weather condition [figure 4]. It is also important to note that multivariate only reflects variables that are continuous. We then started exploring the data using contingency tables and data visualization using a distribution function in JMP to have a better understanding of the correlation of the categorical variables to find if there were any patterns for attendance in the stadium. For highest attendance i.e., attendance more than 80000 it was always broadcasted on TV channels. The variable that had the most impact on attendance was the team which is playing the game. Also, the other factor which was necessary to explore is the stadium capacity with respect to attendance. If the stadium capacity is 50000 and the attendance is also 50000 then the stadium is at its full capacity.

# **Modify:**

One of the major problems we faced with our dataset during the modification process was the columns having too many levels to deal with in a single column to understand whether it had any association with attendance or not. We started by deciding which columns needed to be recoded and excluded. We found that some of the columns were important and that this has a direct effect on attendance based on our examination. However, there are a few columns that needed to be recoded or removed because they did not have much correlation with attendance. The column which we modified or excluded was Opponent since after running the distribution study [figure 5], we found that it had far too many levels to deal with and did not have much relation with attendance. Since we also have opponent rank to deal with, it would provide sufficient detail. Thus, we excluded the Opponents column. We chose to exclude date and time because date has already been divided into three columns and time contains too many specific details to work with and has no association with attendance. Site is being recoded because it contains useful knowledge, but the levels are too difficult. To simplify things, we merged all of the categories with less than 50 records into one, renamed the column site(pre-process) [Figure 6], then hid and excluded the original Site column. Similar to the site, we recoded TV, combining all of the categories with less than 50 records into one. The Results column, which included both the match score and the win/loss (e.g., W 45-56), was recoded because it was in a categorical setting and had too many levels. We agreed to retain just the W and L (Win or Loss) columns, delete the numerical values to simplify things, and rename the new column Result(pre-process), while excluding and hiding the original Result column. Further, we recoded columns PRCP and SNOW as most of those rows are 0, we choose to change those two columns by changing values into if rain/snow then 1 else 0 to reduce the complexity and make it clearer for explanation for our model. The new column name is PRCP(pre-process) and SNOW(pre-process) to which we then excluded and hid the original columns PRCP and SNOW. We chose to exclude and hide the column SNWD, because this column is an extension from the original column snow and we have about 6600 rows and only less than 100 rows are not 0 so it is not really useful to build a model. We also created another column to put Stadium Capacity into categories in case the model needed more nominal variables for predictions. We made 10 levels, starting with 10000 and each increased 10000 to the max of 110000 which covered all the continuous values in the original column. For our target valuable attendance, we also created another column to make it into nominal values just in case some of the models could only work with nominal values. We can still gain a general idea of the range of the predicted attendance for the game. For the new column, we made 12 levels, ranging from 0 to 120000 each increased by 10000, so it would cover all the values in the continued attendance column[figure 7]. After the preprocessing, we now have 19 real column/predictors (not counting 1 extra stadium capacity) and 6672 rows to build our model. We created a validation column that contains 0.6 training, 0.2 validation and 0.2 test, and now the dataset is ready to build.

# **Models:**

# **Linear Regression Model:**

We started to build a linear regression model first [figure 8 ]. First, we put all the predictors into the model. We decided that the fill rate should not be included in the model because fill rate was actually calculated by using attendance, and we are not able to have values for fill rate unless we know the attendance. The R squared is 0.9498 for training, 0.9381 for validation, and 0.9376 for test, and the RASE is 5644, 6264.0 and 6292 respectively, which means this model performs pretty well without overfitting the training set [figure 9]. Since we used all the variables in this model, we should remove some unimportant variables to make the model simpler.

After finding out the balance of number of variables and the amount of errors, the best model we have is with nine variables: team, current losses, stadium capacity, site(pre-processed), TV(pre-processed), conference, Month, Rank, opponent rank [figure 10]. The RASEs are at the lowest with as few variables as possible; for training we have 5672, 6265 for validation, and 6196 for test. We also take a look at our residuals [figure 11]. For the mean residuals, we have almost 0 for training, -116 for validation, and -98 for test. For median residuals, we have 51 for training, -83 for validation, and -190 for test.

# **Decision Tree:**

For our decision tree model, we chose to keep all the variables as predictors but switch into using the nominal attendance we created for decision trees because theoretically decision trees only work with nominal target variables and we can use the confusion matrix to better analyze the accuracy of the correct range the attendance falls into. We first use the best split feature to let it run and find the technical best split, which is 49 splits. However we believe 49 splits are highly complex for a model, and the misclassification rate for testing is about 43%, which is not really low [figure 12]. We decided to manually test out the number of splits to find the least number of splits that can provide a highest R squared and least percentage of misclassification rate. After balancing those values, we figured the best would be having 10 splits with the R squared for the test at 51% and the misclassification rate at 46%. This means the total accuracy is 54%, which is above the baseline of 8.3% with 12 different categories.

# **Bootstrap Forest:**

For our bootstrap forest model, because for the regular decision tree model JMP recommended 49 splits and the split curve shows greatest change at around 9 to 10 splits, so we decide to do the same as the decision tree model, put every possible factors into x, and the nominal attendance as the target variable. We used minimum 10 splits per tree and maximum 50 splits per tree, and kept everything else as default [figure 13]. Because bootstrap forest models pull out data randomly for every model, we tried five times to get the model that provides the highest accuracy. For our best result we saved [figure 14] the misclassification rate for the testing portion at 0.39 and the total accuracy at 0.61. This means our bootstrap model performs better than the regular decision tree model we used, and also a lot better than the baseline probability of 0.083 accuracy.

# **Boosted Tree Model:**

Boosted tree models only support categories 0 and 1 as target variables, so we cannot use it for our dataset [figure 15].

# **Neural Network Model:**

We used the nine variables we used for the linear regression model to build our neural network model because it cannot pick the important variables automatically. Also, looking at the correlation of all the variables with respect to attendance we get the highest correlation in almost the same variable that we got from the linear regression.

Once the predictors are finalized, we checked the model with the default number of nodes in each activation type i.e. tanh =3. We tried using the combination of tanh, linear and gaussian activation function as the gaussian activation function is used for normalization and it is also effective for predicting continuous variables. We tried increasing the number of nodes to check the model’s RMSE changes. The reason for increasing the nodes is the ability of the model to increase in complexity as increasing the number of nodes allows the model to learn the training dataset much better than the previous model. Furthermore, the addition of more layers leads to the discovery of more patterns of resemblance, making any study of new data more valuable. After comparing all the models we found that the first model performs the best given that it has the lowest RMSE, highest R Squared and lowest residuals in validation. [figure 16, 17, 18]. The accuracy for predicting the values is approximately 96%.

# **K Nearest Neighbor Model:**

For the KNN model we performed the predictor screening [figure 19] and fit model along with the help of initial analysis to choose the best predictor for KNN modeling. These variables include Team, Stadium Capacity(box data), Conference, Site(pre-process), Tailgating, TV and Rank to predict attendance and we discarded the other variables for performing modeling as they do not have much correlation with the attendance and thus would negatively impact the model’s accuracy [figure 20]. We checked the model for what K value the RSquare value is the highest at and if the training, validation, and test data have similar RSquared values at the same K value so that the model is accurately predicted and does not overfit. For K = 9 we get the highest RSquare giving the accuracy of approximately 94% for the predicted attendance value.

**Naive Bayes Model:** For our naive bayes model, we first put all 11 nominal variables into the factors and use the recoded nominal attendance as our target variable [figure 21]. From this contribution table, we can see that the last four variables do not contribute as much so we choose to remove those four variables, keeping the rest of the 7 factors. After we do that, we get a model with misclassification rate of 0.41 for training, 0.46 for validation and 0.45 for test, which means it’s not over fitting the training data [figure 22, 23, 24]. The total accuracy of the test is 55%, which is overall better than the baseline of 8.3% for a 12 category target variable, but it is worse than the bootstrap tree model we used prior.

# **Models Assessment:**

We first take a look at all the models we have. For the continuous target variables, the best model we have currently is the first neural model we tried with three tanH nodes as it has the lowest RASE of 5944. For the model using boxed attendance as target, the best model is the Bootstrap model we saved, with 0.3828 misclassification rate. We took an average of the continuous models together, and found out the RASE of the test portion of the average model is 6960, which is not the lowest RASE of our models [figure 25]. The best model currently is still the neural model and the worst is the decision tree model with RASE of 9432.

Next, we tried to average the nominal target variable models to see if the average of the three models would improve the misclassification rate. Our best currently has a 0.3828 misclassification rate. We also took an average of all three models [figure 26] to see if the average of the three is improved and better than the bootstrap model. For the average model, the misclassification rate is 0.4337, which is worse than the bootstrap model. Because there is not much improvement we can do with these three nominal models, we decide to keep the bootstrap as our best nominal model and focus to improve the continuous models.

We want to take out those continuous models that do not perform as well. We tried different combinations and found out that only keeping the best three models, the first neutral model (the current best) with RASE 5944, linear regression model with RASE 6196, and the KNN prediction model with RASE 6194 would produce the best average model [figure 27]. We can see that the RASE for the average model is 5715, and is about 200 lower than 5944. We believe that to have an improvement of around 200 errors, it’s worth adding up a layer of complexity to create an ensemble model to average our linear regression, KNN, and neural network.

# **Results:**

After assessment of different kinds of models and different combinations of average ensemble models, we found that the best continuous model is the average of three models: the linear model, KNN model and the neural model. The RASE is the lowest of all the models, 5715 in the test portion and it also has the highest R squared, 0.94 in the test portion. For nominal models, our current best is the bootstrap model with 0.38 misclassification rate, which is also pretty good. But after our analysis, we believe that since attendance originally is continuous, it is better to have a continuous prediction result so that each game has its unique prediction of attendance. This would be helpful to identify factors that football teams should consider changing or updating if teams want to attract more audience for their games in the future. The mean residuals for this prediction is 109, meaning that on average this model’s prediction of attendance would be 109 less than the actual attendance. When compared with the mean of the actual attendance of 45311, it is a really small number. Since the average model is the average of the three models, we chose to take a look at the three models and determine variables that have greater influence on attendance. For linear regressions, team, current loss and stadium capacity contributed the most effects [figure 29]. The next ones are site, TV, conference, month, rank, and opponent rank. It makes sense that the more popular a team is, the more likely audiences are to attend their games. Also, it makes sense that the more games a team had lost, the more the audience would be less interested in watching their games. We think it is really interesting that the month of the game contributed a lot to the attendance [figure 30]. We think it is because we have more breaks and holidays in November and December than in months like February and March. Conferences serve a similar function as the team name, TV channels, and playing sites also affect the amount of audience attending the game which makes sense.

