FINAL PROJECT

OPIM 5604 – Fall, 2016

Team 8 –

presentation: interesting problem that generated excellent discussion; very good intro and clear problem statement; good discussion of why NN not likely useful in the problem context – overall strong presentation

report:
strengths: very thorough; used techniques beyond what we studied in class; tone of exec summary ok; very thorough analysis that showed strong effort and what appeared to be a solid team effort; writing generally ok with a few points of awkwardness

weaknesses: exec summary should include summary results from best models – important that execs see possibilities up front; I don't understand why you would round when doing continuous target variable estimation

overall – good project and presentation – 96/100

TEAM #8

CLASS SECTION: B12 (EVENING)

TEAM MEMBERS:

Formatted: Indent: First line: 0", Line spacing: single

ANIL JOSHI
ARKOJYOTI HORE
PRASHANT PS
PARTH SHIRAS
SURYA RAMARAJ
SRIRAMAN SRINIVASAN

The work presented here is our team's and our team's work alone.

Table of Contents

The work presented here is our team’s and our team’s work alone.

The work presented here is our team's and our team's work alone.

EXECUTIVE SUMMARY

Problem Description

This document represents a thorough analysis of the dataset "European Soccer Players". The dataset comprises ~~of~~ ratings corresponding to various attributes describing the playing ability of the footballers who are looking forward towards participating in the upcoming European Soccer League. Every season, multiple teams participate in this grand competition and the team management aim to form the best possible composition for respective teams depending on player ratings and potential. The players are selected from two different pools: goalkeepers(GKs) and non-goalkeepers/fieldplayers (NonGKs). Keeping this in perspective, using the SEMMA approach, we tried to achieve the following objectives:

- Based on current overall rating and projected rating, predict if a footballer (GK or NonGK) has high or low potential to grow in the upcoming season (Binary Classification problem)
- Predict the overall rating of a footballer (GK or NonGK) (Regression problem)

Approach

After checking for preliminary discrepancies in the data, we identified the target variables for regression and classification problems respectively. This was followed by univariate and multivariate analysis, missing value treatment, outlier analysis and observing the explanatory ability of the various attributes. Since the data did not provide information on player positions, clustering analysis revealed that there were two well-defined clusters which segregated goalkeepers from the non-goalkeepers in terms of the various playing attributes. Hence, we separated the GKs from the NonGKs and built regression and classification models for each. Finally, based on various model evaluation techniques and costs incurred, we recommended the most suitable models for predicting the required outcomes.

Insights

- The dataset provided was clean and did not have any missing values. However, providing player positions could have aided in more accurate analysis
- The target variable for regression problem could be readily identified as "overall_rating". However, for the classification problem we segmented the players on difference in their future potential ratings and current ratings to determine the target variable ("Potential_Classification"). This aligned with the business perspective since the management would like to sign a player who is more likely to grow
- The exploratory analysis revealed interesting trends to explain the relations between the attributes and their level of significance in generating the predictions
- For regression problem, the requirement was to determine how the attributes are significant in predicting the variance in the overall rating. After evaluating multiple models, we recommended that "Boosted tree" for NonGKs and "Stepwise regression" for GKs. The assessment was done based on factors like the interpreting ability of the model, complexity, error distributions and costs incurred on overestimating and underestimating the overall rating of a player. We were also able to demonstrate how the management could achieve significant cost-savings by opting to use the suggested model instead of the naïve approach, which is selecting players merely on experience and performances
- For classification problem, although the baseline models incurred lower costs, they are not recommended in real time scenario as they would involve the highest number of misclassification instances. After evaluating multiple models, we recommended using "Decision Trees" to predict the potential of a goal keepers and Neural Networks model for non-goal keepers. The assessment was done on factors such as minimization of cost incurred in misclassifications and accuracy of true classifications

The work presented here is our team's and our team's work alone.

- provide results for best model along with your recommendations – where from here ? don't include so many statements about all the things team did

Formatted: List Paragraph, Bulleted + Level: 1 + Aligned at: 0.25" + Indent at: 0.5"

The work presented here is our team's and our team's work alone.

METHODOLOGY FOLLOWED

Since the management required us to analyze the attributes for all the players who are available to participate in the competition for the upcoming season, hence instead of considering a sample of the data, we have used the entire data related to available players. awkward writing - Thus, apart from 'Sampling' ??? , we have conducted our analysis adhering to the SEMMA approach in the following manner:

- Procuring data
- Exploration
  - Identified the continuous and categorical variables
  - Recoded values for a categorical variable since there typographical errors
  - Identified the target variable for regression problem
  - Separated goal keepers from other players using clustering techniques and aligning our understanding with the business requirement.
  - While the goalkeeper dataset ~~comprised of~~had 897 observations~~rows~~, the non-goalkeeper dataset contained 9513 observations
  - For each of the two datasets, viz. goalkeepers (GKs) and non-goalkeepers (NonGKs), calculated the anticipated potential growth of the players using the *potential* and *overall_rating* columns. This helped us to obtain the target variable for classification problem
  - Performed univariate and multivariate analyses to discover hidden patterns in data
- Modification
  - ~~Based on the data exploration,~~ carried out missing value and outlier analysis
  - Tried to reduce data complexity and checked if majority of the information was being captured
- Modeling
  - For each of the two datasets, we modeled a nominal and a continuous target variable ~~aimed to analyze the classification and regression problems~~
  - ~~Hence,~~ multiple predictive models were built to search for an optimal combination of data that reliably predicted the desired outcome
- Assessment
  - We compared the performances of the various models with the baseline model to understand the value-added by the predictions
  - Depending upon various evaluation metrics we gauged the performance of the models and recommended the optimal models for classification and regression problems

Formatted: Highlight

Formatted: Highlight

Formatted: Highlight

The work presented here is our team's and our team's work alone.

PROCURING DATA

- The data was fetched from a sqllite database using the following R script:

Figure 1

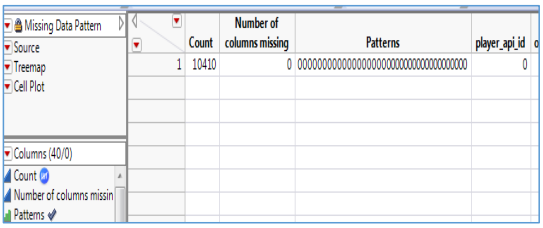
```
#importing .sqlite data
library(RSQLite)
con <- dbConnect(drv=RSQLite::SQLite(), dbname="C:/Users/Arko/Downloads/soccer/database.sqlite")
tables <- dbListTables(con)
tables <- tables[tables != "sqlite_sequence"]
lDataFrames <- vector("list", length=length(tables))

## create a data.frame for each table
for (i in seq(along=tables)) {
  lDataFrames[[i]] <- dbGetQuery(conn=con, statement=paste("SELECT * FROM '", tables[[i]], "'", sep=""))
}
#save the data describing the player attributes
write.csv(tables[[1]],"European Soccer Players.csv")
```

- Since the management required us to analyze data only for the most recently played season, we used the date column to filter observations corresponding to the player attributes only for the previous season
- The dataset comprises of 10,410 observations and 37 columns

EXPLORATORY DATA ANALYSIS

Missing Data Analysis

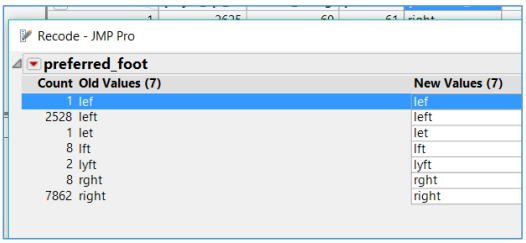
Figure 2



Approach: Select Tables -> Missing Data Pattern
Observation: The dataset does not have any missing values

Recoding the values of the attribute "preferred_foot"

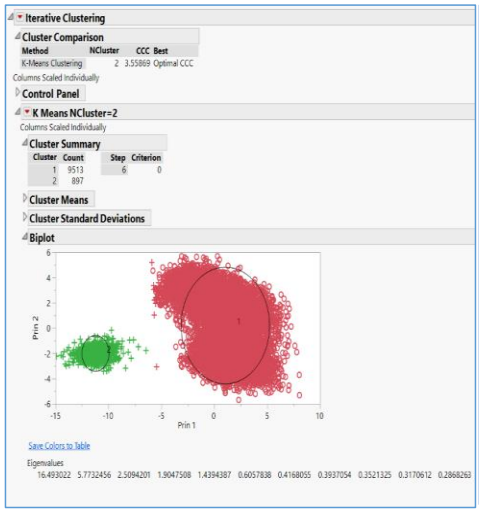
Figure 3



- Looking at the values, we realized that there were multiple typographical errors.
- Using the Recode feature, we corrected the values to "left" and "right"

The work presented here is our team's and our team's work alone.

K-means Clustering



Approach:

- Analyze -> Multivariate Methods -> Cluster
- We performed a cluster analysis using k-means clustering algorithm for different number of clusters (n=2 to n=5) on all continuous variables to discover any distinct clusters in the data.
- The best number of clusters leading to greatest separation was observed at n=2: one with 897 records and the other with 9513 records.

Observation:

- The centroid of the cluster 2 has significantly higher values of GK attributes whereas Cluster 1 has it's centroid comprising of higher values of Non GK attributes. This might indicate that there is clear separation between the two clusters based on GK and Non GK attributes. However, we need to also consider other statistical measures and business knowledge before taking a decision on the data separation

Formatted: Left

Figure 4

Intuition behind Separating Goalkeepers(GKs) and Non-Goalkeepers(NonGKs)

We decided to segregate the observations corresponding to GKs from that of the NonGKs based on the following reasons:

- In accordance with the problem description, the management is looking to select players from two different pools: GKs and NonGKs. Hence if we do not segregate the observations, the management will face difficulty in understanding whether the predicted overall ratings/potential is for a GK or a NonGK
- Upon discussion with team management, we realized that the costs associated with incorrect predictions is significantly different for GKs and NonGKs. Considering the observations together would lead to incorrect cost calculations for models
- Although the dataset does not explicitly provide the player positions, it can be observed from the above clustering analysis that the values of attributes are distinctly different for the two types of footballers.

Hence, we have decomposed the dataset into two parts for GKs and NonGKs respectively: "final_dataset_GK" and "final_dataset_Others"

NOTE: For ease of understanding, after performing modeling and analyses, we have saved the results in the following files:

- "final_dataset_Others_Classification" -> contains results for classification models for NonGKs
- "final_dataset_Others_Regression" -> contains results for regression models for NonGKs
- "final_dataset_GK_Classification" -> contains results for classification models for GKs
- "final_dataset_GK_Regression" -> contains results for regression models for GKs

The work presented here is our team's and our team's work alone.

ok – good explanation -

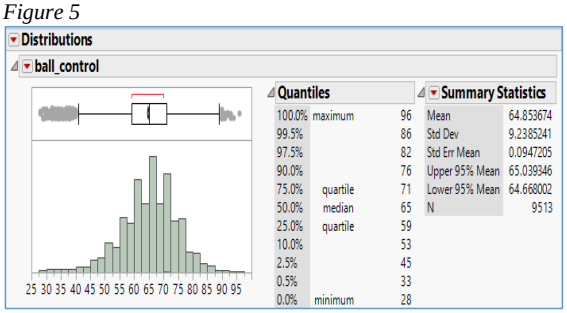
The work presented here is our team's and our team's work alone.

Univariate Analysis

As part of the data exploration phase, we conducted univariate analysis on all the variables. Following are the univariate analysis for some of the variables (Few more have been mentioned in Appendix):

- "ball_control"

Figure 5

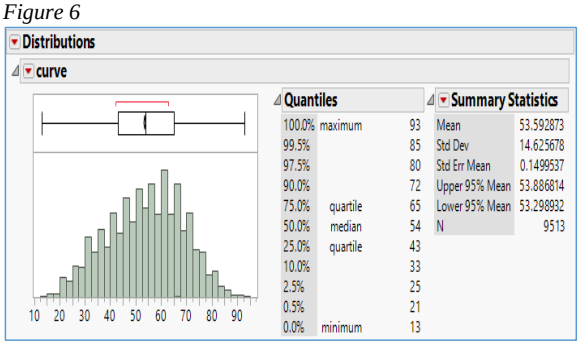


Observations:

1) The distribution is slightly left skewed
2) Although the attribute has 145 outliers which lie outside the 3IQR region, we are not ignoring them since these can represent valuable players with proficiency in ball controlling

- "curve"

Figure 6

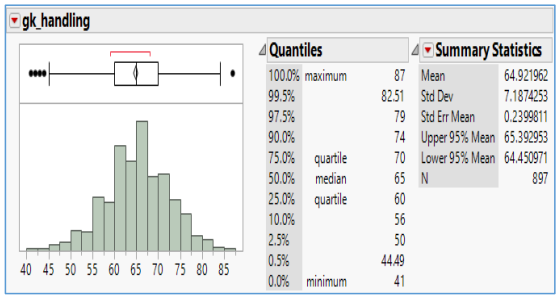


Observations:

1) The variable indicates the ability of a footballer to curve the shot in the air
2) There are no outliers in this column
3) Also, the data is close to being normally distributed
4) The variable does not require any transformation and original values can be used for further analysis

- "gk_handling"

Figure 6



Observations:

1) The variable indicates the proficiency of a goalkeeper in handling the ball
2) There are only 5 outliers in this column which lie outside the 3 IQR region
3) During multivariate analysis, we can observe how the overall rating and potential of a goalkeeper depends on this attribute

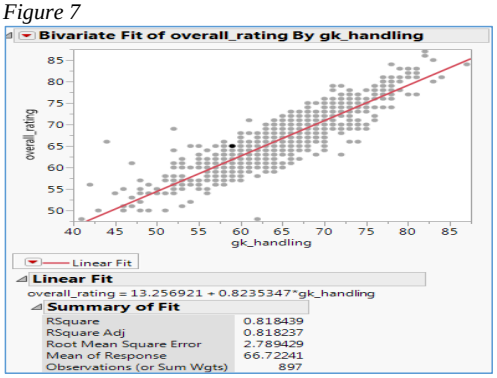
The work presented here is our team's and our team's work alone.