For the neural model, since it cannot decide which variables are important automatically, we used the same variables as the linear regression model. For the KNN model, the most important variables are Team, Stadium Capacity(box data), Rate, Conference, Site(pre-process), tailgating, TV and rank, which are similar to the ones we used in linear regression. So overall, those variables are important to consider for football teams to bring in a greater audience.

# **Conclusion and Recommendations:**

Our recommendation for football teams is that if they want to attract more audience to attend their games, they should consider changing factors such as the game site, choosing November or December to hold the game and include more relevant TV channels. Further, having more games shown on television and perhaps on multiple channels would provide the opportunity to increase revenue from broadcasting and advertisements and maybe even greater liking of teams. If teams also want to improve their attendance in the long run, winning more games would increase their own ranking and allow them to gain more popularity overall. Also, considering the inclusion of tailgating at more games would also reel in additional people at games because many enjoy the idea of pre-game partying and entertainment prior to a game. Additionally, there are some games in which the stadium is full majority of the time however the capacity is limited so it disallows the school to gain more revenue because of that. With that, increasing stadium capacity would allow for higher attendance, and thus additional tickets to be sold.

# References:

1. College Football Game <https://www.kaggle.com/jeffgallini/college-football-attendance-2000-to-2018>
2. Data Mining for Business Analytics: Concepts, Techniques, and Applications with JMP Pro (Authors: Galit Shmueli, Peter C. Bruce, Mia L. Stephens, Nitin R. Patel)
3. Class PPT/Lecture Notes of Professor Jennifer Eigo

**Appendix:**

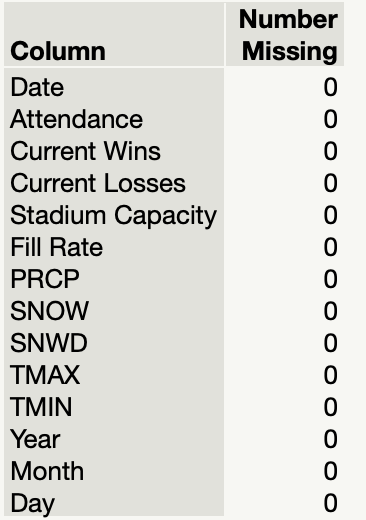


Figure 1(missing values)

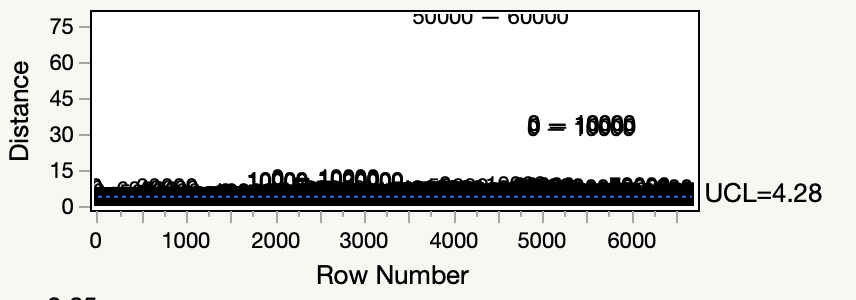


Figure 2 (outliers)

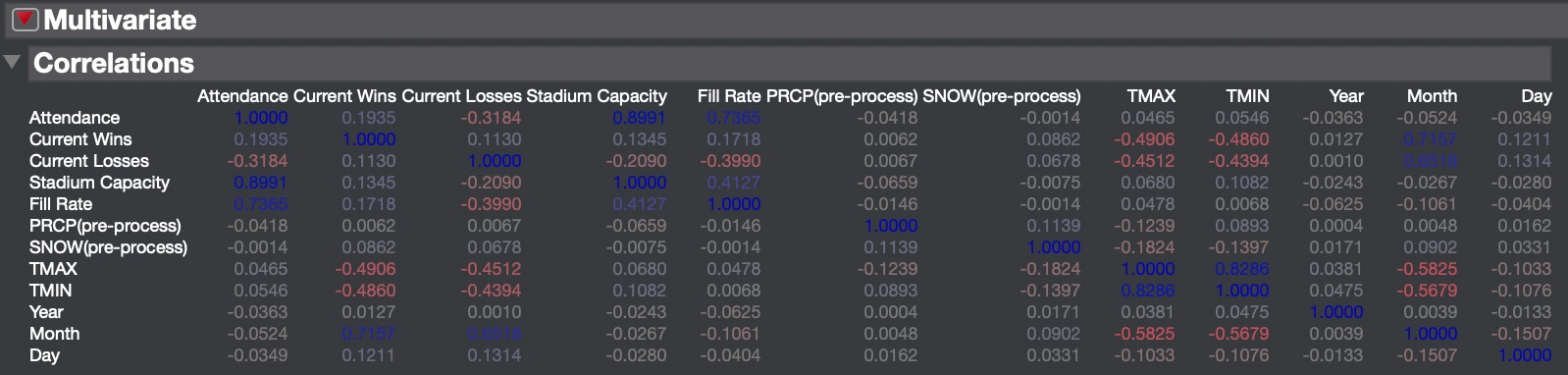


Figure 3 (multivariate analysis)

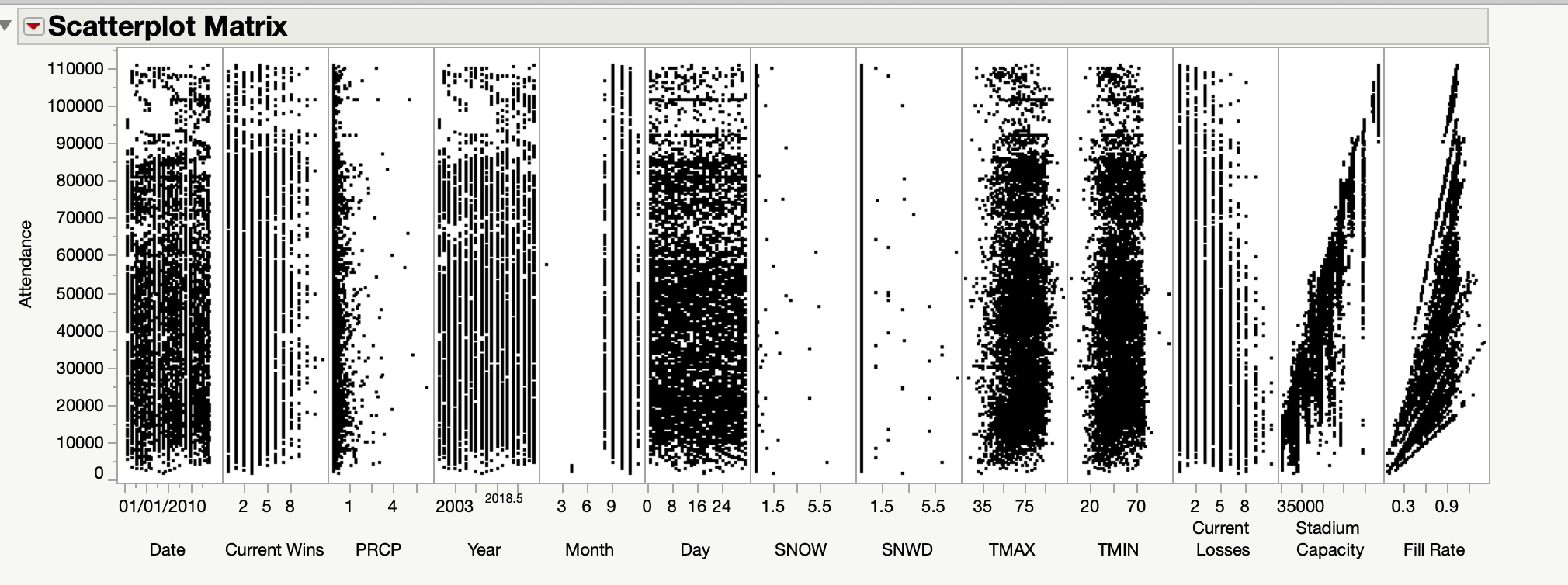


Figure 4 (scatter plot matrix with attendance)

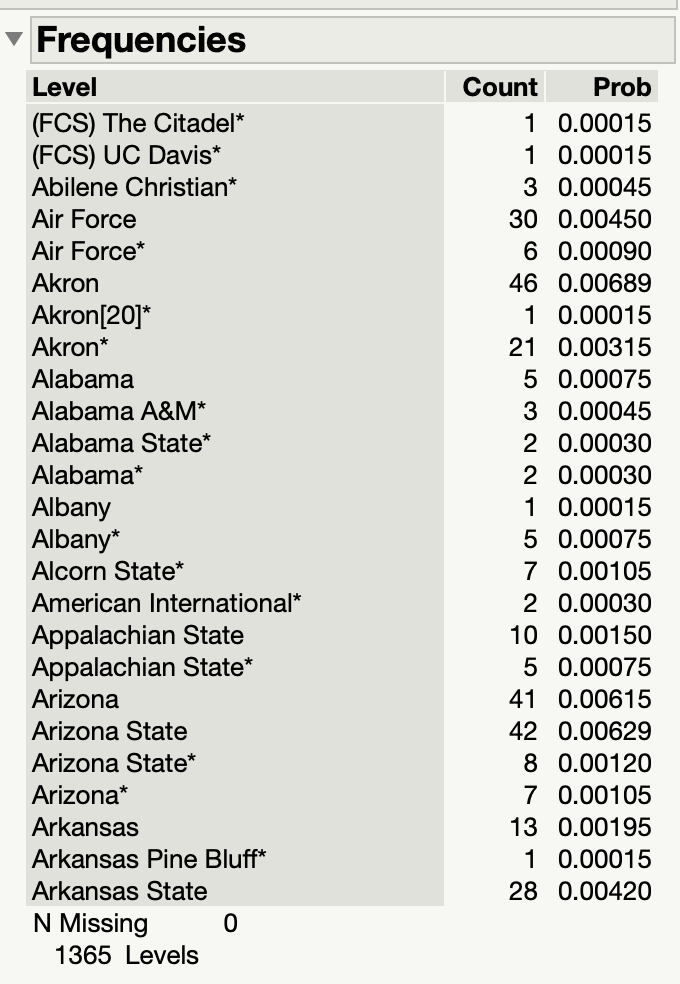


Figure 5 (distribution of opponent)

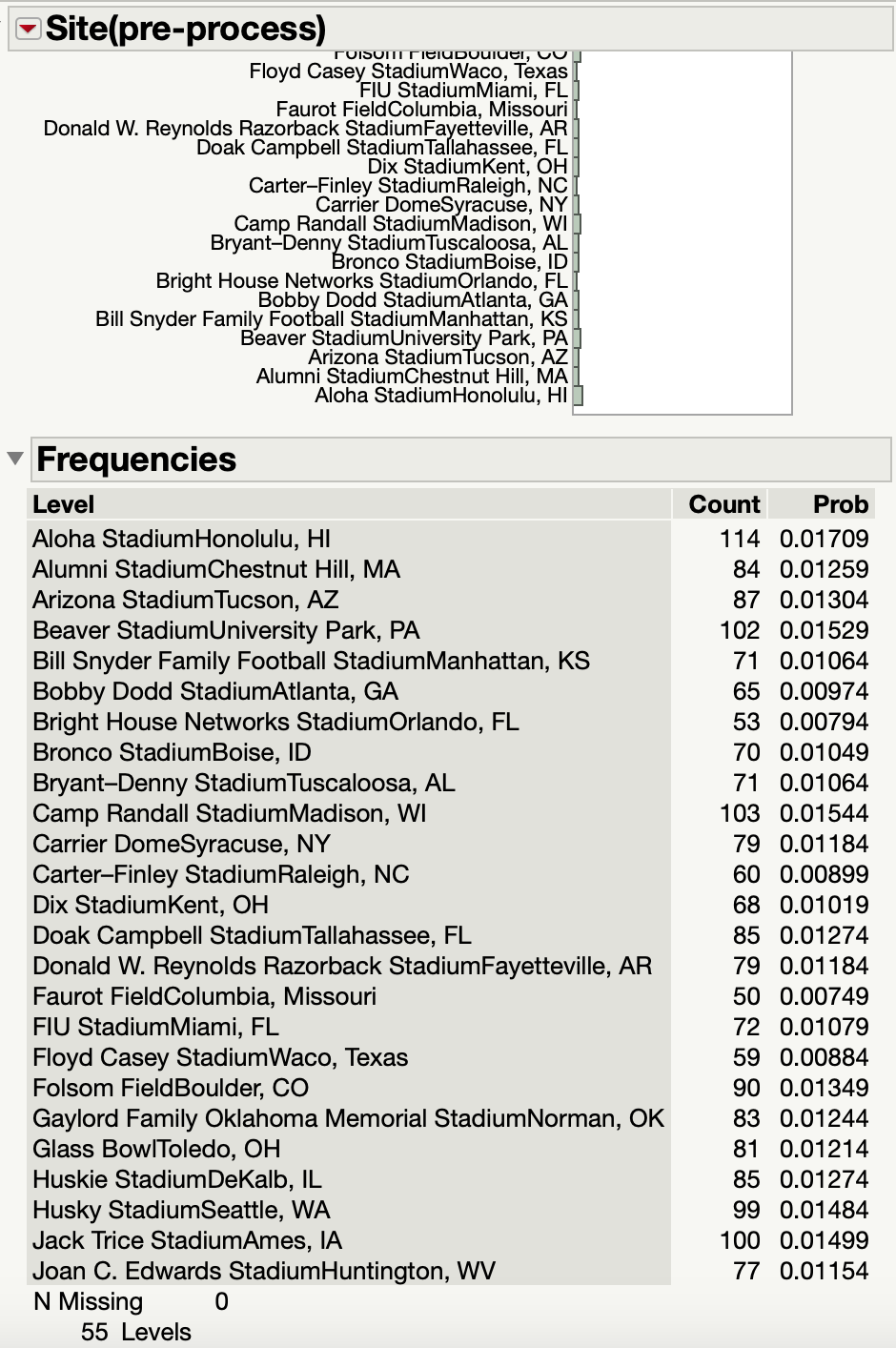


Figure 6 (site after pre-process)

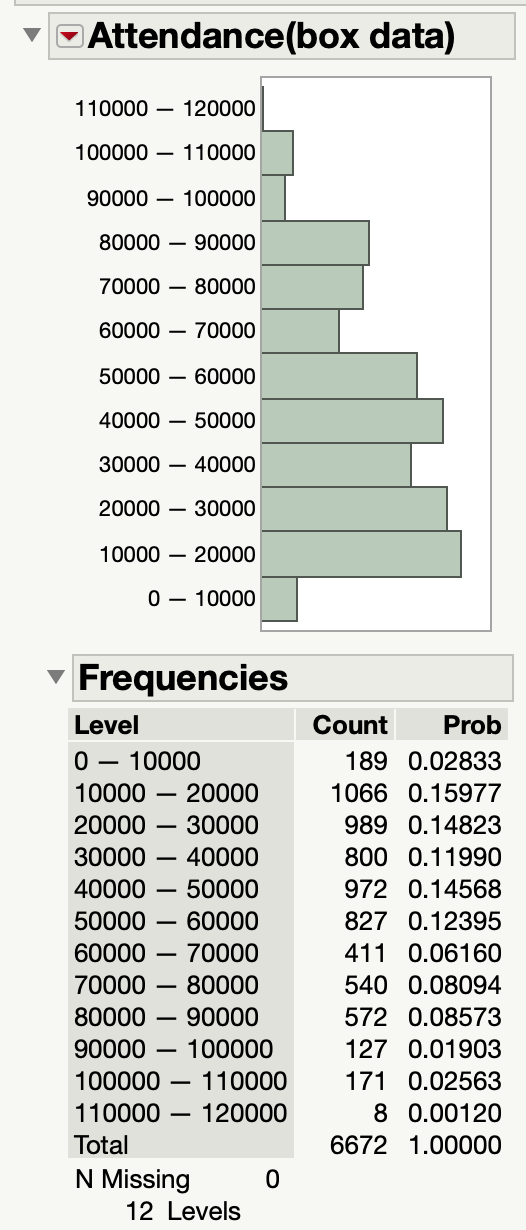


Figure 7 (nominal column of attendance)

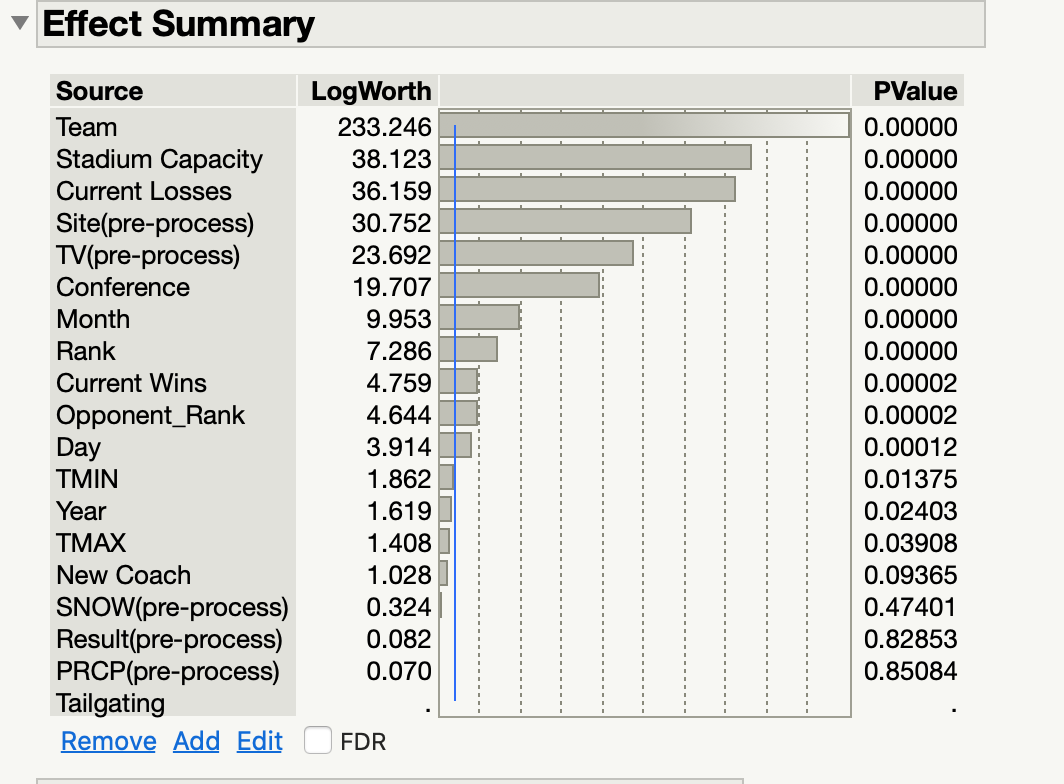


Figure 8 (linear regression with all variables)

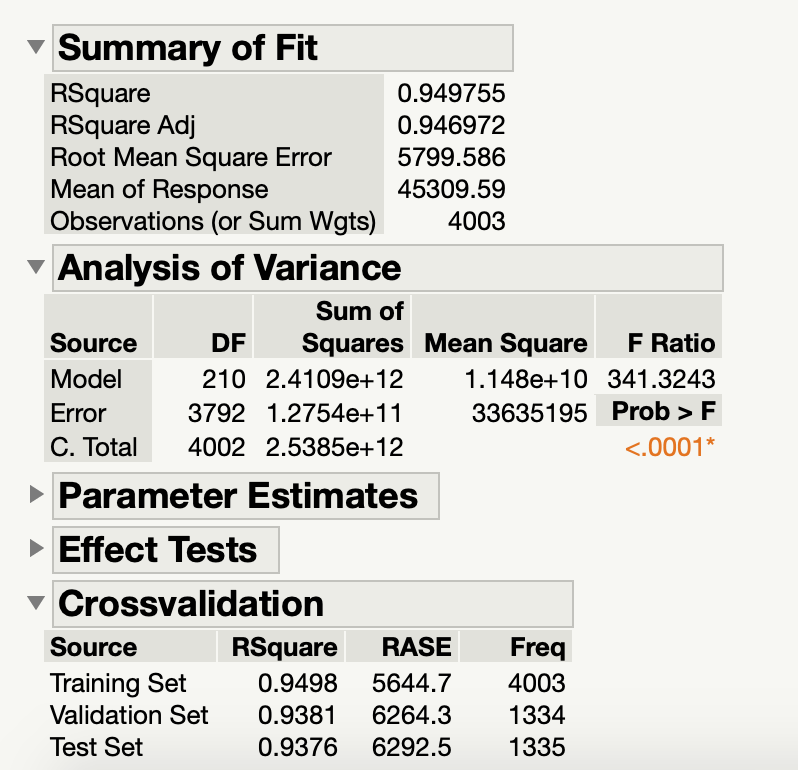


Figure 9 (summary statistics)