Multivariate Analysis

After observing the trends of the attributes individually, we then conducted multivariate analysis. The objective was to explore how predictors are related with each other as well as with the target variables. Following are the descriptions for some of the conducted analyses (Few more have been mentioned in Appendix):

- overall_rating vs gk_handling

Figure 7

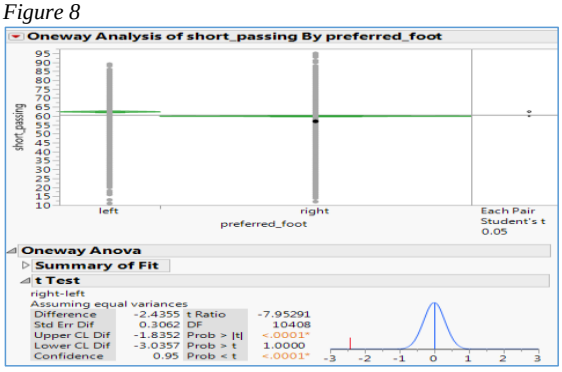


Observations:

1) As decided during univariate analysis, we analyzed the ability of gk_handling in interpreting the variance of overall_rating. The high value of R-square demonstrates strong interpretation ability
2) The scatter plot indicates a positive linear relationship between the two

- Claim: Players whose preferred foot is left have the same short passing rating as those whose preferred foot is right

Figure 8



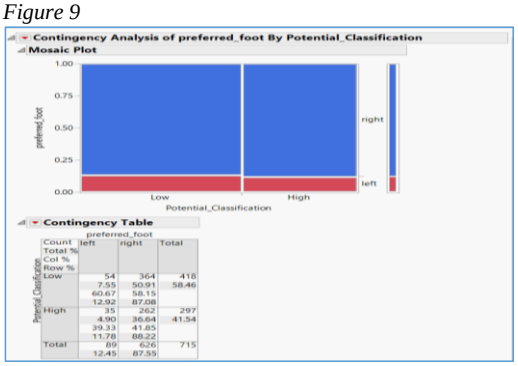
Observations:

1) Significance level (α) considered: 0.02
2) Since the Prob > |t| (< 0.0001) is less than the considered significance level, hence we can reject the claim

The work presented here is our team's and our team's work alone.

- Potential_Classification vs preferred_foot

Figure 9

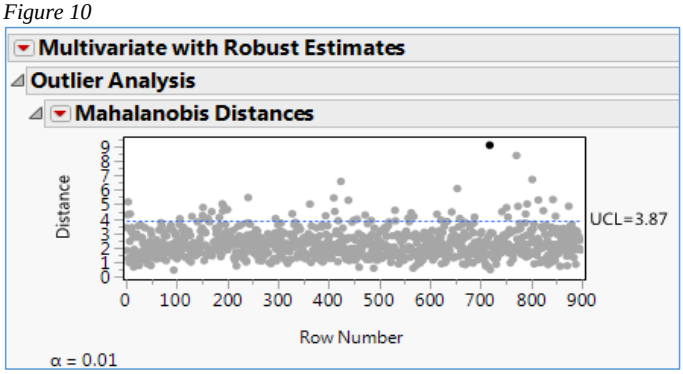


Observations:

1) The mosaic plot between the two variables indicates that in the data related to non-goalkeepers, majority of the footballers who have a high potential to grow are right-footed

- Robust Outlier Analysis for Goalkeepers
  - Approach: In the dataset for goalkeepers, considering all the continuous predictors, select Cols -> Modeling Utilities -> Explore Outliers -> Multivariate Robust Outliers
  - Selected significance level: 0.01

Figure 10



  - We saved the Mahalanobis distance for each observation and considered a threshold value of 8.50 such that any observation with a distance above this value will be considered as a potential outlier
  - It was observed that there was only one goalkeeper with player_api_id = 206641 which appeared to be an outlier. However, this player had a high potential to grow as per the target column "Potential_Classification". The management might be interested in considering this goalkeeper and hence we decided not to exclude this goalkeeper from analysis

The work presented here is our team's and our team's work alone.

CLASSIFICATION MODELING

Model Selection Criteria

The model selection will be done on two parameters-

- Maximizing the accuracy in correctly predicting the potential class of a player
- Minimizing the cost associated in misclassifying the potential of players (*For data related to salary of goalkeepers and non-goalkeepers please refer to references section*)

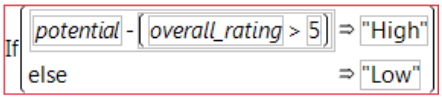
As per management, the costs will vary for misclassifying a low potential player as a high potential player and vice versa since the salaries offered to a footballer will depend on their projected potential. After doing further research on player salaries, we came up with following costs for different types of players:

- Cost per unit of misclassifying a low potential GK as a high potential GK (False Positives): $165,000
- Cost per unit of misclassifying a high potential GK as a low potential GK (False Negatives): $50,000
- Cost per unit of misclassifying a low potential NonGK as a high potential NonGK player (False Positives): $300,000
- Cost per unit of misclassifying a high potential NonGK as a low potential NonGK player (False Negatives): $48,000

From the above figures, it is evident that the cost associated with False Positives is much higher than the cost associated with False Negatives. The management will hire a low potential player for a higher salary in case of a False Positive. Hence it is more important for us to minimize the cost associated with False Positives which in turn will help us in bringing the overall cost down.

Deriving the Dependent Variable

- For classification problem, the objective is to classify whether a footballer has "High" or "Low" potential to grow based on his current rating ("overall_rating") and his projected rating("potential")
- As per management, the club manager will be interested in those players for whom potential can be improved by five points (that is, potential – overall_rating > 5)
- Hence, we calculated the difference between potential and overall rating. The dependent variable, "Potential_Classification", was derived using following formula:



Baseline Models

We have considered the following two scenarios while building the baseline models:

- Model which predicts all the players as low potential players
- Model which predicts all the players as high potential players

These models will serve as a basis for understanding how the management can reduce the costs of misclassification by utilizing the predicted outcomes from the suggested model. The following tables shows the cost calculations for the two type of baseline models separately for GKs and Non-GK players

Goal-keepers-

	Total Actual Low	Total Predicted Low	False Negatives cost ($)	Total Actual High	Total Predicted High	False Positives cost ($)	Total Cost ($)
Baseline Low	240	359	5,950,000	119	0	0	5,950,000
Baseline High	240	90	0	119	359	39,600,000	39,600,000

Non-Goal-keepers-

	Total Actual Low	Total Predicted Low	False Negatives cost ($)	Total Actual High	Total Predicted High	False Positives cost ($)	Total Cost ($)
Baseline Low	2232	3805	75,504,000	0	0	0	75,504,000
Baseline High	0	0	0	1573	3805	669,600,000	669,600,000

Modeling for Goal Keepers

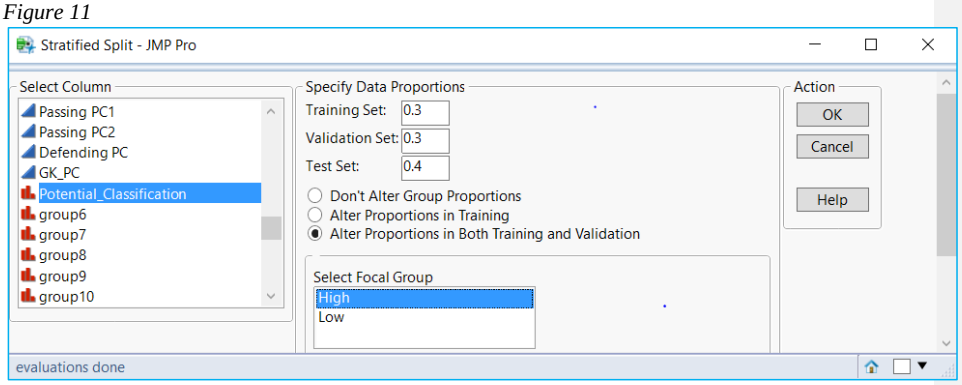
Data Stratification

In this process, we have taken balanced training and validation samples but unbalanced test sample. Comparing the model's performance on the three samples, if we observe that the confusion matrices for training and validation are significantly different, it can be inferred that the model suffers from overfitting

Approach: Add-ins -> Stratified Split Balanced

- Select 0.3 for Training Set, 0.3 for Validation Set, 0.4 for Test set
- Select the target variable (Potential_Classification) and 'Alter Proportions in Both Training and Validation' with Focal Group 'High'
- Take a subset of rows where Cluster gives us only GK variables

Figure 11



- After the stratification process, the dataset has 715 rows which can be used for modeling. We need to find the best model which will predict whether a player has high or low potential.
- Following are the variables that will be used in building the model for 'GK' players

The work presented here is our team's and our team's work alone.

- Target variable
  - Potential_classification
- Independent explanatory variables
  - Gk_diving
  - Gk_handling
  - Gk_kicking
  - Gk_positioning
  - Gk_reflexes

Algorithms Used

A. Decision Trees

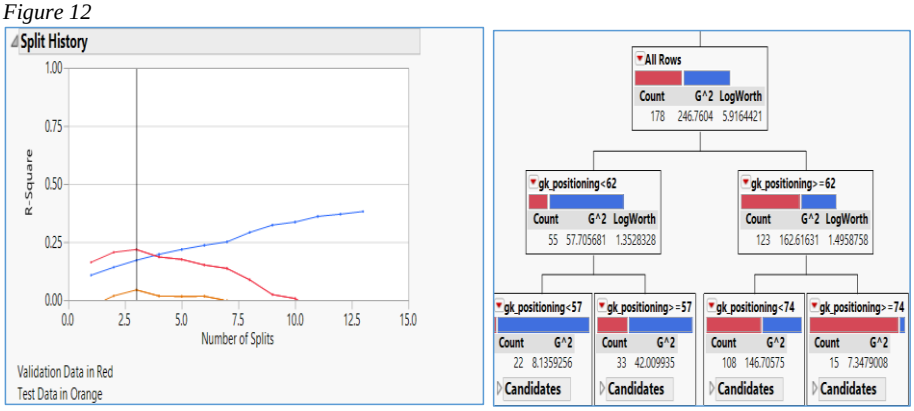
Approach: Analyze -> Modeling -> Partition

- Keep splitting till all matching records have the same output value and good classification accuracy without overfitting the model

Results:

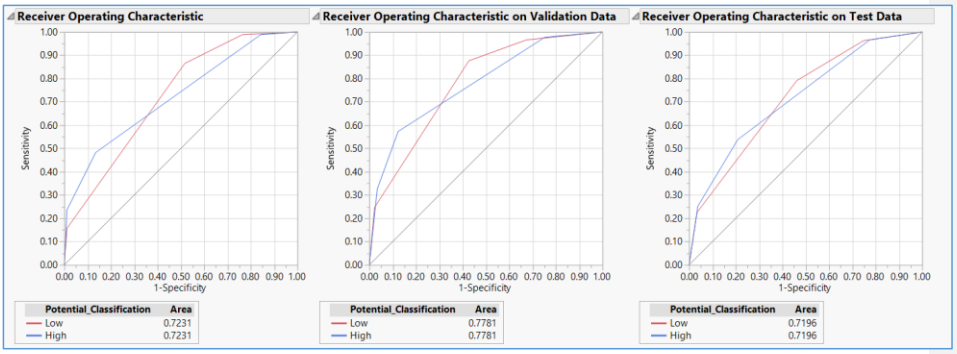
- As indicated in the Split History (Figure 12), we tried out more than 3 splits. However, more number of splits only increased the complexity of the decision trees without any significant improvement
- Hence, we stopped at the optimum level of 3 splits wherein a single variable contributed to the entire model
- The overall accuracy of the model on test data is 70.75%
- From the tree structure, the highest information gain and the final expansion is on the variable gk_positioning

Figure 12



- The ROC index is 0.7196 (indicates a strong model) for the test data which can be observed from the following graph (explanation for ROC curves provided in Appendix) :

Figure 13



- The following figure shows the fit details for the Training, Test and Validation datasets (considered a cut-off of 0.5 to convert the predicted probabilities into binary outcomes; will vary the threshold in the following section and observe the differences):

Figure 14

Fit Details

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.1714	0.2177	0.0438	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.2819	0.3474	0.0752	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.5744	0.5422	0.6074	∑ -Log(ρ[j])/n
RMSE	0.4465	0.4278	0.4560	√ ∑(y[j]-ρ[j])²/n
Mean Abs Dev	0.4029	0.3685	0.4014	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.3258	0.2753	0.2925	∑ (ρ[j]≠ρMax)/n
N	178	178	359	n

Confusion Matrix

Training

Actual Potential_Classification	Predicted Count Low	High
Low	77	12
High	46	43

Validation

Actual Potential_Classification	Predicted Count Low	High
Low	78	11
High	38	51

Test

Actual Potential_Classification	Predicted Count Low	High
Low	190	50
High	55	64

	True Positives	True Negatives	False Positives	False Negatives
Training	48.31%	86.52%	13.48%	51.69%
Validation	57.30%	87.64%	12.36%	42.70%
Test	53.78%	79.17%	20.83%	46.22%

The work presented here is our team's and our team's work alone.

Selecting a probability cut-off-

Probability	False Positives	Cost of False Positives ($)	False Negatives	Cost of False Negatives ($)	Total Cost ($)
0.35	186	30,690,000	4	200,000	30,890,000
0.4	186	30,690,000	4	200,000	30,890,000
0.45	50	8,250,000	55	2,750,000	11,000,000
0.5	50	8,250,000	55	2,750,000	11,000,000
0.55	50	8,250,000	55	2,750,000	11,000,000
0.6	50	8,250,000	55	2,750,000	11,000,000
0.65	50	8,250,000	55	2,750,000	11,000,000

- To convert the predicted probabilities to binary outcomes ("High" and "Low"), we have selected 0.5 as the optimal threshold value since it minimizes the total cost incurred

B. Neural Networks Algorithm

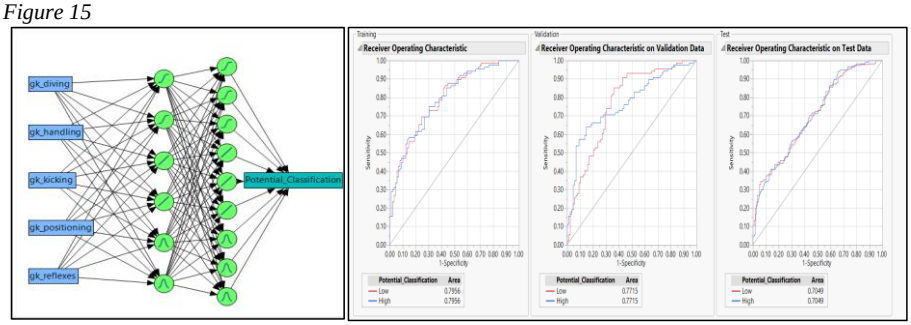
Approach: Analyze -> Modeling -> Neural

- To find out the best fit algorithm which gives better accuracy, we played with the number of functions at the two layers (Learning Layer and Classification Layer)

Results:

- After trying different combinations, we found that the combination of all three functions NGaussian, NTanH and NGaussian function drove the best accuracy rate for validation/test data
- Hence, we considered (NTanH(3) NLinear(3) NGaussian(3) NTanH2(2) NLinear2(2) NGaussian2 (2)) as weights, which gave best values for model evaluation metrics on test data
- Like Decision Trees model, we have considered multiple cut-off values for converting the predicted probabilities to binary outcome and selected 0.6 as the optimal threshold value
- The overall accuracy of the model on test data is 67.13%

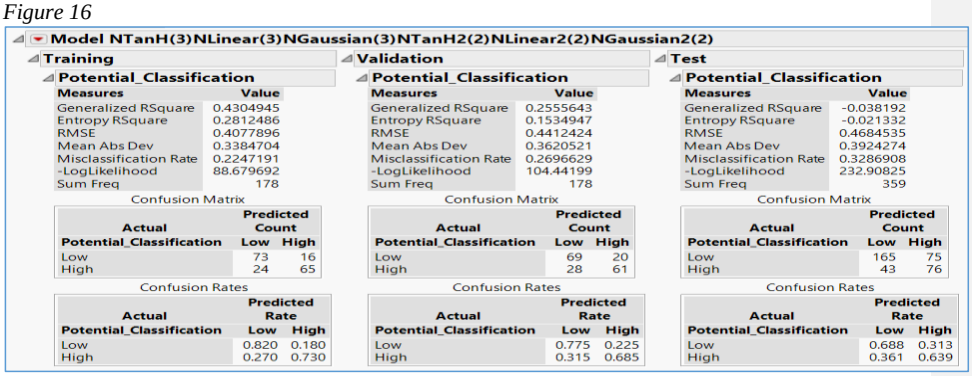
Figure 15



- The ROC index is 0.7049 for the test data which can be observed from the above graph
- The following figure shows the fit details for the Training, Test, and Validation datasets:

The work presented here is our team's and our team's work alone.