Figure 10 (best linear regression model)

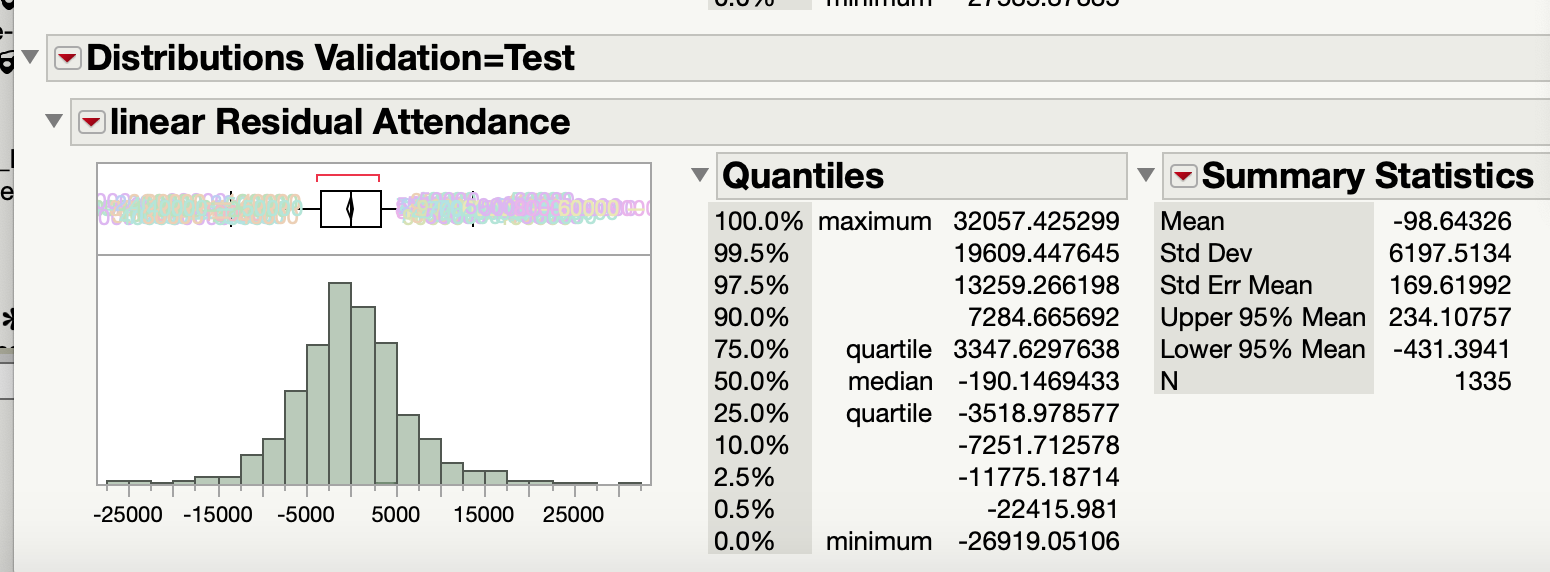


Figure 11 (residuals of linear regression)

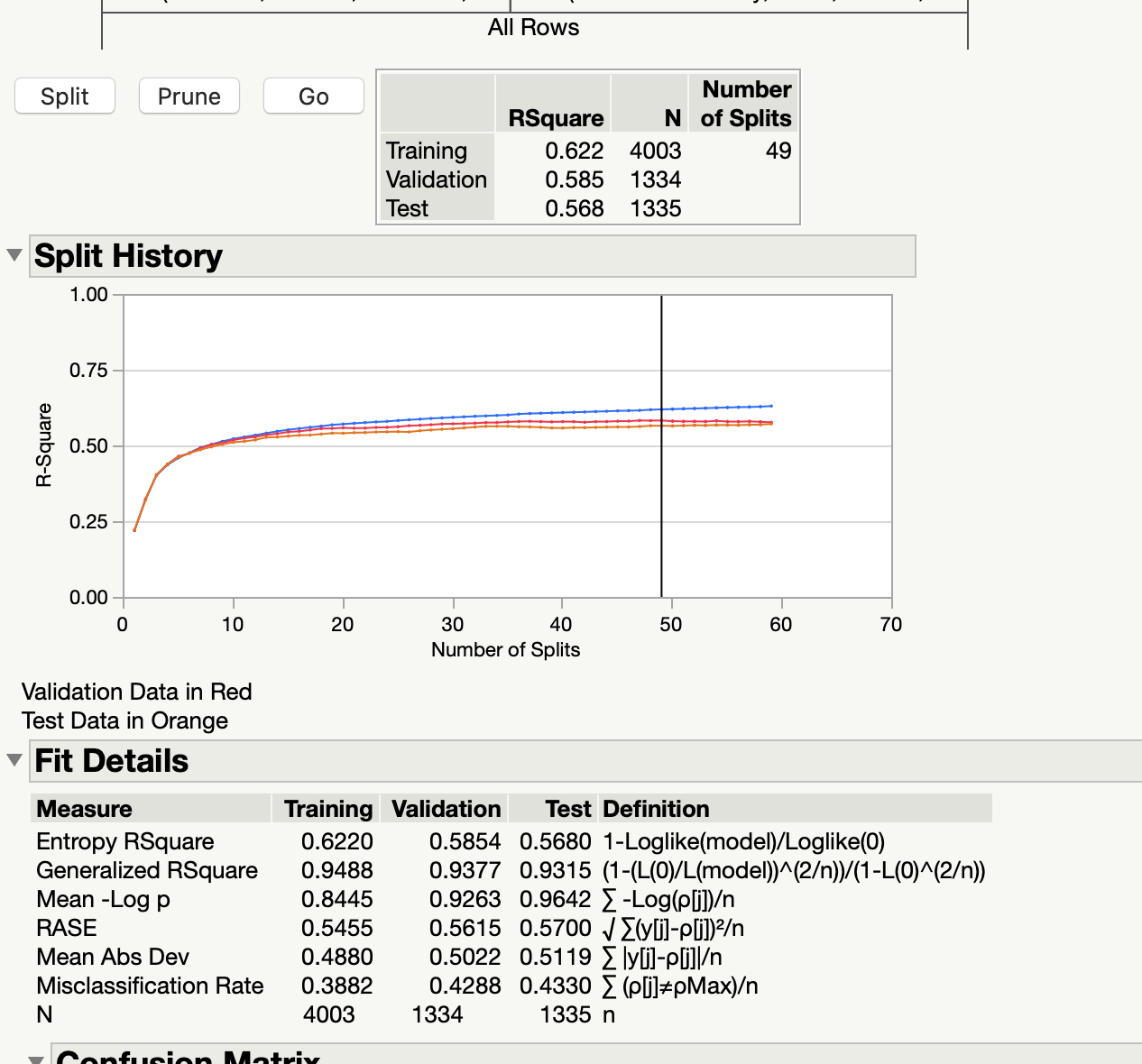


Figure 12 (decision tree 49 splits)

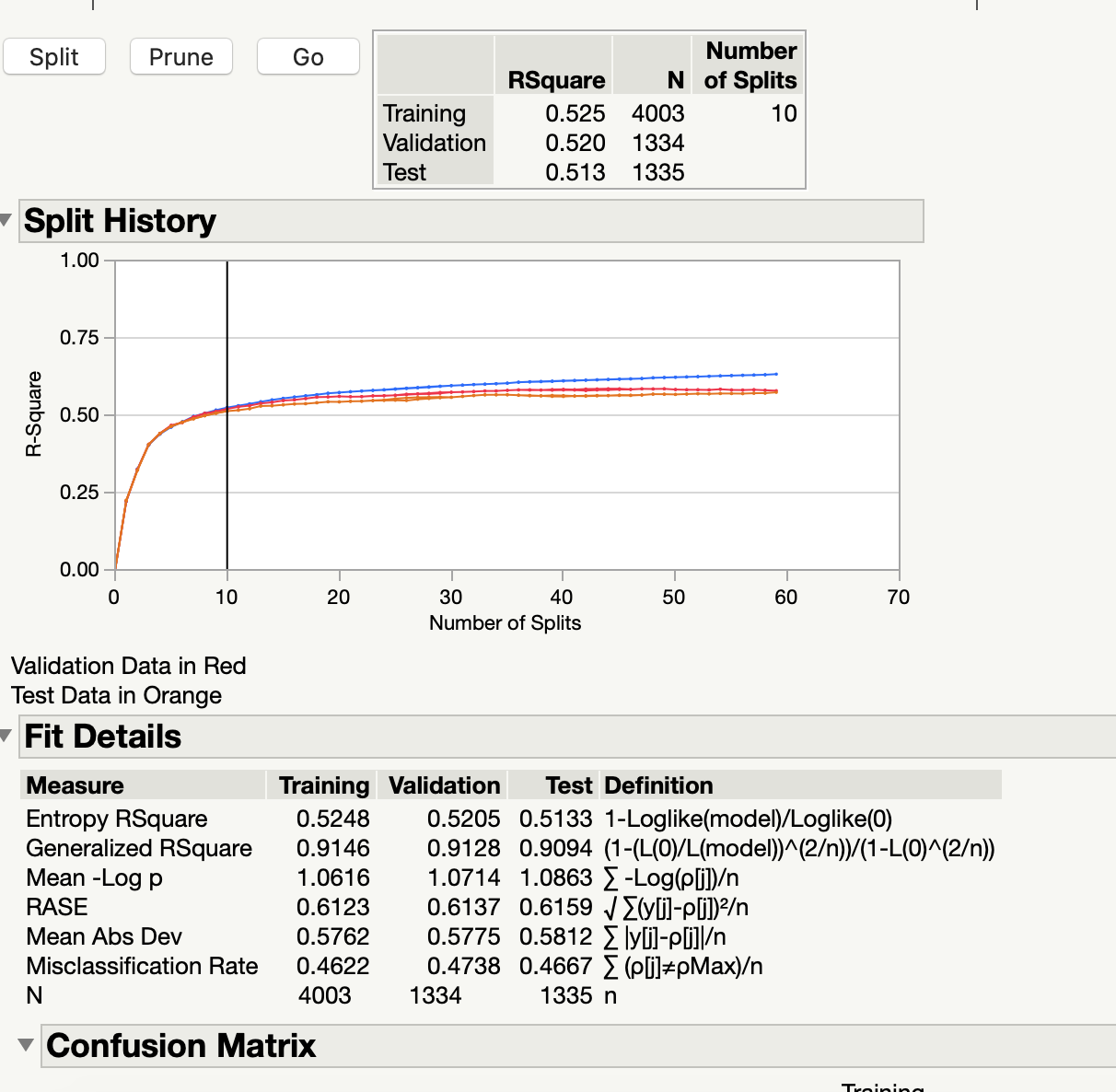


Figure 13 (decision tree 10 splits)