Figure 16



	True Positives	True Negatives	False Positives	False Negatives
Training	73.03%	82.02%	17.98%	26.97%
Validation	68.54%	77.53%	22.47%	31.46%
Test	63.87%	68.75%	31.25%	36.13%

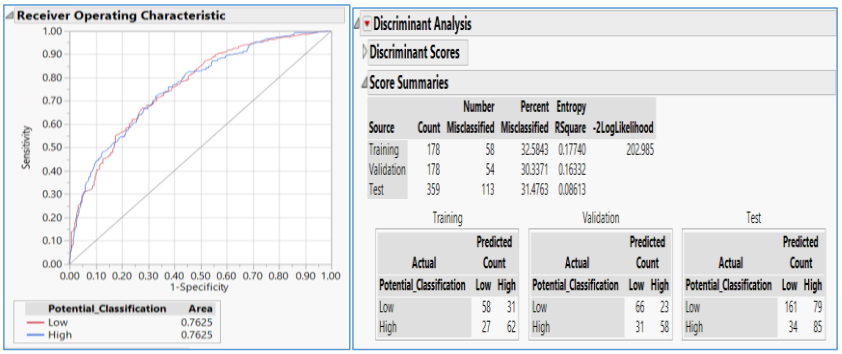
C. Discriminant Analysis

Approach: Analyze -> Multivariate Methods -> Discriminant Analysis

- Linear Discriminant Analysis (LDA) extracts the discriminant function in which the coefficients of linear combination tell us about the discriminative ability of a variable.

Results:

- We placed the continuous independent variables in Y (Covariates) and the categorical variable in X (Categories)
- Similar to the Decision Trees model, we have considered multiple cut-off values for converting the predicted probabilities to binary outcome and selected 0.6 as the optimal threshold value
- The ROC curve and model fit details are as shown:
  Figure 17



The work presented here is our team's and our team's work alone.

- The overall accuracy of the model on test data is 68.52% and the ROC index is 0.7625

	True Positives	True Negatives	False Positives	False Negatives
Training	69.66%	65.17%	34.83%	30.34%
Validation	65.17%	74.16%	25.84%	34.83%
Test	71.43%	67.08%	32.92%	28.57%

good summary tables – easy to follow

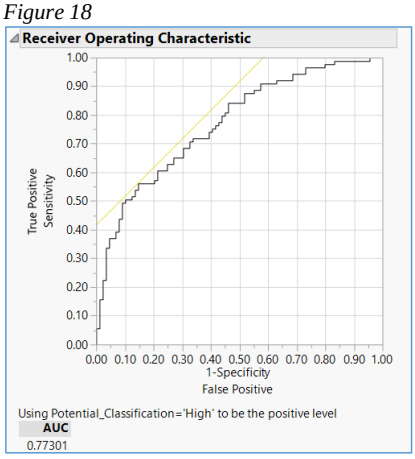
D. Logistic Regression

Approach: Analyze -> Fit Model

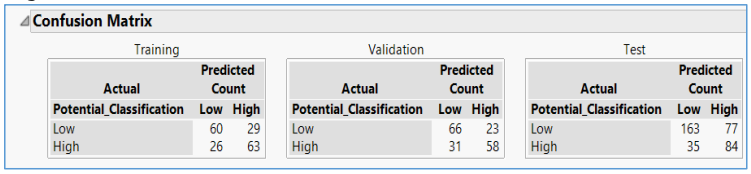
- This approach fits probabilities for the response levels using a logistic function and it is considered to better predict the groups when it is not reasonable to assume that independent variables are normally distributed

Results:

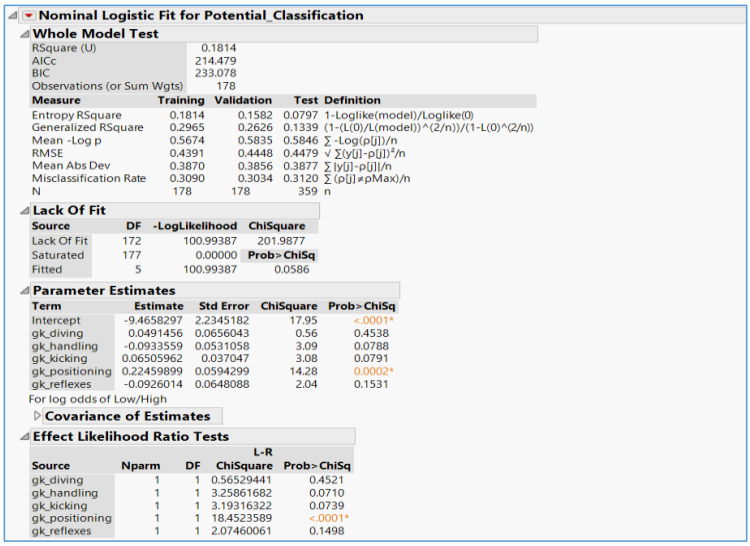
- The optimal threshold value selected = 0.5 (using a similar approach as Decision Trees model)
- Overall accuracy of the model on test data is 68.80% and the ROC index is 0.7730:
  Figure 18



- The following figure shows the fit details for the Training, Test and Validation datasets:
  Figure 19



The work presented here is our team's and our team's work alone.



Nominal Logistic Fit for Potential_Classification

Whole Model Test

RSquare (U)	0.1814
AICc	214.479
BIC	233.078
Observations (or Sum Wgts)	178

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.1814	0.1582	0.0797	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.2965	0.2626	0.1339	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.5674	0.5835	0.5846	$\sum$ -Log(ρ[j])/n
RMSE	0.4391	0.4448	0.4479	$\sqrt{\sum(y[j]-ρ[j])^2/n}$
Mean Abs Dev	0.3870	0.3856	0.3877	$\sum$ |y[j]-ρ[j]|/n
Misclassification Rate	0.3090	0.3034	0.3120	$\sum$ (ρ[j]≠ρMax)/n
N	178	178	359	n

Lack Of Fit

Source	DF	-LogLikelihood	ChiSquare
Lack Of Fit	172	100.99387	201.9877
Saturated	177	0.00000	Prob>ChiSq
Fitted	5	100.99387	0.0586

Parameter Estimates

Term	Estimate	Std Error	ChiSquare	Prob>ChiSq
Intercept	-9.4658297	2.2345182	17.95	<.0001*
gk_diving	0.0491456	0.0656043	0.56	0.4538
gk_handling	-0.0933559	0.0531058	3.09	0.0788
gk_kicking	0.06505962	0.037047	3.08	0.0791
gk_positioning	0.22459899	0.0594299	14.28	0.0002*
gk_reflexes	-0.0926014	0.0648088	2.04	0.1531

For log odds of Low/High

Covariance of Estimates

Effect Likelihood Ratio Tests

Source	Nparm	DF	L-R ChiSquare	Prob>ChiSq
gk_diving	1	1	0.56529441	0.4521
gk_handling	1	1	3.25861682	0.0710
gk_kicking	1	1	3.19316322	0.0739
gk_positioning	1	1	18.4523589	<.0001*
gk_reflexes	1	1	2.07460061	0.1498

	True Positives	True Negatives	False Positives	False Negatives
Training	70.79%	67.42%	32.58%	29.21%
Validation	65.17%	74.16%	25.84%	34.83%
Test	70.59%	67.92%	32.08%	29.41%

E. Ensemble Model

Approach:

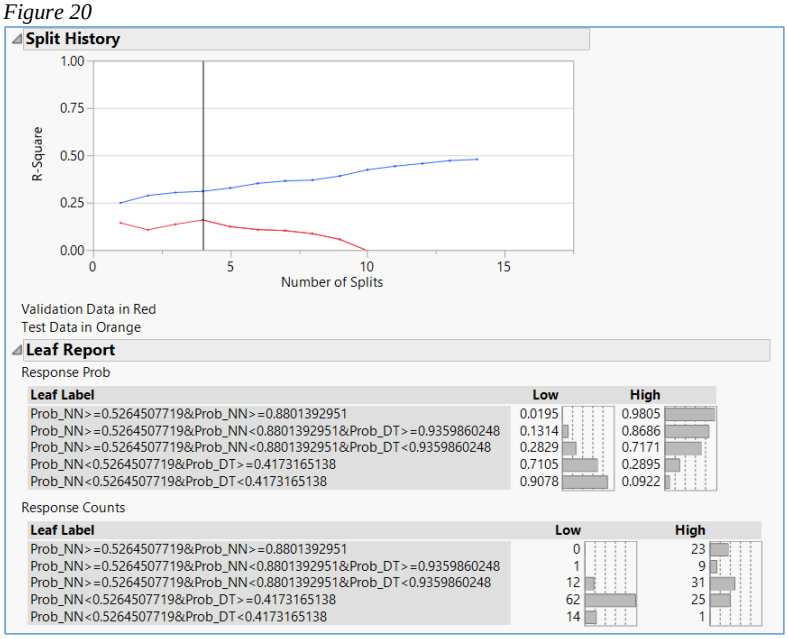
- We had saved the probability formula for models performed (Decision Trees, Logistic Regression, Neural Networks and Discriminant Analysis). In Ensemble model, we gave all the predicted probabilities corresponding to Probability = High for the above models as input to the decision tree and created an ensemble model.

Results:

- The model's true positive rate 59.66% is ~~very~~ less thank ~~when compared to~~ other models
- The optimal threshold value selected = 0.5 (using a similar approach as Decision Trees model)
- Overall model accuracy on test data is 67.4%
- Even though the model accurately predicts true negatives with a rate of 71.25%, the number of false negatives (give %) is ~~considerably~~ high

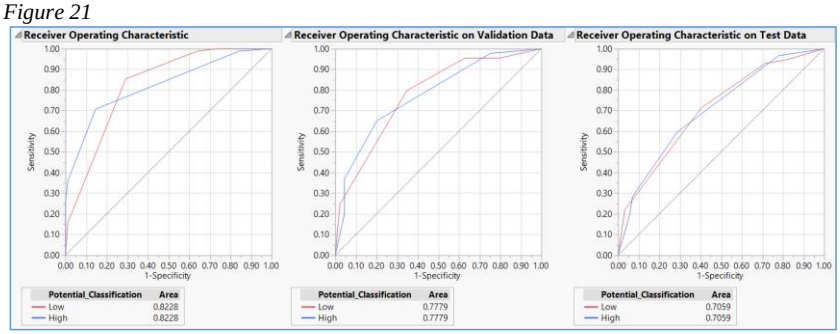
The work presented here is our team's and our team's work alone.

Figure 20



- It can be seen from the ROC curve that the model is predicting less false positives and false negatives for the training dataset but the accuracy drops for the validation and the test datasets: [should present %'s – specific values aid clarity --]

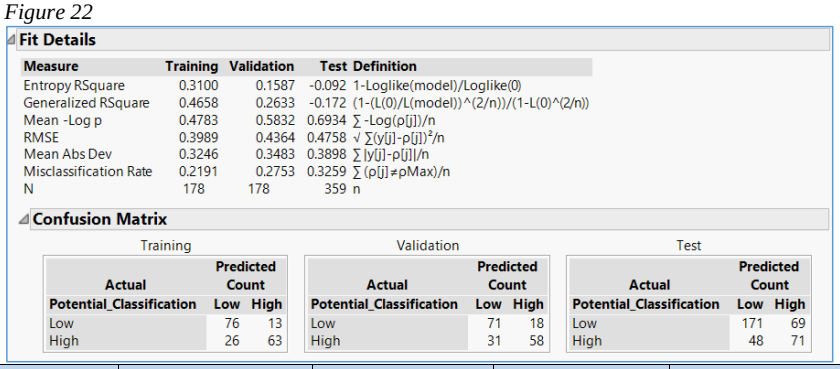
Figure 21



- Following figure shows us the fit details for the training, validation, and test datasets

The work presented here is our team's and our team's work alone.

Figure 22



Fit Details

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.3100	0.1587	-0.092	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.4658	0.2633	-0.172	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.4783	0.5832	0.6934	∑ -Log(ρ[j])/n
RMSE	0.3989	0.4364	0.4758	√ ∑(y[j]-ρ[j])²/n
Mean Abs Dev	0.3246	0.3483	0.3898	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.2191	0.2753	0.3259	∑ (ρ[j]≠ρMax)/n
N	178	178	359	n

Confusion Matrix

Training

Actual	Predicted Count	
Potential_Classification	Low	High
Low	76	13
High	26	63

Validation

Actual	Predicted Count	
Potential_Classification	Low	High
Low	71	18
High	31	58

Test

Actual	Predicted Count	
Potential_Classification	Low	High
Low	171	69
High	48	71

	True Positives	True Negatives	False Positives	False Negatives
Training	70.79%	85.39%	14.61%	29.21%
Validation	65.17%	79.78%	20.22%	34.83%
Test	59.66%	71.25%	28.75%	40.34%

Calculation of Model Cost:

Test	Model Complexity	Overall Accuracy	False Positives	Cost of False Positives	False Negatives	Cost of False Negatives	Total Cost
Baseline low	Simple	66.98%	0	0	119	5,959,000	5,959,000
Baseline high	Simple	33.14%	240	39,600,000	0	0	39,600,000
Decision Trees	Simple	66.48%	50	8,250,000	55	2,750,000	11,000,000
Neural Networks	Complex	66.31%	75	12,375,000	43	2,150,000	14,525,000
Discriminant Analysis	Moderate	69.26%	79	13,035,000	85	4,250,000	17,285,000
Logistic Regression	Simple	69.26%	72	11,880,000	35	1,750,000	13,630,000
Ensemble Model	Moderate	65.46%	69	11,385,000	48	2,400,000	13,785,000

The work presented here is our team's and our team's work alone.