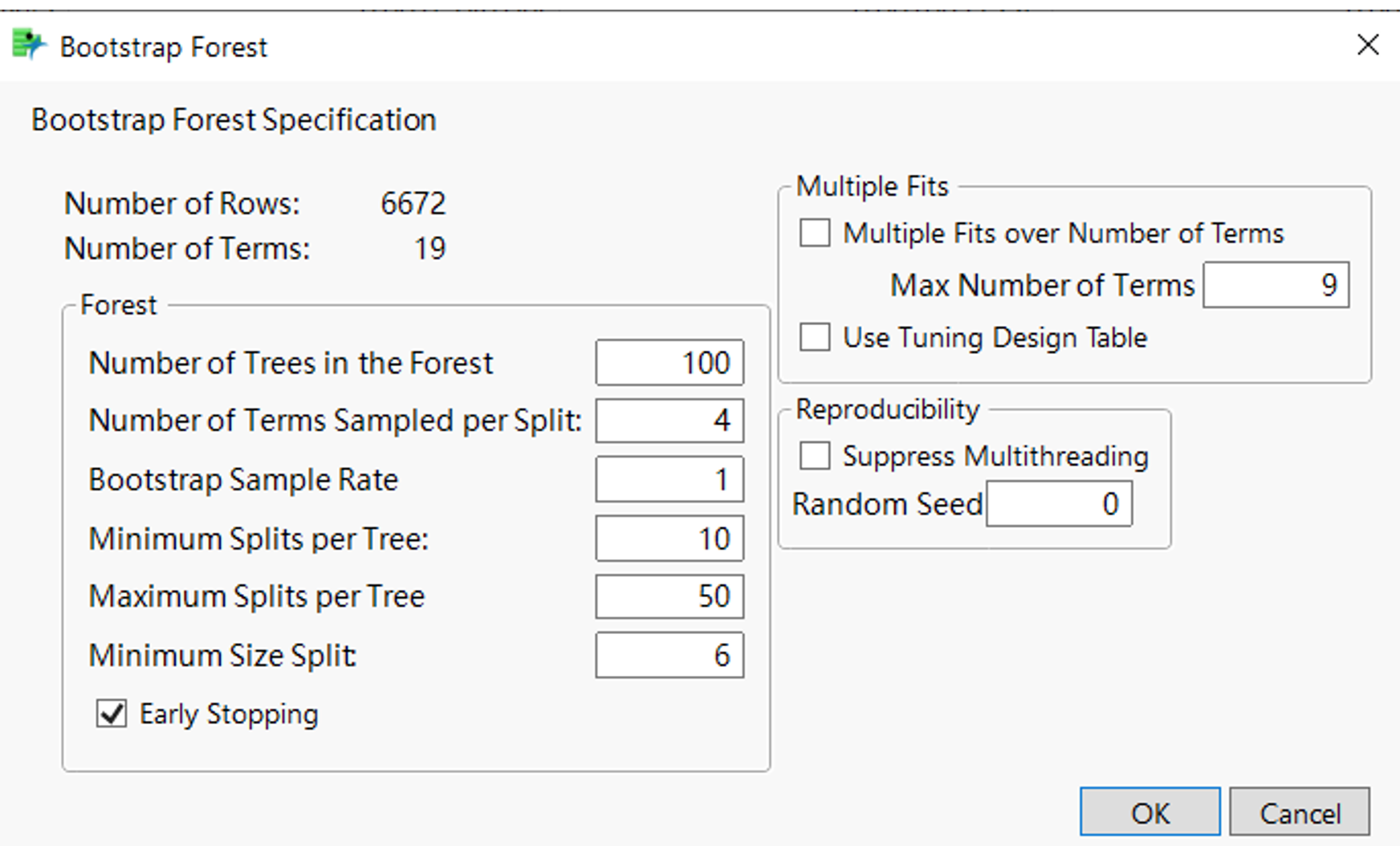


Figure 13 (settings for bootstrap forest)

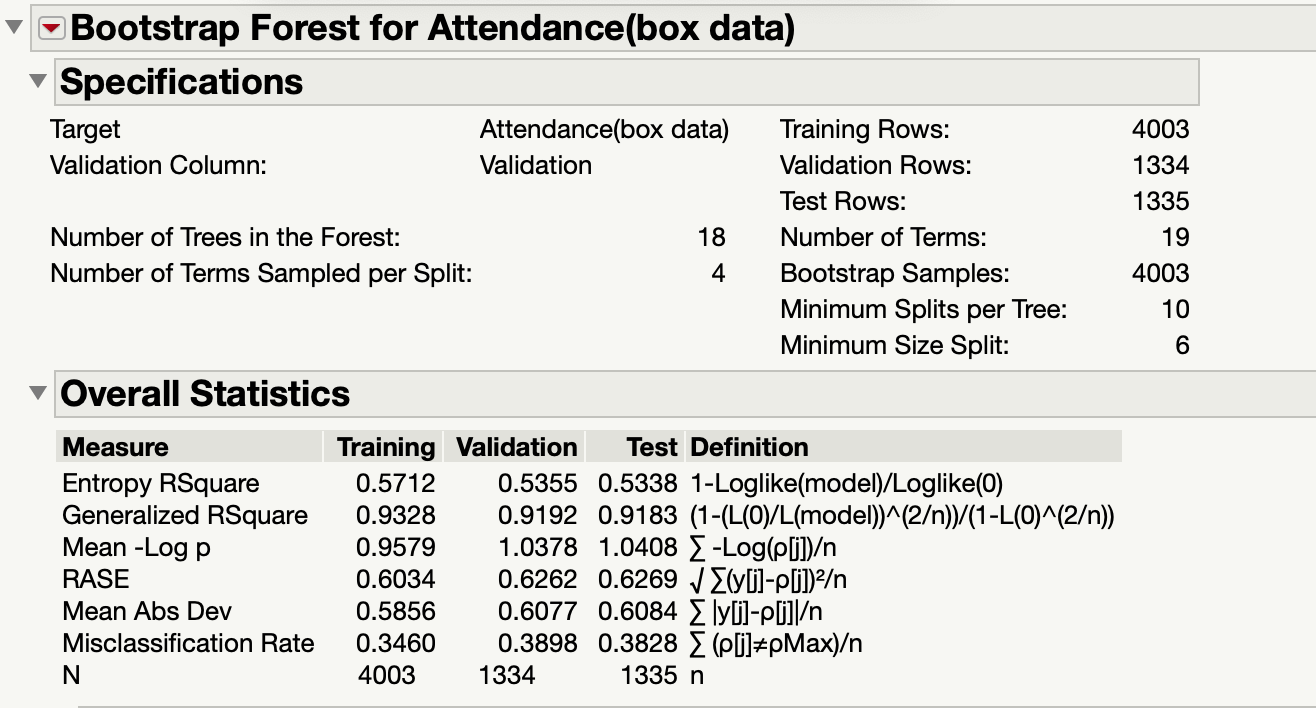


Figure 14 (summary statistics of bootstrap forest)

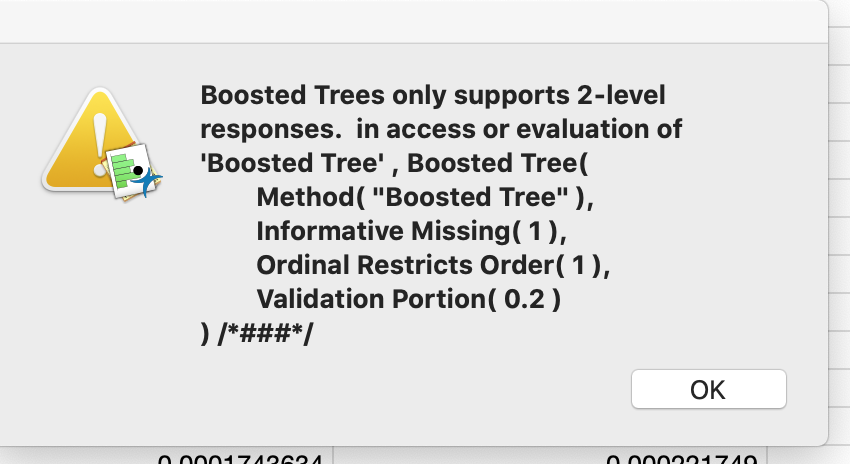


Figure 15 (boosted tree)

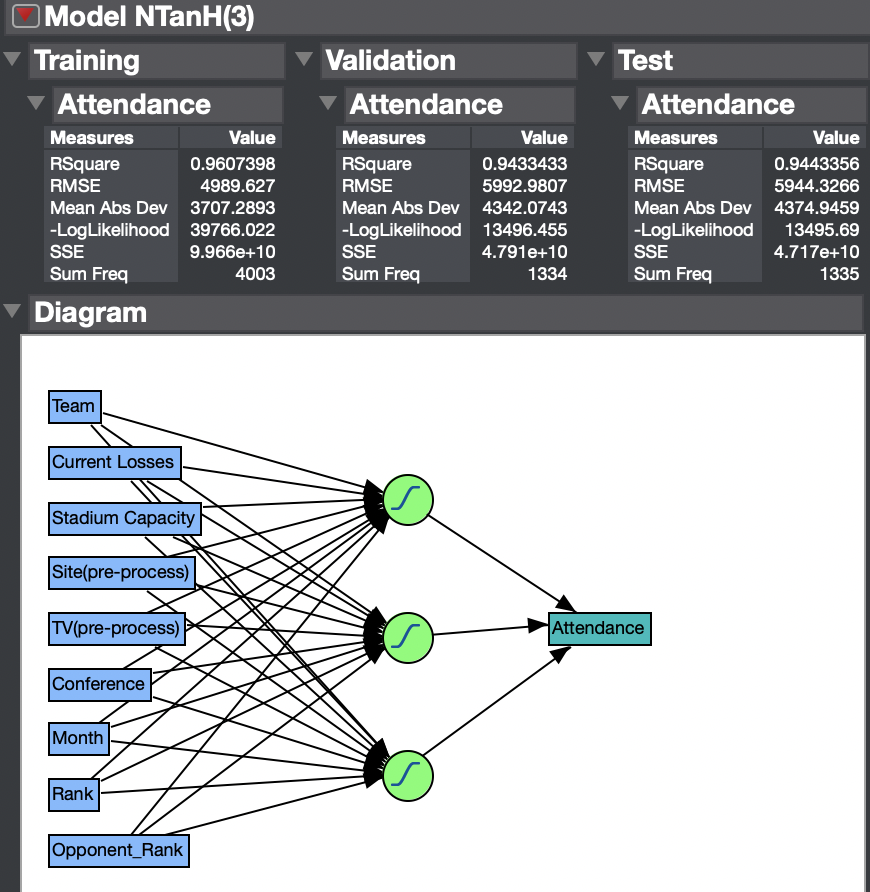


Figure 16 (neural model with 3 TanH)