Recommendations for Model Selection:

- Per the business setting, we would recommend using the Decision tree~~s~~ model as it is helping us minimize the total cost of misclassifications. Furthermore, tree structure is easy to understand and offers a comparable accuracy with respect to other models on test data
- However, if the management is interested in having maximum accuracy of true predictions then we would recommend using the Logistic regression model which has the highest model accuracy on test data and the cost associated with the incorrectly predicting the potential of a goalkeeper as "High" is also low. Although this cost is minimum for Ensemble model, it is highly complex and hence not recommended

ok

Modeling for Non-Goalkeepers

Data Stratification

Approach:

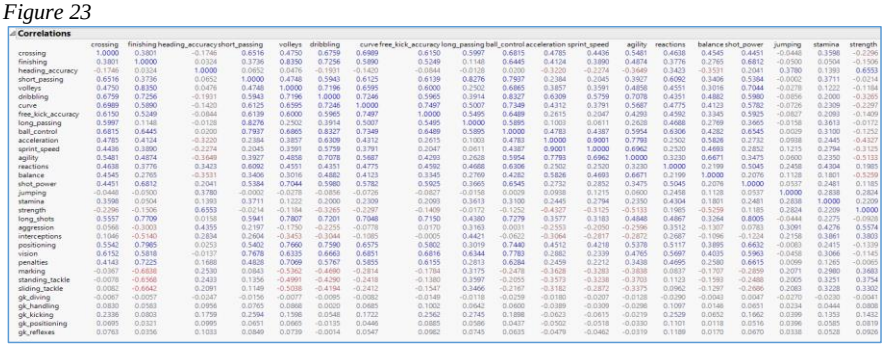
- For stratifying the data, we have applied the same approach as mentioned in "Data Stratification" for goalkeepers
- After the exploration, variable reduction and stratification, the dataset now has 8,523 rows which can be used for modeling. We need to find the best model which will predict whether a player is high or low potential to grow. Following are the variables that will be used in building the model for 'Non-GK' players
    - Target variable -
        - Potential_Classification
    - Independent explanatory variables –
        - Dribbling_PC1_std
        - Dribbling_PC2_std
        - Shooting_PC_std
        - Passing_PC1_std
        - Passing_PC2_std
        - Defending_PC_std
        - Aggression_std
        - jumping_std
        - stamina_std
        - strength_std
        - reactions_std
        - balance_std
        - heading_accuracy_std

Multivariate Correlations

Approach: Analyze -> Multivariate Methods -> Multivariate

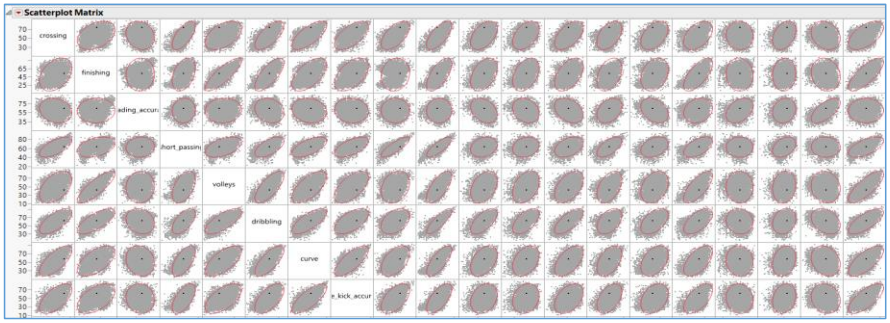
- We performed a correlation analysis on all continuous variables to understand relations between different variables
- This helped us reduce the number of variables for modeling. Following are the correlation matrix and scatterplot matrix :

Figure 23



Correlations

	crossing	finishing	heading_accuracy	short_passing	volleys	dribbling	curve	free_kick_accuracy	long_passing	ball_control	acceleration	sprint_speed	agility	reactions	balance	shot_power	jumping	stamina	strength
crossing	1.0000	0.3801	-0.1746	0.6516	0.4750	0.6759	0.6989	0.6150	0.5997	0.6815	0.4785	0.4436	0.5481	0.4638	0.4545	0.4451	-0.0446	0.3598	-0.2296
finishing	0.3801	1.0000	0.0324	0.3736	0.8350	0.7256	0.5890	0.5249	0.1148	0.6445	0.4124	0.3890	0.4874	0.3776	0.2765	0.6812	-0.0500	0.0504	-0.1506
heading_accuracy	-0.1746	0.0324	1.0000	0.0652	0.0476	-0.1931	-0.1420	-0.0844	-0.0128	0.0200	-0.3220	-0.2274	-0.3649	0.3423	-0.3531	0.2041	0.3780	0.1393	0.6553
short_passing	0.6516	0.3736	0.0652	1.0000	0.4748	0.5943	0.6125	0.6139	0.8276	0.7937	0.2384	0.2045	0.3927	0.6092	0.3406	0.5394	-0.0002	0.3711	-0.0214
volleys	0.4750	0.8350	0.0476	0.4748	1.0000	0.7196	0.6595	0.6000	0.2502	0.6865	0.3857	0.3591	0.4858	0.4551	0.3016	0.7044	-0.0278	0.1222	-0.1184
dribbling	0.6759	0.7256	-0.1931	0.5943	0.7196	1.0000	0.7246	0.5965	0.3914	0.8327	0.6309	0.5759	0.7078	0.4351	0.4882	0.5980	-0.0856	0.2000	-0.3265
curve	0.6989	0.5890	-0.1420	0.6125	0.6595	0.7246	1.0000	0.7487	0.5007	0.7349	0.4312	0.3791	0.5687	0.4775	0.4123	0.5782	-0.0726	0.2309	-0.2297
free_kick_accuracy	0.6150	0.5249	-0.0844	0.6139	0.6000	0.5965	0.7497	1.0000	0.5495	0.6489	0.2615	0.2047	0.4293	0.4592	0.3345	0.5925	-0.0827	0.2093	-0.1409
long_passing	0.5997	0.1148	-0.0128	0.8276	0.2502	0.3914	0.5007	0.5495	1.0000	0.5895	0.1003	0.0611	0.2628	0.4688	0.2769	0.3665	-0.0158	0.3613	-0.0172
ball_control	0.6815	0.6445	0.0200	0.7937	0.6865	0.8327	0.7349	0.6489	0.5895	1.0000	0.4783	0.4387	0.5954	0.6306	0.4282	0.6545	0.0029	0.3100	-0.1252
acceleration	0.4785	0.4124	-0.3220	0.2384	0.3857	0.6309	0.4312	0.2615	0.1003	0.4783	1.0000	0.9001	0.7793	0.2502	0.5826	0.2732	0.0938	0.2445	-0.4327
sprint_speed	0.4436	0.3890	-0.2274	0.2045	0.3591	0.5759	0.3791	0.2047	0.0611	0.4387	0.9001	1.0000	0.6962	0.2520	0.4693	0.2832	0.1215	0.2794	-0.3125
agility	0.5481	0.4874	-0.3649	0.3927	0.4858	0.7078	0.5687	0.4293	0.2628	0.5954	0.7793	0.6962	1.0000	0.3230	0.6671	0.3475	0.0600	0.2350	-0.5133
reactions	0.4638	0.3776	0.3423	0.6092	0.4551	0.4351	0.4775	0.4592	0.4688	0.6306	0.2502	0.2520	0.3230	1.0000	0.2199	0.5045	0.2458	0.4304	0.1985
balance	0.4545	0.2765	-0.3531	0.3406	0.3016	0.4882	0.4123	0.3345	0.2769	0.4282	0.5826	0.4693	0.6671	0.2199	1.0000	0.2076	0.1128	0.1801	-0.5259
shot_power	0.4451	0.6812	0.2041	0.5384	0.7044	0.5980	0.5782	0.5925	0.3665	0.6545	0.2732	0.2852	0.3475	0.5045	0.2076	1.0000	0.0337	0.2481	0.1185
jumping	-0.0446	-0.0500	0.3780	-0.0002	-0.0278	-0.0856	-0.0726	-0.0827	-0.0158	0.0029	0.0938	0.1215	0.0600	0.2458	0.1128	0.0337	1.0000	0.2838	0.2824
stamina	0.3598	0.0504	0.1393	0.3711	0.1222	0.2000	0.2309	0.2093	0.3613	0.3100	0.2445	0.2794	0.2350	0.4304	0.1801	0.2481	0.2838	1.0000	0.2209
strength	-0.2296	-0.1506	0.6553	-0.0214	-0.1184	-0.3265	-0.2297	-0.1409	-0.0172	-0.1252	-0.4327	-0.3125	-0.5133	0.1985	-0.5259	0.1185	0.2824	0.2209	1.0000
long_shots	0.5557	0.7709	0.0158	0.5941	0.7807	0.7201	0.7948	0.7150	0.4380	0.7279	0.3577	0.3183	0.4848	0.4867	0.3264	0.8005	-0.0444	0.2275	-0.0928
aggression	0.0568	-0.3003	0.4355	0.2197	-0.1750	-0.2255	-0.0778	0.0170	0.3163	0.0031	-0.2553	-0.2050	-0.2596	0.3512	-0.1307	0.0783	0.3091	0.4276	0.5574
interceptions	0.1046	-0.5140	0.2834	0.2604	-0.3453	-0.3044	-0.1085	-0.0005	0.4421	-0.0622	-0.3064	-0.2817	-0.2672	0.2987	-0.1096	-0.1224	0.2158	0.3861	0.3803
positioning	0.5542	0.7985	0.0253	0.5402	0.7660	0.7590	0.6575	0.5802	0.3019	0.7440	0.4512	0.4210	0.5378	0.5117	0.3895	0.6632	-0.0083	0.2415	-0.1319
vision	0.6152	0.5818	-0.0137	0.7678	0.6335	0.6663	0.6851	0.6816	0.6344	0.7783	0.2882	0.2339	0.4765	0.5697	0.4035	0.5963	-0.0458	0.3066	-0.1145
penalties	0.4143	0.7225	0.1668	0.4828	0.7069	0.5767	0.5855	0.6155	0.2813	0.6284	0.2459	0.2212	0.3438	0.4695	0.2580	0.6615	0.0099	0.1265	-0.0065
marking	-0.0367	-0.6838	0.2530	0.0843	-0.5362	-0.4690	-0.2814	-0.1784	0.3175	-0.2478	-0.3628	-0.3283	-0.3838	0.0837	-0.1707	-0.2859	0.2071	0.2980	0.3683
standing_tackle	-0.0078	-0.8368	0.2433	0.1356	-0.4991	-0.4290	-0.2418	-0.1380	0.3597	-0.2055	-0.3573	-0.3238	-0.3703	0.1123	-0.1593	-0.2488	0.2005	0.3251	0.3754
sliding_tackle	0.0082	-0.6642	0.2091	0.1149	-0.5038	-0.4194	-0.2412	-0.1547	0.3466	-0.2167	-0.3162	-0.2872	-0.3375	0.0962	-0.1297	-0.2686	0.2083	0.3228	0.3302
gk_diving	-0.0067	-0.0057	-0.0247	-0.0156	-0.0077	-0.0095	-0.0082	-0.0149	-0.0118	-0.0259	-0.0180	-0.0207	-0.0128	-0.0290	-0.0043	0.0047	-0.0270	-0.0230	-0.0041
gk_handling	0.0830	0.0583	0.0956	0.0765	0.0868	0.0020	0.0685	0.1002	0.0642	0.0690	-0.0189	-0.0309	-0.0286	0.1097	0.0148	0.0651	0.0234	0.0444	0.0808
gk_kicking	0.2336	0.0803	0.1759	0.2594	0.1598	0.0548	0.1722	0.2562	0.2745	0.1898	-0.0623	-0.0615	-0.0219	0.2529	0.0652	0.1962	0.0399	0.1353	0.1432
gk_positioning	0.0695	0.0321	0.0995	0.0651	0.0665	-0.0135	0.0446	0.0885	0.0586	0.0437	-0.0502	-0.0518	-0.0330	0.1101	0.0118	0.0516	0.0396	0.0585	0.0819
gk_reflexes	0.0763	0.0356	0.1033	0.0849	0.0739	-0.0014	0.0547	0.0982	0.0745	0.0635	-0.0479	-0.0462	-0.0319	0.1189	0.0170	0.0670	0.0338	0.0528	0.0926



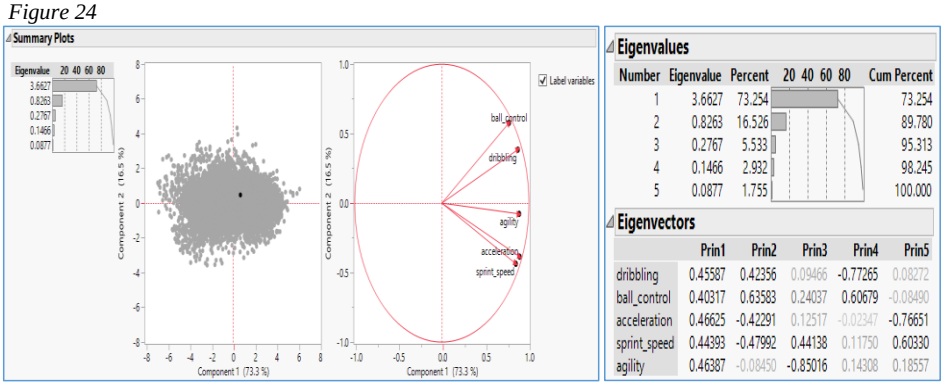
- Based on our observation of the results and understanding of the parameters that define various skills of a soccer player, we identified groups of variables that capture similar information and were highly correlated with each other o
- We identified 4 such groupings of variables. We decided to perform Principal component analysis for decreasing the number of variables (thereby reducing information overlap) on the identified groups of variables while retaining most of the information ok

Principal Component Analysis

Approach: Analyze -> Multivariate Analysis -> Principal Components

I. Group 1 (Dribbling) – dribbling, ball_control, acceleration, sprint_speed, agility

Figure 24



Eigenvalues

Number	Eigenvalue	Percent	Cum Percent
1	3.6627	73.254	73.254
2	0.8263	16.526	89.780
3	0.2767	5.533	95.313
4	0.1466	2.932	98.245
5	0.0877	1.755	100.000