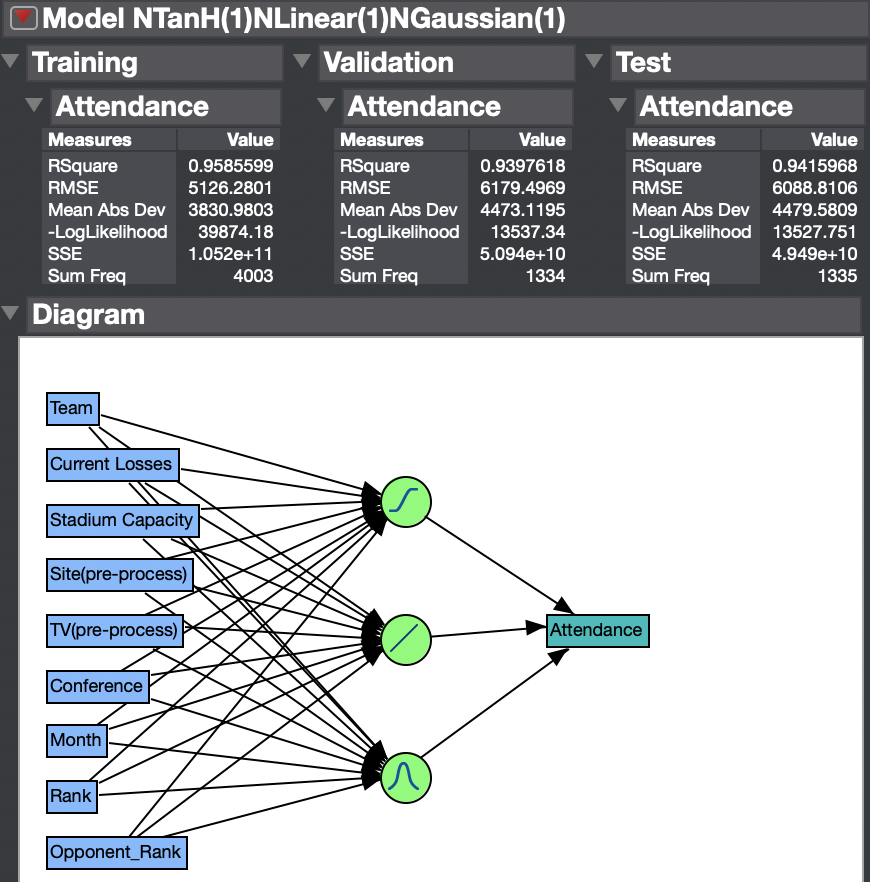


Figure 17 (neural with 1tanH, 1linear and 1Gaussuian)

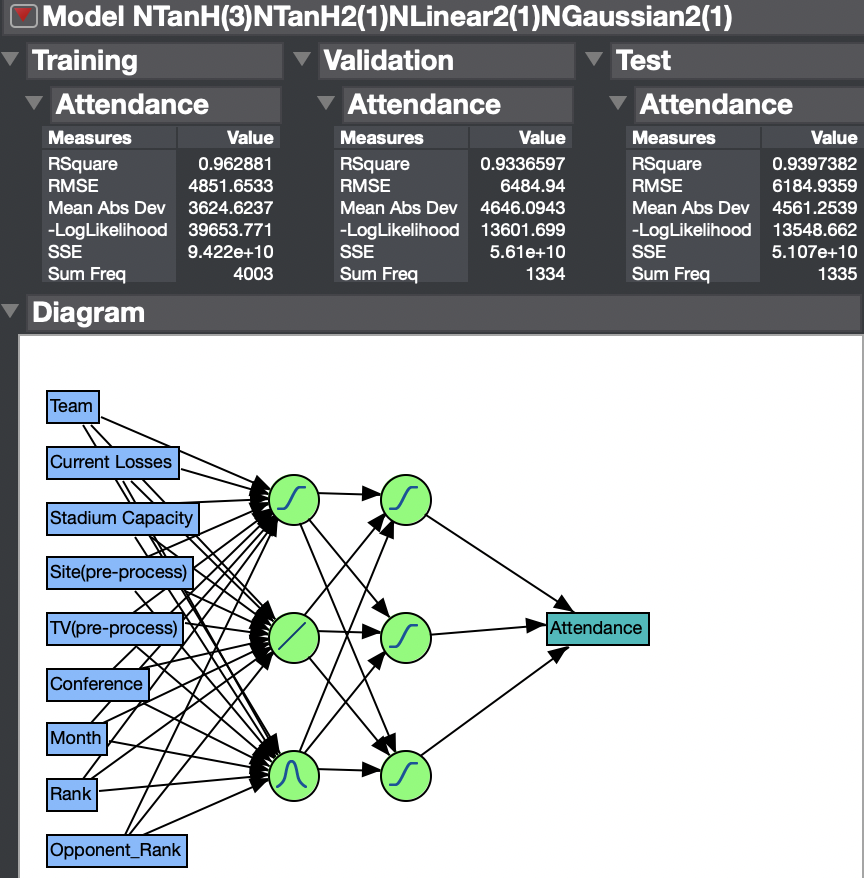


Figure 18 ( neural with 2 layers of nodes)

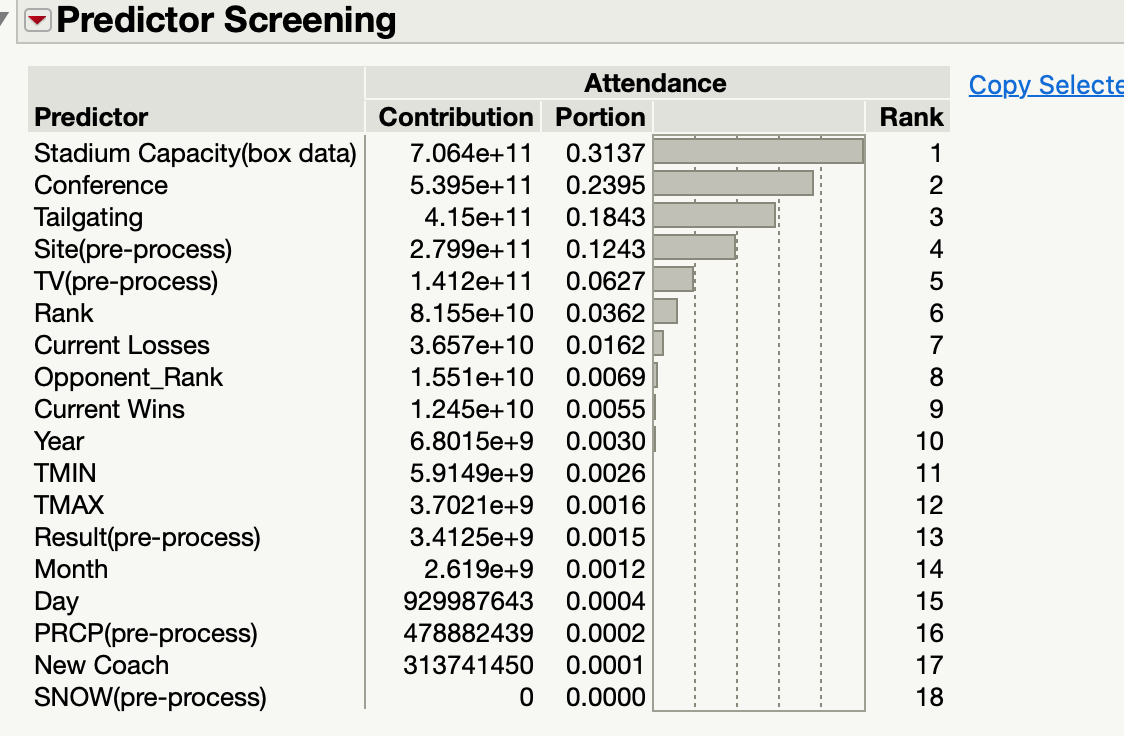


Figure 19 （predictor screening for KNN)

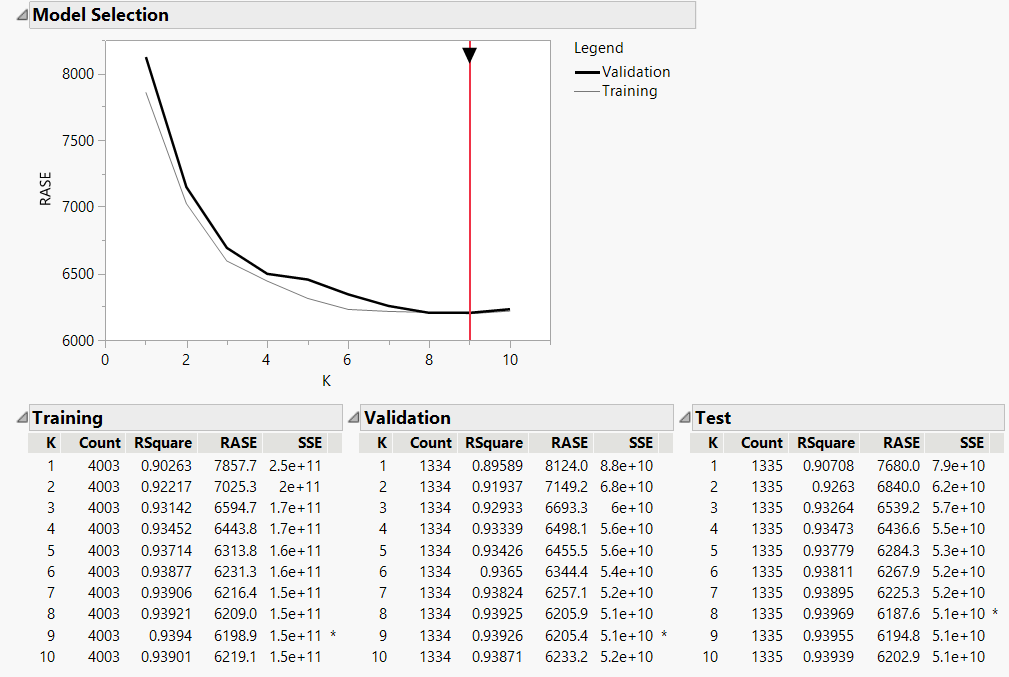


Figure 20 (KNN model)

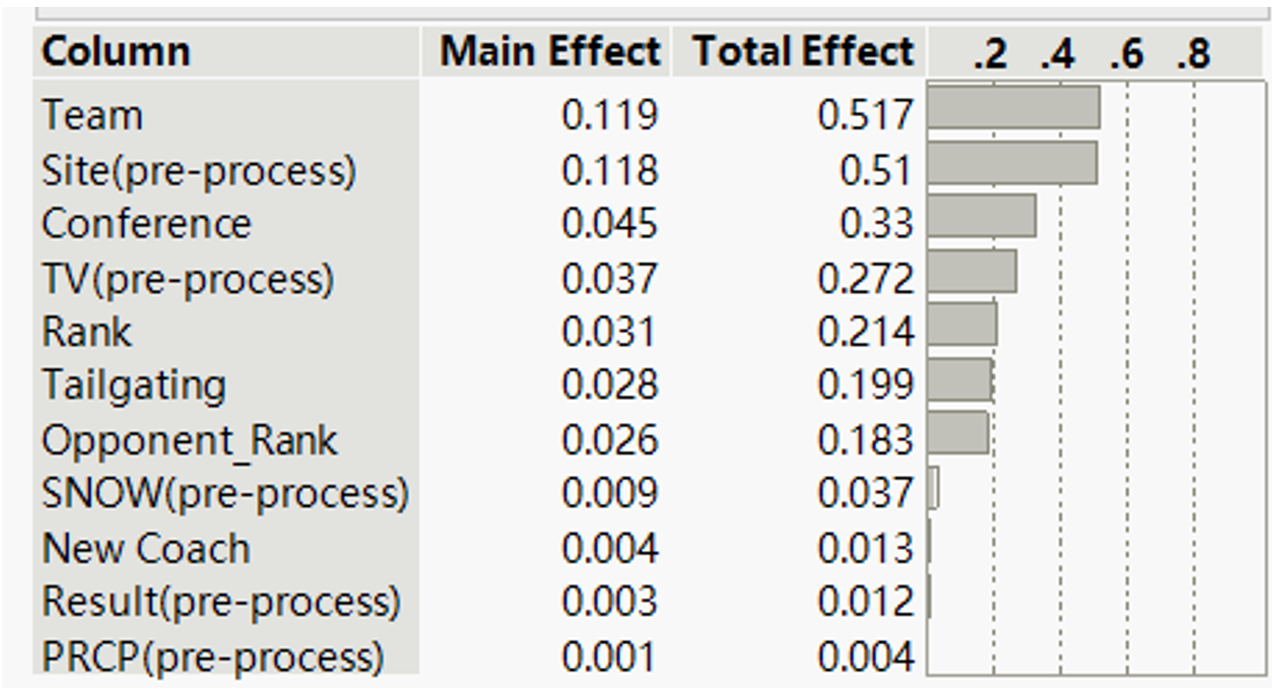


Figure 21 (naive baye first trail variables)

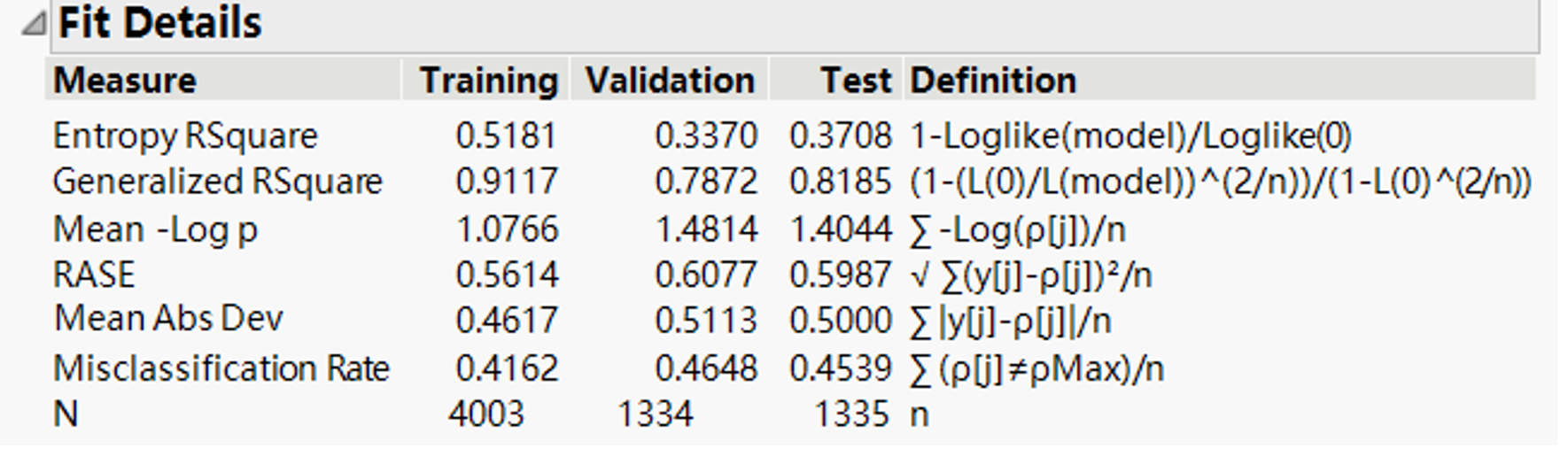
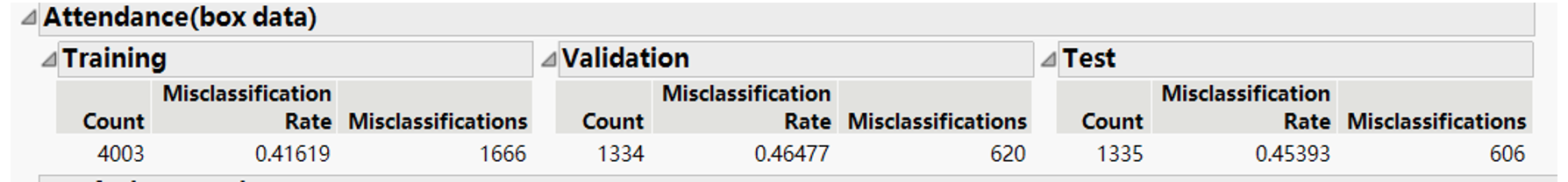


Figure 22 (naive baye fit detail)

Figure 23 (naive bayes partition comparisons)

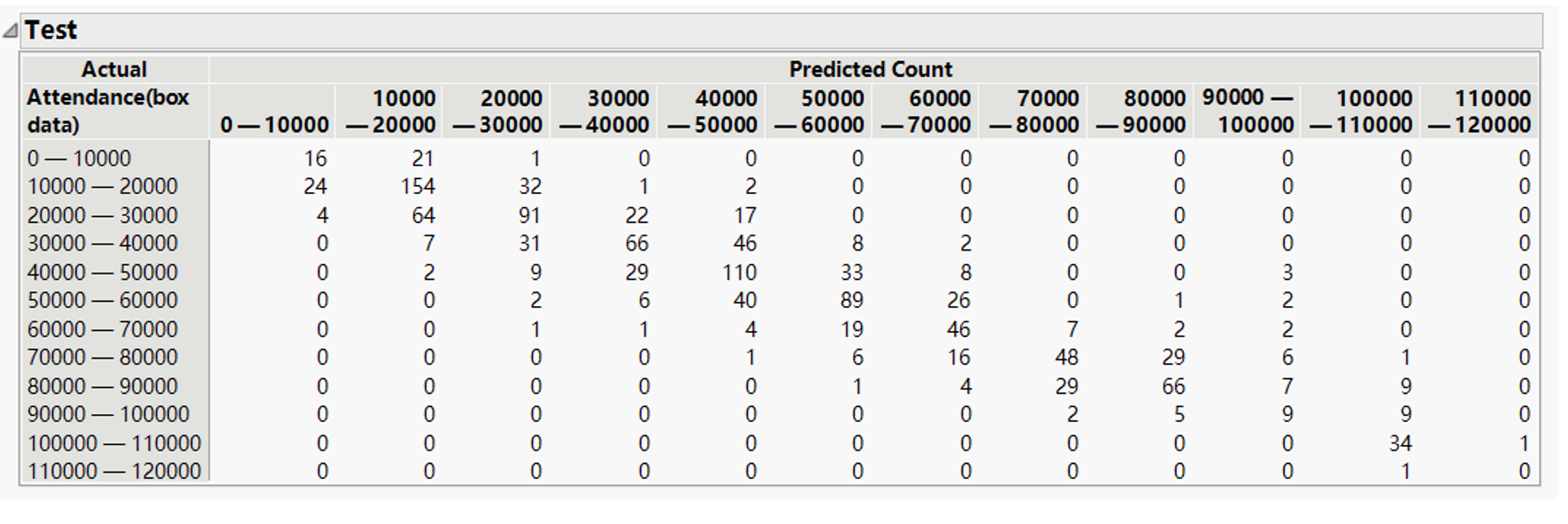


Figure 24 (naive baye best confusion matrix)