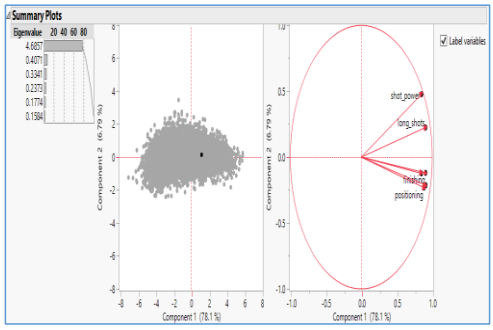
Eigenvectors

	Prin1	Prin2	Prin3	Prin4	Prin5
dribbling	0.45587	0.42356	0.09466	-0.77265	0.08272
ball_control	0.40317	0.63583	0.24037	0.60679	-0.08490
acceleration	0.46625	-0.42291	0.12517	-0.02347	-0.76651
sprint_speed	0.44393	-0.47992	0.44138	0.11750	0.60330
agility	0.46387	-0.08450	-0.85016	0.14308	0.18557

- The principal components 1 and 2 that we have obtained from the analysis cumulatively have an Eigen value of 89.78%
- We decided to retain both the principal components for modeling which captures 89.78% of the information conveyed by the five original variables – dribbling, ball_control, acceleration, sprint_speed, agility ok good approach

II. Group 2 (Shooting) – penalties, finishing, long_shots, volleys, shot_power, positioning

Figure 25



Eigenvalues

Number	Eigenvalue	Percent	Cum Percent
1	4.6857	78.095	78.095
2	0.4071	6.785	84.880
3	0.3341	5.568	90.448
4	0.2373	3.955	94.402
5	0.1774	2.957	97.359
6	0.1584	2.641	100.000

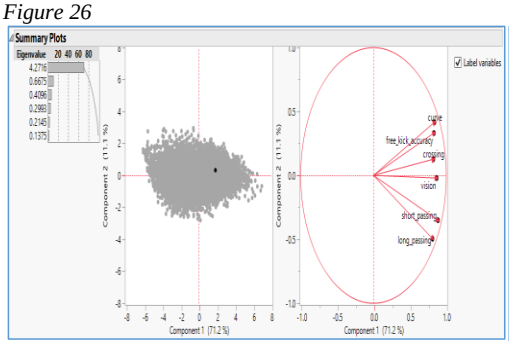
Eigenvectors

	Prin1	Prin2	Prin3	Prin4	Prin5	Prin6
penalties	0.39079	-0.19284	0.89370	-0.08792	-0.04720	0.03792
finishing	0.41993	-0.33379	-0.24281	-0.25100	0.09839	-0.76197
long_shots	0.41874	0.35298	-0.14740	0.03308	-0.82292	0.00596
volleys	0.41857	-0.18695	-0.29456	-0.55079	0.16783	0.60955
shot_power	0.39214	0.74894	0.03014	0.09175	0.51845	-0.08485
positioning	0.40820	-0.36186	-0.18150	0.78510	0.11800	0.19793

- The principal components 1 that we have obtained from the analysis has an Eigen value of 78.09%
- We decided to retain the principal components for the purpose of modeling which captures 78.09% of the effect of the six original variables – penalties, finishing, long_shots, volleys, shot_power, positioning

III. Group 3 (Passing) – crossing, short_passing, curve, freekick_accuracy, long_passing, vision

Figure 26



Eigenvalues

Number	Eigenvalue	Percent	Cum Percent
1	4.2716	71.193	71.193
2	0.6675	11.125	82.318
3	0.4096	6.826	89.144
4	0.2993	4.989	94.133
5	0.2145	3.575	97.708
6	0.1375	2.292	100.000

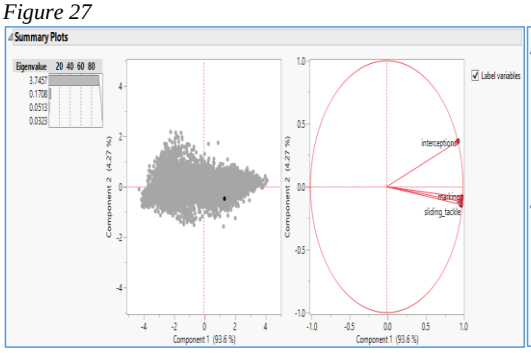
Eigenvectors

	Prin1	Prin2	Prin3	Prin4	Prin5	Prin6
crossing	0.39868	0.15562	0.81018	-0.13884	0.37570	-0.00417
short_passing	0.42881	-0.42842	-0.08062	-0.09830	-0.14862	-0.77093
curve	0.40572	0.50949	0.03719	-0.07505	-0.74849	0.09251
free_kick_accuracy	0.40195	0.40608	-0.35790	0.60877	0.40069	-0.11953
long_passing	0.39270	-0.60594	0.07470	0.40528	-0.17117	0.52867
vision	0.42046	-0.02453	-0.44950	-0.65618	0.29440	0.32142

- The principal components 1 and 2 that we have obtained from the analysis have a cumulative Eigen value of 82.31%
- We decided to retain the principal components 1 and 2 for modeling which help us capture 82.31% of the information conveyed by of the six original variables – crossing, short_passing, curve, free_kick_accuracy, long_passing, vision good approach – good explanation

IV. Group 4 (Defending) – interceptions, marking, standing_tackle, sliding_tackle

Figure 27



Eigenvalues

Number	Eigenvalue	Percent	Cum Percent
1	3.7457	93.642	93.642
2	0.1708	4.269	97.911
3	0.0513	1.282	99.193
4	0.0323	0.807	100.000

Eigenvectors

	Prin1	Prin2	Prin3	Prin4
interceptions	0.48194	0.87170	-0.01590	0.08728
marking	0.50437	-0.28376	0.79227	0.19338
standing_tackle	0.50878	-0.20355	-0.19846	-0.81260
sliding_tackle	0.50447	-0.34377	-0.57677	0.54283

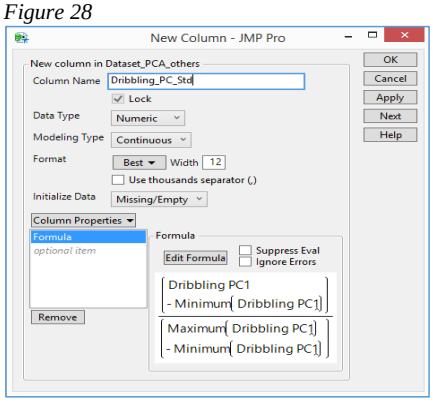
- The principal component 1 that we have obtained from the analysis has an Eigen value of 93.64%
- We decided to retain only 1 principal component for the purpose of modeling which help us capture 93.64 % of the information conveyed by the four original variables – interceptions, marking, standing_tackle, sliding_tackle

Standardization

The considered variables require standardization in this case since the principal components belong to a scale of 0,1, while the predictors belong to a scale of 1 to 100. Since the models for NonGKs have a blend of both principal components of certain predictors and original values for the rest, we standardized all the considered predictors to a scale of (0,1) good

The work presented here is our team's and our team's work alone.

Figure 28



Approach: Cols -> New Column -> Enter column name and add standardization formula

- Different continuous variables have different scales of measurement and need to be brought down to a uniform scale so that a variable doesn't dominate others while building the model
- All continuous variables are transformed on to a scale of (0,1)
- A new column is created for each variable with the suffix '_std'. An example is shown in the adjacent figure. The original columns were excluded from analysis, but not dropped since they may be used later

Algorithms Used

A. Decision Trees

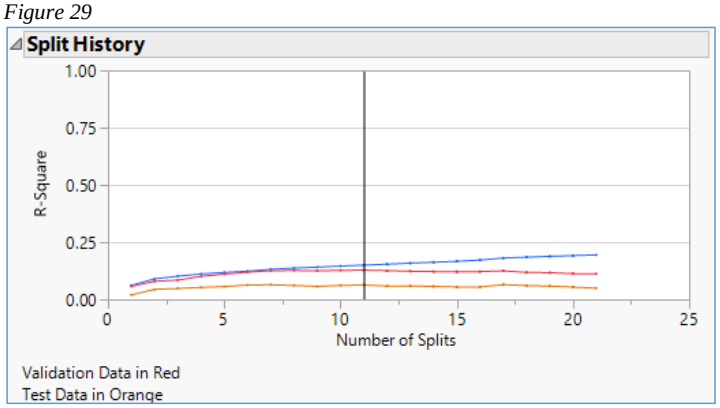
Approach: Analyze -> Modeling -> Partition

- Keep splitting till all matching records have the same output value and good classification accuracy without overfitting the model.

Results*:*

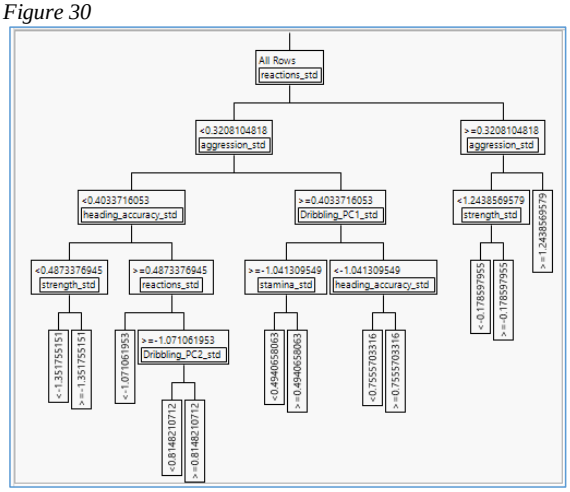
- After 12 splits, the decision tree was still trying to improve the accuracy of training data but for Validation/Test data there wasn't any significant improvement
- Hence, we stopped at 12 splits as the decision tree was trying to over fit the model and accuracy stopped increasing post that

Figure 29



- Following figure shows us the tree diagram for the decision trees. It is observed that the maximum information gain and expansion power is on the variable reactions_std

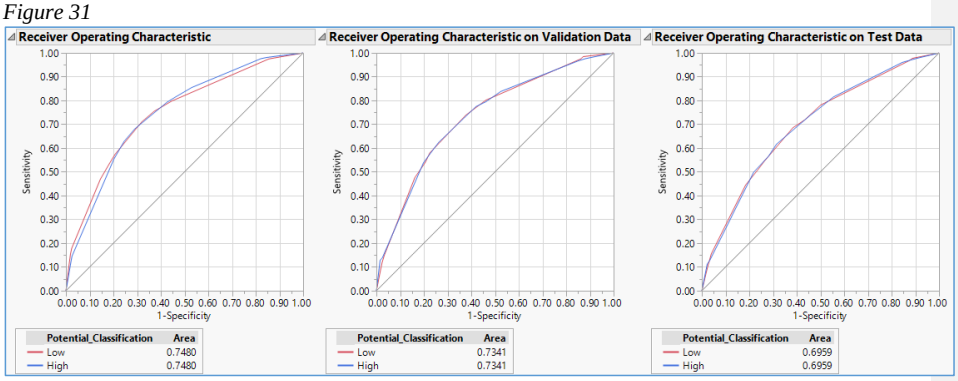
Figure 30



The work presented here is our team's and our team's work alone.

- The ROC index for the test dataset is 0.6959:

Figure 31



- The following figure shows the fit details for the Training, Testing and Validation datasets:

Figure 32

Fit Details

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.1505	0.1284	0.0636	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.2510	0.2174	0.1113	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.5888	0.6042	0.6349	∑ -Log(p[j])/n
RMSE	0.4490	0.4559	0.4709	√ ∑(y[j]-p[j])²/n
Mean Abs Dev	0.4038	0.4112	0.4247	∑ |y[j]-p[j]|/n
Misclassification Rate	0.3045	0.3199	0.3430	∑ (p[j]≠pMax)/n
N	2358	2360	3805	n

Confusion Matrix

Training

Actual	Predicted	
Potential_Classification	Low	High
Low	837	342
High	376	803

Validation

Actual	Predicted	
Potential_Classification	Low	High
Low	823	357
High	398	782

Test

Actual	Predicted	
Potential_Classification	Low	High
Low	1530	702
High	603	970

	True Positives	True Negatives	False Positives	False Negatives
Training	68.1%	70.9%	29.0%	31.8%
Validation	66.2%	69.7%	30.2%	33.7%
Test	61.6%	68.5%	31.4%	38.3%

B. Neural Networks Algorithm

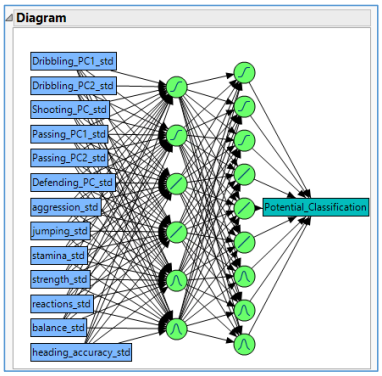
Approach: Analyze -> Modeling -> Neural

- To find out the best fit algorithm which gives better accuracy, we played with the number of functions at the two layers (Learning Layer and Classification Layer).

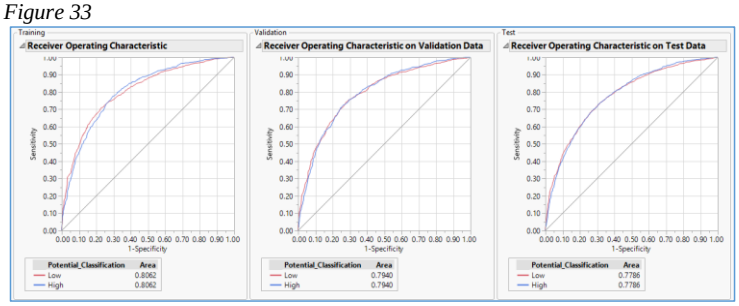
Results:

- After trying different combinations, we found that the combination of all three functions NGaussian, NTanH and NGaussian function drove the best accuracy rate for validation/test data

- Hence, we considered (NTanH(3) NLinear(3) NGaussian(3) NTanH2(2) NLinear2(2) NGaussian2 (2)) as weights, which gave best values for model evaluation metrics on test data
  Figure 33

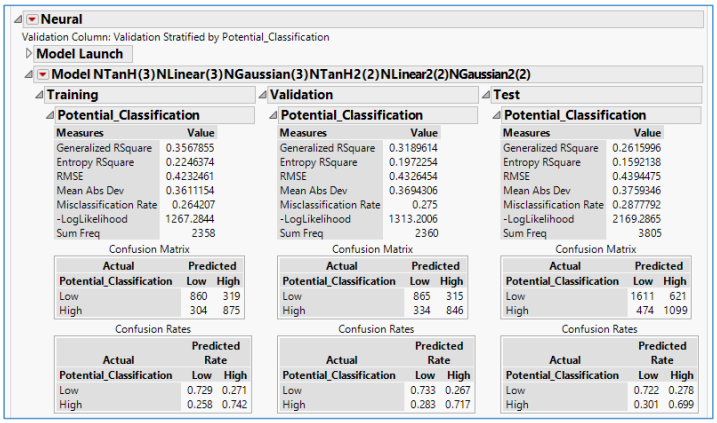


- The ROC index for the test dataset is 0.7786 as seen from the figure below:
  Figure 33



- The fit details for the training, validation and test dataset are as follows:
  Figure 34

The work presented here is our team's and our team's work alone.



Neural

Validation Column: Validation Stratified by Potential_Classification

Model Launch

Model NTanH(3)NLinear(3)NGaussian(3)NTanH2(2)NLinear2(2)NGaussian2(2)

Training — **Potential_Classification**

Measures	Value
Generalized RSquare	0.3567855
Entropy RSquare	0.2246374
RMSE	0.4232461
Mean Abs Dev	0.3611154
Misclassification Rate	0.264207
-LogLikelihood	1267.2844
Sum Freq	2358

Confusion Matrix

Actual Potential_Classification	Predicted Low	Predicted High
Low	860	319
High	304	875

Confusion Rates

Actual Potential_Classification	Predicted Rate Low	Predicted Rate High
Low	0.729	0.271
High	0.258	0.742

Validation — **Potential_Classification**

Measures	Value
Generalized RSquare	0.3189614
Entropy RSquare	0.1972254
RMSE	0.4326454
Mean Abs Dev	0.3694306
Misclassification Rate	0.275
-LogLikelihood	1313.2006
Sum Freq	2360

Confusion Matrix

Actual Potential_Classification	Predicted Low	Predicted High
Low	865	315
High	334	846

Confusion Rates

Actual Potential_Classification	Predicted Rate Low	Predicted Rate High
Low	0.733	0.267
High	0.283	0.717

Test — **Potential_Classification**

Measures	Value
Generalized RSquare	0.2615996
Entropy RSquare	0.1592138
RMSE	0.4394475
Mean Abs Dev	0.3759346
Misclassification Rate	0.2877792
-LogLikelihood	2169.2865
Sum Freq	3805

Confusion Matrix

Actual Potential_Classification	Predicted Low	Predicted High
Low	1611	621
High	474	1099

Confusion Rates

Actual Potential_Classification	Predicted Rate Low	Predicted Rate High
Low	0.722	0.278
High	0.301	0.699

	True Positives	True Negatives	False Positives	False Negatives
Training	74.2%	72.9%	27.1%	25.8%
Validation	71.7%	73.3%	26.7%	28.3%
Test	69.9%	72.2%	27.8%	30.1%

C. Discriminant Analysis

Approach: Analyze -> Multivariate Methods -> Discriminant Analysis

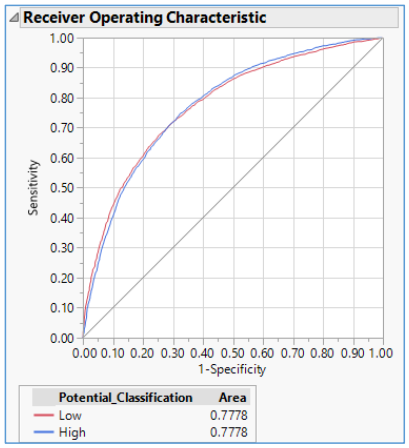
- Linear Discriminant Analysis (LDA) extracts the discriminant function in which the coefficients of linear combination tell us about the discriminative ability of a variable.

Results:

- We placed the continuous independent variables in Y (Covariates) and the categorical variable in X (Categories)
- The model helped us to predict a low number of false positives and false negatives. It also provided a good accuracy for true positives and true negatives at the same time
- The ROC index for the test dataset is 0.7778:

Figure 35

The work presented here is our team's and our team's work alone.



- The fit details for the training, validation and test dataset are as follows:

Figure 36

Score Summaries

Source	Count	Number Misclassified	Percent Misclassified	Entropy RSquare	-2LogLikelihood
Training	2358	666	28.2443	0.18002	2680.43
Validation	2360	674	28.5593	0.18978	
Test	3805	1125	29.5664	0.15372	

Training

Actual	Predicted	
Potential_Classification	Low	High
Low	853	326
High	340	839

Validation

Actual	Predicted	
Potential_Classification	Low	High
Low	867	313
High	361	819

Test

Actual	Predicted	
Potential_Classification	Low	High
Low	1604	628
High	497	1076

The work presented here is our team's and our team's work alone.

	True Positives	True Negatives	False Positives	False Negatives
Training	71.1%	72.3%	27.6%	28.8%
Validation	69.4%	73.4%	26.5%	30.5%
Test	68.4%	71.8%	28.1%	31.5%

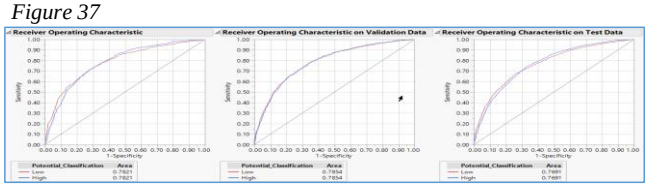
D. Logistic Regression

Approach: Analyze -> Fit Model

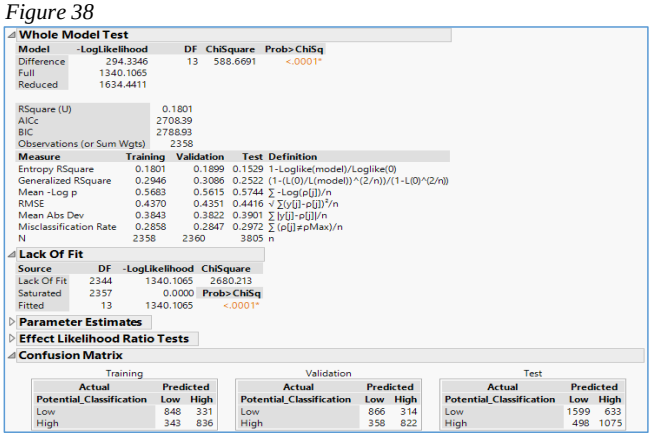
- This approach fits probabilities for the response levels using a logistic function and it is considered to better predict the groups when it is not reasonable to assume that independent variables are normally distributed.

Results:

- We placed the categorical variable in the Y Column and added the independent continuous variables as dependent variables
- The model performed better in terms of predicting the true negatives as compared to the true positives
- The ROC index for the test dataset is 0.7691:
  Figure 37



- The fit details for the training, test and validation dataset are as follows
  Figure 38



Whole Model Test

Model	-LogLikelihood	DF	ChiSquare	Prob>ChiSq
Difference	294.3346	13	588.6691	<.0001*
Full	1340.1065			
Reduced	1634.4411			

RSquare (U)	0.1801
AICc	2708.39
BIC	2788.93
Observations (or Sum Wgts)	2358

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.1801	0.1899	0.1529	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.2946	0.3086	0.2522	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.5683	0.5615	0.5744	∑ -Log(p[j])/n
RMSE	0.4370	0.4351	0.4416	√ ∑(y[j]-p[j])²/n
Mean Abs Dev	0.3843	0.3822	0.3901	∑ |y[j]-p[j]|/n
Misclassification Rate	0.2858	0.2847	0.2972	∑ (p[j]≠pMax)/n
N	2358	2360	3805	n

Lack Of Fit

Source	DF	-LogLikelihood	ChiSquare
Lack Of Fit	2344	1340.1065	2680.213
Saturated	2357	0.0000	Prob>ChiSq
Fitted	13	1340.1065	<.0001*

Parameter Estimates

Effect Likelihood Ratio Tests

Confusion Matrix

Training

Actual Potential_Classification	Predicted Low	Predicted High
Low	848	331
High	343	836

Validation

Actual Potential_Classification	Predicted Low	Predicted High
Low	866	314
High	358	822

Test

Actual Potential_Classification	Predicted Low	Predicted High
Low	1599	633
High	498	1075

The work presented here is our team's and our team's work alone.

	True Positives	True Negatives	False Positives	False Negatives
Training	70.9%	71.9%	28.0%	29.0%
Validation	69.6%	73.3%	26.6%	30.3%
Test	68.3%	71.6%	28.3%	31.6%

Calculation of Model Cost:

Test	Complexity	Overall Accuracy	False Positives	Cost of False Positives ($)	False Negative	Cost of False Negatives ($)	Total Cost ($)
Baseline low	Simple	58.65%	0	0	1,573	75,504,000	75,504,000
Baseline high	Simple	41.34%	2,232	669,600,000	0	0	669,600,000
Decision Trees	Simple	65.05%	702	210,600,000	603	28,944,000	239,544,000
Neural Networks	Complex	71.05%	621	186,300,000	474	22,752,000	209,052,000
Discriminant Analysis	Moderate	70.1%	628	188,400,000	497	23,856,000	212,256,000
Logistic Regression	Simple	68.35%	633	189,900,000	498	23,904,000	213,804,000

Recommendations

- Per the business setting we would recommend using the Neural Networks model as it is helping us minimize the overall cost associated with the misclassifications. Furthermore, out of all the models, it is the most has the highest accuracy in predicting the potential of players. proofread?
- However, if the management is interested in having a simple to understand model, we ~~will~~ recommend using the Logistic regression model which ~~perform~~performs~~ing~~ well in terms of accuracy as well as does not incur very high costs. provide specifics

Formatted: Highlight

The work presented here is our team's and our team's work alone.

REGRESSION MODELING

Modeling for Non-Goalkeepers

Methodology

We ran multiple models and after comparing the output summary statistics and error distributions, we took a decision on which model to be selected. The maximum difference between the actual and predicted values for the overall_rating which is acceptable by the management is 8 units. However, the cost of overestimation of ratings is different from the cost of underestimating them, we have adopted the following approach for calculating the two types of costs that can be incurred:

1) Underestimation:
    a. Occurs when overall_rating(actual) – overall_rating(predicted) > 8
    b. Per unit cost (decided by business): $48,000
    c. Cost due to underestimating: $48,000 * (error – 8), where error = overall_rating(actual) – overall_rating(predicted)
2) Overestimation:
    a. Occurs when overall_rating(predicted) – overall_rating(actual) > 8
    b. Per unit cost (decided by business): $300,000
    c. Cost due to overestimating: $300,000 * (error – 8), where error = overall_rating(predicted) – overall_rating(actual)

As evident from the per unit costs, the cost of overestimating a player's rating is comparatively more severe from the management's perspective~~.~~ than underestimating. Hence, in addition to the evaluation metrics mentioned above, we have also considered the two types of costs for finalizing the model.

Rounding off the Predicted values

The values of the provided overall_rating provided are in whole numbers. Also, while referring to the official EASports FIFA website, we realized that the ratings are always provided in whole numbers. Moreover, from the management's point of view, they would like to consider as many footballers as possible who satisfy their requirements. Keeping all these points in perspective, we have rounded off the predicted overall_rating values in following manner:

- If overall_rating(predicted) > overall_rating(actual) -> round off to the largest number less than or equal to overall_rating(actual)
- If overall_rating(predicted) < overall_rating(actual) -> round off to the smallest number greater than or equal to overall_rating(actual)

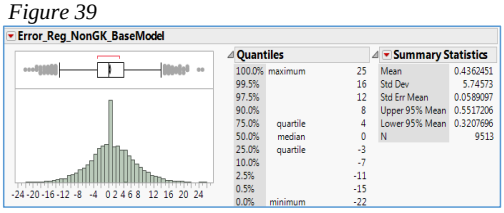
why do this? you are performing a continuous estimation so errors are also continuous – why round up or down?

Algorithms Used

A. Reg_NonGK_BaseModel: Baseline model

1) The baseline predicted value is the average value of the target variable "overall_rating". The baseline predictions will help us ascertain the value added by the predictions generated by the subsequent models
2) The rounded off predictions are stored in the column "Reg_Pred_NonGK_BaseModel_Rounded", while the error values (overall_rating – Reg_Pred_NonGK_BaseModel_Rounded) are stored in "Error_Reg_NonGK_BaseModel"
3) The overall error distribution is as follows:

Figure 39



Observations

- Number of instances related to underestimation: 245
- Number of instances related to overestimation: 157
- Total cost of underestimation: $39.50 million
- Total cost of overestimation: $179.10 million

B. Reg_NonGK_Model1: Standard least squares regression

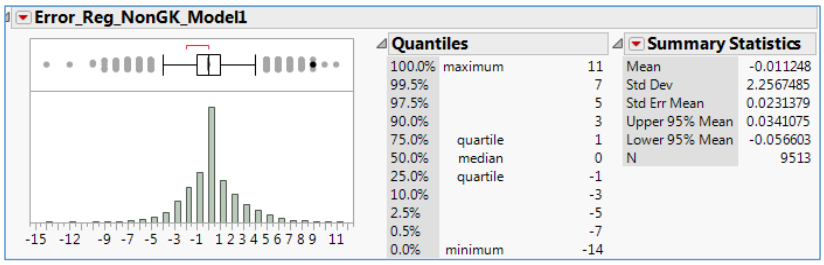
1) Select Analyze -> Fit Model. After selecting "overall_rating", a continuous variable as target, we selected Standard least squares in Personality. All the continuous predictors were considered for building the model
2) The rounded off predictions for this model are saved in the column "Reg_Pred_NonGK_Model1_Rounded" and the errors are stored in the column "Error_Reg_NonGK_Model1"
3) Equation to depict how the target variable is related to the predictors:

Figure 40

overall_rating = 7.53561727871792 + -0.00319533991251211 * crossing + 0.0138311983749624 *finishing + 0.0976173006790447 * heading_accuracy + 0.0481948946116517 *short_passing + 0.0106854728839567 *volleys + -0.0390886606490738 * dribbling +0.00683099176651877 * curve + 0.00902598503971721 * free_kick_accuracy +0.0070786619453161 * long_passing + 0.269013016132616 * ball_control +0.0511302209308127 * acceleration + 0.0555915079538262 * sprint_speed + -0.0199928670616339 * agility + 0.29470027624035 * reactions + .0285077891699038 * balance + 0.0234780793251582 * shot_power + 0.0074438974133366 * jumping +-0.0224786382042397 * stamina + 0.0723870715899864 * strength + -0.041998789640976 * long_shots + 0.00262984769524374 * aggression + 0.028400 1974386725 *interceptions + -0.0315724761929019 * positioning + 0.0089783885036613 * vision + 0.0371577688044146 * penalties + 0.0302287187615499 * marking+0.0270765506074353 *standing_tackle + -0.0410107510299523 * sliding_tackle

4) Overall error distribution and Summary of Fit:

Figure 41



The work presented here is our team's and our team's work alone.