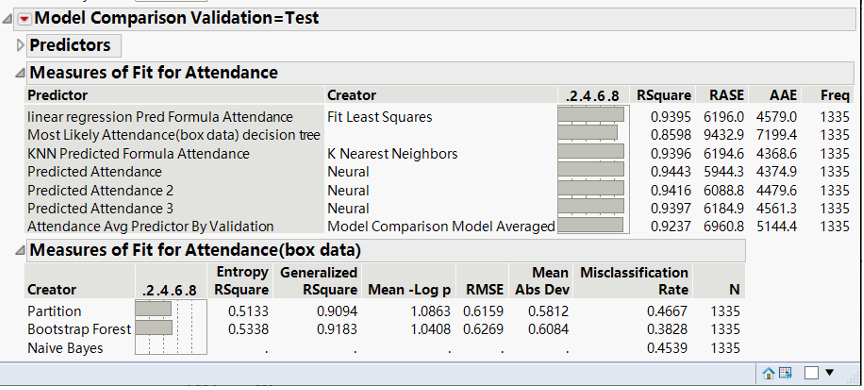


Figure 25 (ensemble model average of all continuous models)

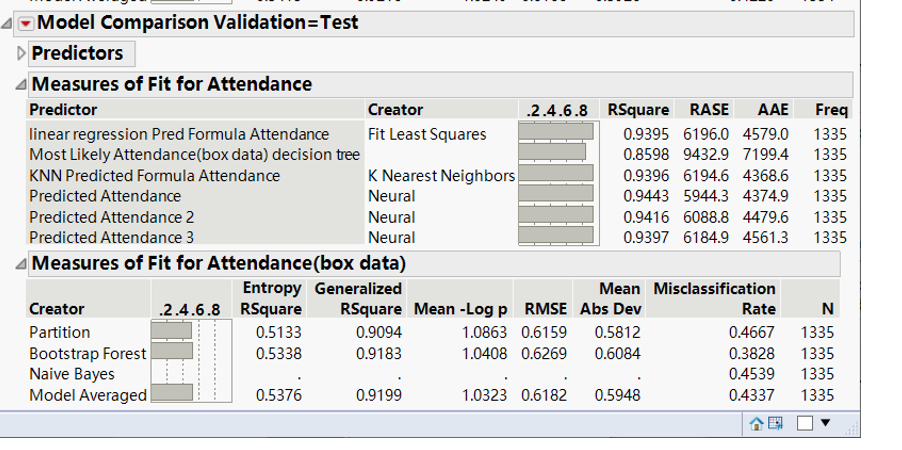
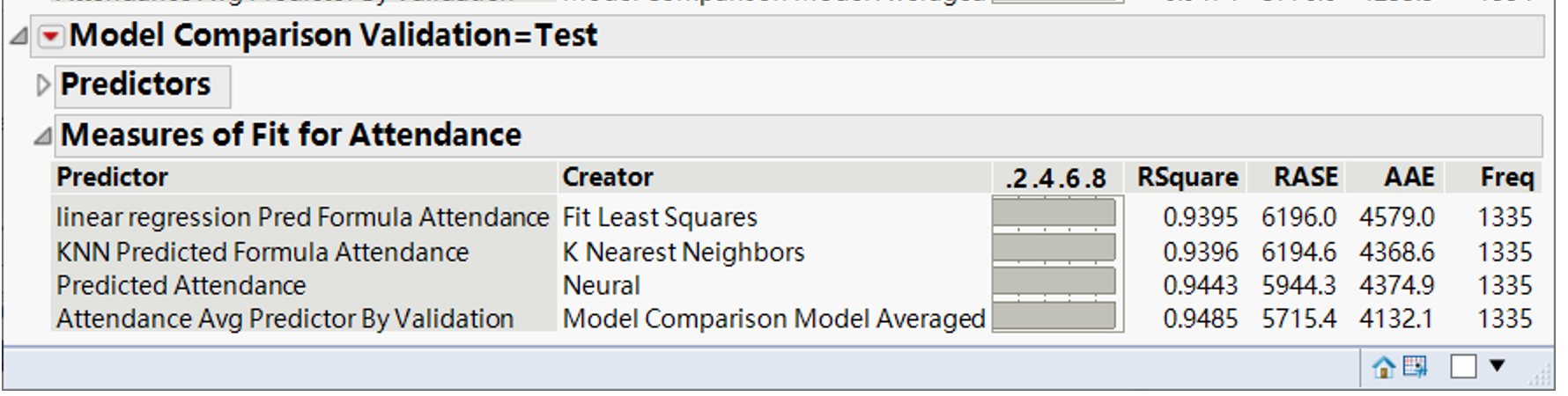


Figure 26 (average of all nominal models)

Figure 27 (average of our best three models, also the overall best)

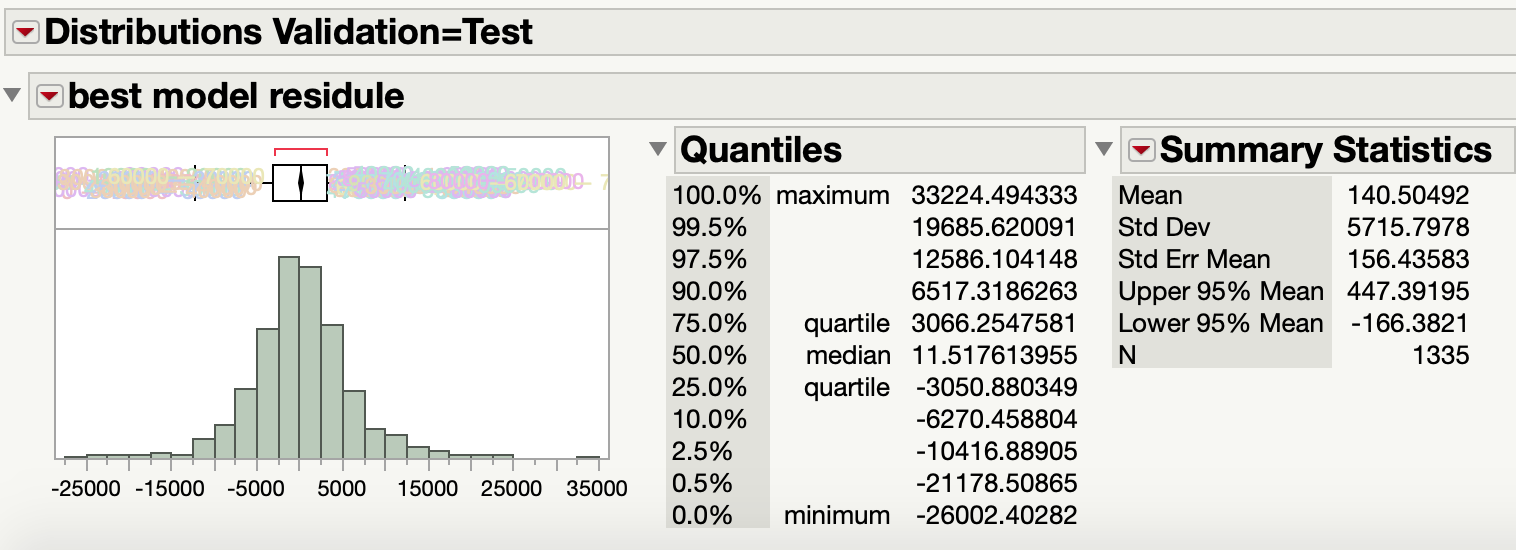


Figure 28 (residual of our best model)

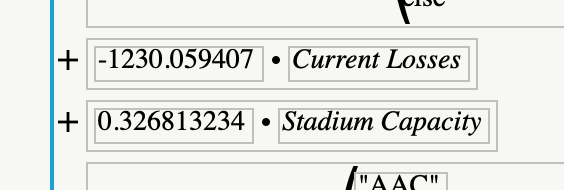


Figure 29 (formula of linear regression model)

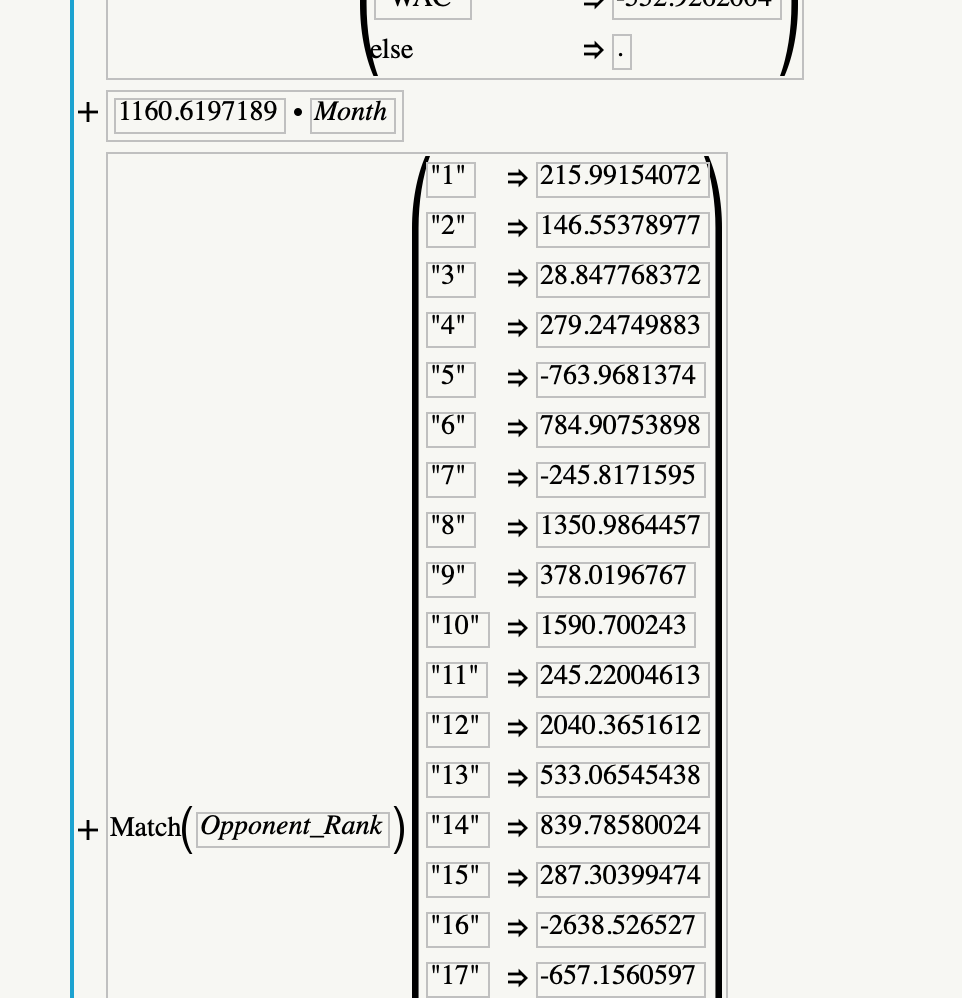


Figure 30 (formula of linear regression model)