Summary of Fit	
RSquare	0.825311
RSquare Adj	0.823798
Root Mean Square Error	2.531553
Mean of Response	67.04629
Observations (or Sum Wgts)	3262

Crossvalidation			
Source	RSquare	RASE	Freq
Training Set	0.8253	2.5203	3262
Validation Set	0.8133	2.6914	3343
Test Set	0.8199	2.5997	2908

these values are calculated without rounding

5) Parameter Estimates:

Figure 42

Parameter Estimates				
Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	7.5356173	0.699638	10.77	<.0001*
crossing	-0.003195	0.006234	-0.51	0.6083
finishing	0.0138312	0.00753	1.84	0.0663
heading_accuracy	0.0976173	0.006463	15.10	<.0001*
short_passing	0.0481949	0.012684	3.80	0.0001*
volleys	0.0106855	0.006088	1.76	0.0793
dribbling	-0.039089	0.009313	-4.20	<.0001*
curve	0.006831	0.005978	1.14	0.2532
free_kick_accuracy	0.009026	0.005734	1.57	0.1156
long_passing	0.0070787	0.008992	0.79	0.4312
ball_control	0.269013	0.013087	20.56	<.0001*
acceleration	0.0511302	0.01257	4.07	<.0001*
sprint_speed	0.0555915	0.011609	4.79	<.0001*
agility	-0.019993	0.008395	-2.38	0.0173*
reactions	0.2947003	0.009472	31.11	<.0001*
balance	0.0285078	0.006886	4.14	<.0001*
shot_power	0.0234781	0.007101	3.31	0.0010*
jumping	0.0074439	0.005847	1.27	0.2031
stamina	-0.022479	0.006844	-3.28	0.0010*
strength	0.0723871	0.007064	10.25	<.0001*
long_shots	-0.041999	0.00715	-5.87	<.0001*
aggression	0.0026298	0.005797	0.45	0.6501
interceptions	0.0284002	0.006775	4.19	<.0001*
positioning	-0.031572	0.006995	-4.51	<.0001*
vision	0.0089784	0.00773	1.16	0.2455
penalties	0.0371578	0.006388	5.82	<.0001*
marking	0.0302287	0.009359	3.23	0.0013*
standing_tackle	0.0270766	0.011341	2.39	0.0170*
sliding_tackle	-0.041011	0.009783	-4.19	<.0001*

Observations:
1) Significance level (α) considered: 0.02
2) The attributes with Prob>|t| greater than the considered significance level are least significant in interpreting the variance of target variable
3) Observing the error distribution statistics in **Figure 41**, we can infer that this model is less prone to error compared to the baseline model (**Figure 39**). However, whether to accept or reject this model can be decided after all the models have been analyzed
4) Number of instances of test data related to underestimation: 5
5)Number of instances of test data related to overestimation: 5
6)Total cost of underestimation: $384,000
7)Total cost of overestimation: $3.30 million

C. Reg_NonGK_Model2: Stepwise regression

1) Select Analyze -> Fit Model. After selecting "overall_rating", a continuous variable as target, we selected Stepwise Regression in Personality. The predictors considered are visible in the equation in Figure 44.
2) The rounded off predictions for this model are saved in the column "Reg_Pred_NonGK_Model2_Rounded" and the errors are stored in the column "Error_Reg_NonGK_Model2"
3) Selecting Predictors for the model:

Figure 43

SSE	DFE	RMSE	RSquare	RSquare Adj	Cp	p	AICc	BIC
21146.429	3247	2.5519806	0.8217	0.8209	67.611961	15	15386.43	15483.7

Current Estimates

Lock	Entered	Parameter	Estimate	nDF	SS	"F Ratio"	"Prob>F"
☑	☑	Intercept	7.55811324	1	0	0.000	1
☐	☐	crossing	0	1	3.283552	0.504	0.47775
☐	☐	finishing	0	1	16.86785	2.591	0.10755
☑	☑	heading_accuracy	0.10073442	1	1945.573	298.740	4.2e-64
☑	☑	short_passing	0.05337028	1	185.4228	28.471	1.02e-7
☐	☐	volleys	0	1	25.14793	3.865	0.04939
☐	☐	dribbling	0	1	86.21404	13.288	0.00027
☐	☐	curve	0	1	13.96621	2.145	0.14311
☐	☐	free_kick_accuracy	0	1	32.06377	4.929	0.02647
☐	☐	long_passing	0	1	6.59788	1.013	0.31424
☑	☑	ball_control	0.25130522	1	3184.996	489.051	5e-101
☐	☐	acceleration	0	1	45.36329	6.978	0.00829
☑	☑	sprint_speed	0.06990762	1	823.5643	126.457	8.3e-29
☐	☐	agility	0	1	22.91154	3.521	0.06069
☑	☑	reactions	0.2997397	1	6905.223	1060.286	2e-201
☑	☑	balance	0.03076231	1	162.6142	24.969	6.13e-7
☑	☑	shot_power	0.02406135	1	76.04297	11.676	0.00064
☐	☐	jumping	0	1	7.743878	1.189	0.27559
☐	☐	stamina	0	1	62.38432	9.604	0.00196
☑	☑	strength	0.0733819	1	859.5882	131.988	5.7e-30
☑	☑	long_shots	-0.0336901	1	182.2659	27.987	1.3e-7
☐	☐	aggression	0	1	1.152654	0.177	0.67404
☑	☑	interceptions	0.03300203	1	172.4934	26.486	2.81e-7
☑	☑	positioning	-0.0316893	1	161.6959	24.828	6.6e-7
☐	☐	vision	0	1	13.25209	2.035	0.15376
☑	☑	penalties	0.04840943	1	424.6763	65.208	9.4e-16
☑	☑	marking	0.03782627	1	129.3135	19.856	8.63e-6
☐	☐	standing_tackle	0	1	40.2041	6.183	0.01295
☑	☑	sliding_tackle	-0.0305706	1	98.30829	15.095	0.0001

Approach:
1) Using the "Step" feature, we keep selecting the predictors depending on their p-value (the variable with smallest p-value is of highest significance in interpreting the variance in target variable and is hence selected first)
2) The process is continued till the p-value of all the selected attributes is less than the considered significance level of 0.02

The work presented here is our team's and our team's work alone.

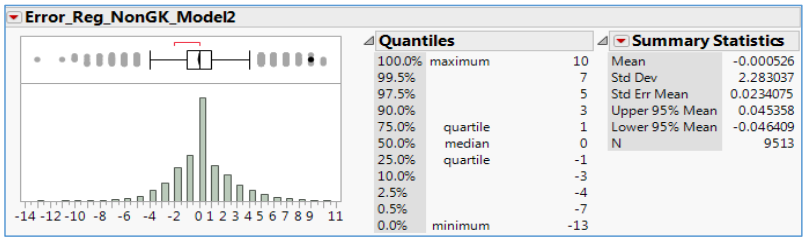
4) Equation to depict how the target variable is related to the predictors:

Figure 44

overall_rating = 7.55811323617957 + 0.100734419773773 * heading_accuracy + 0.0533702811275622 *short_passing + 0.251305219157842 * ball_control + 0.0699076238838194 *sprint_speed + 0.299739701927828 * reactions + 0.0307623148552061 * balance +0.0240613488422038 * shot_power + 0.0733818966386915 * strength +-0.0336901446976621 * long_shots + 0.0330020327292115 * interceptions +-0.0316893476065957 * positioning + 0.0484094328463976 * penalties +0.0378262691875998 * marking + -0.030570577415612 * sliding_tackle

5) Overall error distribution and Summary of Fit:

Figure 45



Summary of Fit	
RSquare	0.821712
RSquare Adj	0.820943
Root Mean Square Error	2.551981
Mean of Response	67.04629
Observations (or Sum Wgts)	3262

Crossvalidation			
Source	RSquare	RASE	Freq
Training Set	0.8217	2.5461	3262
Validation Set	0.8084	2.7265	3343
Test Set	0.8145	2.6380	2908

The work presented here is our team's and our team's work alone.

6) Parameter Estimates:

Figure 46

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	7.5581132	0.684355	11.04	<.0001*
heading_accuracy	0.1007344	0.005828	17.28	<.0001*
short_passing	0.0533703	0.010002	5.34	<.0001*
ball_control	0.2513052	0.011364	22.11	<.0001*
sprint_speed	0.0699076	0.006217	11.25	<.0001*
reactions	0.2997397	0.009205	32.56	<.0001*
balance	0.0307623	0.006156	5.00	<.0001*
shot_power	0.0240613	0.007042	3.42	0.0006*
strength	0.0733819	0.006387	11.49	<.0001*
long_shots	-0.03369	0.006368	-5.29	<.0001*
interceptions	0.033002	0.006413	5.15	<.0001*
positioning	-0.031689	0.00636	-4.98	<.0001*
penalties	0.0484094	0.005995	8.08	<.0001*
marking	0.0378263	0.008489	4.46	<.0001*
sliding_tackle	-0.030571	0.007868	-3.89	0.0001*

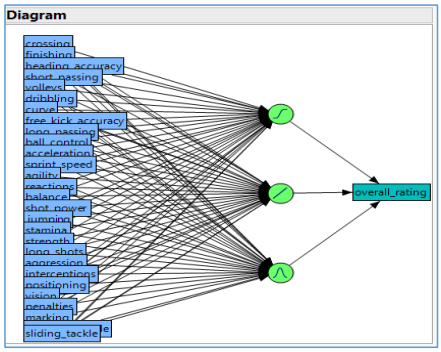
Observations:

1) The R-square value (Figure 45) indicates that this model can achieve an interpreting ability comparable to that of Reg_NonGK_Model1, where all the variables were considered (Figure 41)
2) The complexity of the model is less since only 14 out of 36 predictors were involved
3) Number of instances of test data related to underestimation: 6
4)Number of instances of test data related to overestimation: 7
5)Total cost of underestimation: $384,000
6)Total cost of overestimation: $4.20 million
7) The cost of overestimation is greater than that in case of Reg_NonGK_Model1 which is a disadvantage of this model

D. Reg_NonGK_Model3: Neural network

1) Select Analyze -> Modeling -> Neural
2) After multiple combinations of the number of functions at the two layers of neural network, we observed that the following combination gave the best possible fit in terms of complexity, R-square and RMSE values:
   NTanH(1), NLinear(1), NGaussian(1), NTanH2(0), NLinear2(0), NGaussian2(0)
3) Neural network diagram and snapshot of hidden layers' parameter estimates:

Figure 47



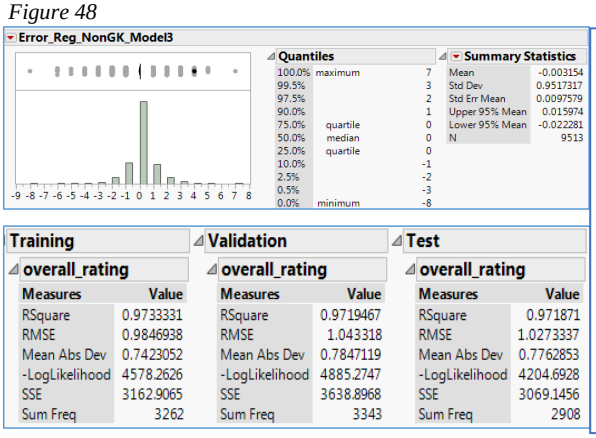
Estimates

Parameter	Estimate
H1_1:crossing	0.000132
H1_1:finishing	0.001408
H1_1:heading_accuracy	0.002327
H1_1:short_passing	0.002309
H1_1:volleys	0.000285
H1_1:dribbling	0.001702
H1_1:curve	0.000215
H1_1:free_kick_accuracy	0.000185
H1_1:long_passing	0.000226
H1_1:ball_control	0.006589
H1_1:acceleration	0.000904
H1_1:sprint_speed	0.001133
H1_1:agility	-0.00056
H1_1:reactions	0.005054
H1_1:balance	0.000378
H1_1:shot_power	0.001045
H1_1:jumping	0.000114
H1_1:stamina	0.000944
H1_1:strength	0.001841
H1_1:long_shots	9.556e-5
H1_1:aggression	0.000942
H1_1:interceptions	0.00249
H1_1:positioning	0.002801
H1_1:vision	0.00134
H1_1:penalties	0.000379
H1_1:marking	0.00329
H1_1:standing_tackle	0.005019
H1_1:sliding_tackle	0.001997

4) The rounded off predictions for this model are saved in the column "Reg_Pred_NonGK_Model3_Rounded" and the errors are stored in the column "Error_Reg_NonGK_Model3"
5) Error distribution and Summary of Fit:

The work presented here is our team's and our team's work alone.

Figure 48



Observations:

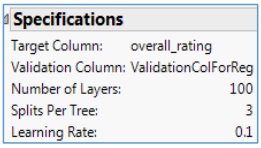
1) The R-square value on test data (Figure 48) is significantly higher compared to the linear regression and stepwise regression models (Figure 41 and 45 respectively)
2) There are no instances of overestimation or underestimation in the test data which implies that the predicted ratings are ????? within the acceptable range of the management
3) Since this is a black box model, we cannot get an idea about the significance of different attributes in predicting the variance of target column. The model complexity is also high

Formatted: Highlight

E. Reg_NonGK_Model4: Boosted Tree Regression

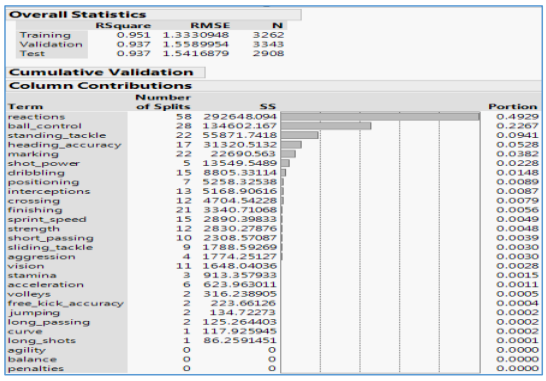
6) Select Analyze -> Modeling -> Partition -> Method -> Boosted Tree. After we varied the values of parameters like Number of Layers, Learning Rate among others and model complexity, we decided on the following values for the parameters:

Figure 49

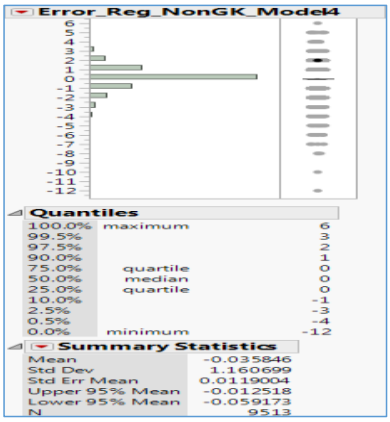


7) The rounded off predictions for this model are saved in the column "Reg_Pred_NonGK_Model4_Rounded" and the errors are stored in the column "Error_Reg_NonGK_Model4"
8) Evaluation Metrics, Column Contributions and Error Distributions:

Figure 50



The work presented here is our team's and our team's work alone.



Observations:

1) The R-square value on test data (Figure 50) is comparable to the one obtained during neural networks model (Figure 48), but is significantly better than the ones corresponding to Linear regression and Stepwise regression (Figure 41 and Figure 45 respectively)
2) The contribution of the attributes explaining the variance in target variable can be better ascertained than in neural networks
3) Number of instances of test data related to underestimation: 0
4) Number of instances of test data related to overestimation: 1
5) Total cost of underestimation: $0
6) Total cost of overestimation: $1.20 million

Model Performance Comparison on Test Data:

Name	R-Square	RMSE	Max. error	Mean error	Underestimation Instances	Cost of Underestimation ($)	Overestimation instances	Cost of Overestimation ($)	Complexity
Baseline			25	0.44	245	39500000	157	179.10 million	Minimum
Least squares	0.83	2.53	11	-0.01	5	384000	5	3.30 million	Simple
Stepwise	0.82	2.55	10	-0.001	6	384000	7	4.20 million	Most Simple
Neural Network	0.97	0.99	7	-0.003	0	0	0	0	Very Complex
Boosted tree	0.94	1.54	6	-0.036	0	0	1	1.20 million	Medium

Recommendations on Model Selection:

1) Considering all the evaluation metrics mentioned above, the boosted tree model can be recommended as the best model
2) However, as stated before, if the management is concerned only with overall rating and not with the significance of the attributes, then neural network can be selected
3) The least squares and stepwise regression models are comparatively simple in terms of model complexity. However, they have multiple instances of underestimating and overestimating the predicted values of overall_rating which might not be acceptable by the management

Analysis for Goalkeepers

Methodology

The approach is same as in the case of non-goalkeepers except for the fact that the costs associated with overestimation and underestimation of the ratings for the goalkeepers. They are mentioned as follows:

1) Underestimation:
    a. Occurs when overall_rating(actual) – overall_rating(predicted) > 8
    b. Per unit cost (decided by business): $50,000

c. Cost due to underestimating: $50,000 * (error – 8), where error = overall_rating(actual) – overall_rating(predicted)

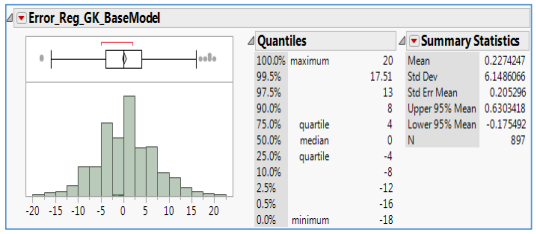
2) Overestimation:
   a. Occurs when overall_rating(predicted) – overall_rating(actual) > 8
   b. Per unit cost (decided by business): $165,000
   c. Cost due to overestimating: $165,000 * (error – 8), where error = overall_rating(predicted) – overall_rating(actual)

Algorithms Used

A. Reg_GK_BaseModel: Baseline model

a. The baseline predicted value is the average value of the target variable "overall_rating". The baseline predictions will help us ascertain the value added by the predictions generated by the subsequent models
b. The rounded off predictions are stored in the column "Reg_Pred_GK_BaseModel_Rounded", while the error values (overall_rating – Reg_Pred_GK_BaseModel_Rounded) are stored in "Error_Reg_GK_BaseModel"
c. The overall error distribution is as follows:

Figure 51



Observations:

- Number of instances related to underestimation: 24
- Number of instances related to overestimation: 21
- Total cost of underestimation: $3.60 million
- Total cost of overestimation: $11.06 million

B. Reg_GK_Model1: Standard least squares regression

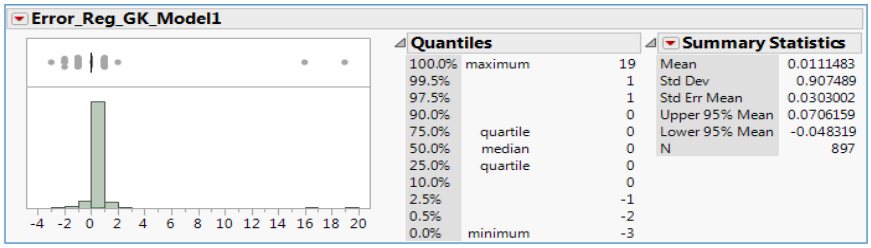
a. Select Analyze -> Fit Model. After selecting "overall_rating", a continuous variable as target, we selected Standard least squares in Personality. All the continuous predictors were considered for building the model
b. The rounded off predictions for this model are saved in the column "Reg_Pred_GK_Model1_Rounded" and the errors are stored in the column "Error_Reg_GK_Model1"
c. Equation to depict how the target variable is related to the predictors:

Figure 52

overall_rating = 1.41095814956425 + 0.240925700066856 * gk_diving + 0.227231428112422 * gk_handling + 0.0381701328779631 * gk_kicking + 0.220519182382447 * gk_positioning +0.246329055795156 * gk_reflexes

d. Overall error distribution and Summary of Fit:

Figure 53



Summary of Fit	
RSquare	0.959184
RSquare Adj	0.958508
Root Mean Square Error	1.31865
Mean of Response	66.76299
Observations (or Sum Wgts)	308

Crossvalidation			
Source	RSquare	RASE	Freq
Training Set	0.9592	1.3057	308
Validation Set	0.9904	0.6521	334
Test Set	0.9653	1.2016	255

e. Parameter Estimates:

Figure 54

Parameter Estimates				
Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	1.4109581	0.80976	1.74	0.0824
gk_diving	0.2409257	0.027464	8.77	<.0001*
gk_handling	0.2272314	0.021441	10.60	<.0001*
gk_kicking	0.0381701	0.015486	2.46	0.0143*
gk_positioning	0.2205192	0.020666	10.67	<.0001*
gk_reflexes	0.2463291	0.024434	10.08	<.0001*

Observations:
1) Significance level (α) considered: 0.02
2) The attributes with Prob>|t| greater than the considered significance level are least significant in interpreting the variance of target variable
3) Observing the error distribution statistics in **Figure 53**, we can infer that this model is less prone towards making overestimation errors compared to the baseline model (**Figure 51**). However, whether to accept or reject this model can be decided after all the models have been analyzed
4) Number of instances of test data related to underestimation: 1
5)Number of instances of test data related to overestimation: 0
6)Total cost of underestimation: $400,000
7)Total cost of overestimation: 0

C. Reg_GK_Model2: Stepwise regression

a. Select Analyze -> Fit Model. After selecting "overall_rating", a continuous variable as target, we selected Stepwise Regression in Personality. The predictors considered are visible in the equation in Figure 56.
b. The rounded off predictions for this model are saved in the column "Reg_Pred_GK_Model2_Rounded" and the errors are stored in the column "Error_Reg_GK_Model2"
c. Selecting Predictors for the model:

Figure 55

SSE	DFE	RMSE	RSquare	RSquare Adj	Cp	p	AICc	BIC
535.69329	303	1.3296483	0.9584	0.9578	10.075565	5	1056.812	1078.913

Current Estimates

Lock	Entered	Parameter	Estimate	nDF	SS	"F Ratio"	"Prob>F"
☑	☑	Intercept	1.68907603	1	0	0.000	1
☐	☑	gk_diving	0.25540965	1	157.6036	89.144	1e-18
☐	☑	gk_handling	0.23722232	1	220.7385	124.855	1.7e-24
☐	☐	gk_kicking	0	1	10.56442	6.076	0.01426
☐	☑	gk_positioning	0.23004542	1	223.2613	126.282	1e-24
☐	☑	gk_reflexes	0.24443078	1	174.1902	98.526	2.7e-20

Approach:

1. Using the "Step" feature, we keep selecting the predictors depending on their p-value
2. The process is continued till the p-value of all the selected attributes is less than the considered significance level of 0.02
3. Since including "gk_kicking" does not improve the R-square value by a great margin and it has the highest p-value, we have not considered this attribute for the stepwise model

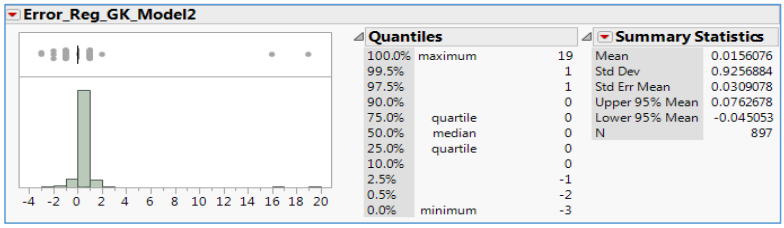
d. Equation to depict how the target variable is related to the predictors:

Figure 56

overall_rating = 1.68907602716538 + 0.255409645517566 * gk_diving + 0.237222319447208 * gk_handling + 0.230045423481804 * gk_positioning + 0.244430778210101 * gk_reflexes

e. Overall error distribution and Summary of Fit:

Figure 57



Quantiles

100.0%	maximum	19
99.5%		1
97.5%		1
90.0%		0
75.0%	quartile	0
50.0%	median	0
25.0%	quartile	0
10.0%		0
2.5%		-1
0.5%		-2
0.0%	minimum	-3

Summary Statistics

Mean	0.0156076
Std Dev	0.9256884
Std Err Mean	0.0309078
Upper 95% Mean	0.0762678
Lower 95% Mean	-0.045053
N	897

Summary of Fit

RSquare	0.958363
RSquare Adj	0.957813
Root Mean Square Error	1.329648
Mean of Response	66.76299
Observations (or Sum Wgts)	308

Crossvalidation

Source	RSquare	RASE	Freq
Training Set	0.9584	1.3188	308
Validation Set	0.9898	0.6729	334
Test Set	0.9631	1.2405	255

The work presented here is our team's and our team's work alone.

f. Parameter Estimates:

Figure 58

Parameter Estimates

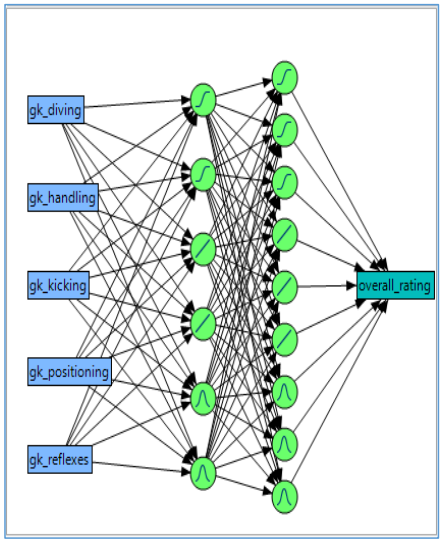
Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	1.689076	0.808549	2.09	0.0375*
gk_diving	0.2554096	0.027051	9.44	<.0001*
gk_handling	0.2372223	0.02123	11.17	<.0001*
gk_positioning	0.2300454	0.020471	11.24	<.0001*
gk_reflexes	0.2444308	0.024625	9.93	<.0001*

Observations:
1) Number of instances of test data related to underestimation: 1
2)Number of instances of test data related to overestimation: 0
3)Total cost of underestimation: $400,000
4)Total cost of overestimation: 0
5) Even after excluding "gk_kicking", we observe that the performance of this model is same as that of Reg_GK_Model1 in terms of costs and the R-square values are also comparable.

D. Reg_GK_Model3: Neural network

a. Select Analyze -> Modeling -> Neural
b. After multiple combinations of the number of functions at the two layers of neural network, we observed that the following combination gave the best possible fit in terms of R-square and RMSE values:
   NTanH(3), NLinear(3), NGaussian(3), NTanH2(2), NLinear2(2), NGaussian2(2)
c. Neural network diagram and snapshot of hidden layers' parameter estimates:

Figure 59

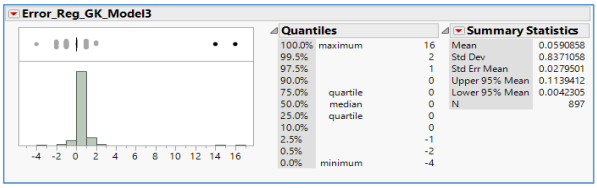


Estimates

Parameter	Estimate
H2_1:gk_diving	-0.10065
H2_1:gk_handling	-0.10923
H2_1:gk_kicking	-0.04808
H2_1:gk_positioning	-0.05178
H2_1:gk_reflexes	0.024992
H2_1:Intercept	18.7067
H2_2:gk_diving	0.017506
H2_2:gk_handling	0.077303
H2_2:gk_kicking	0.099083
H2_2:gk_positioning	-0.11667
H2_2:gk_reflexes	0.017376
H2_2:Intercept	-6.41865
H2_3:gk_diving	-0.06239
H2_3:gk_handling	-0.01312
H2_3:gk_kicking	-0.16426
H2_3:gk_positioning	0.007482
H2_3:gk_reflexes	-0.09363
H2_3:Intercept	21.67316
H2_4:gk_diving	0.035079
H2_4:gk_handling	0.049546
H2_4:gk_kicking	0.007574
H2_4:gk_positioning	0.053149
H2_4:gk_reflexes	0.077707
H2_4:Intercept	-15.833
H2_5:gk_diving	-0.08856
H2_5:gk_handling	0.130937
H2_5:gk_kicking	-0.03002
H2_5:gk_positioning	0.014697
H2_5:gk_reflexes	-0.00568
H2_5:Intercept	-0.92239
H2_6:gk_diving	-0.09493
H2_6:gk_handling	-0.04934
H2_6:gk_kicking	-0.0215
H2_6:gk_positioning	-0.05252
H2_6:gk_reflexes	0.105889
H2_6:Intercept	6.86947

d. The predictions for this model are saved in the column "Reg_Pred_GK_Model3" and the absolute errors are stored in the column "Error_Reg_GK_Model3"
e. Error distribution and Summary of Fit:

Figure 60



Training		Validation		Test	
overall_rating		overall_rating		overall_rating	
Measures	Value	Measures	Value	Measures	Value
RSquare	0.9702282	RSquare	0.9853614	RSquare	0.9627981
RMSE	1.1151768	RMSE	0.8071661	RMSE	1.2449719
Mean Abs Dev	0.6198736	Mean Abs Dev	0.6003635	Mean Abs Dev	0.6145718
-LogLikelihood	470.60906	-LogLikelihood	402.37404	-LogLikelihood	417.70313
SSE	383.03474	SSE	217.6067	SSE	395.23852
Sum Freq	308	Sum Freq	334	Sum Freq	255

Observations:

a. Number of instances of test data related to underestimation: 2
b. Number of instances of test data related to overestimation: 0
c. Total cost of underestimation: $700,000
d. Total cost of overestimation: 0
e. The R-square value on test data (**Figure 60)** is significantly the same when compared to the previous models is significantly higher compared to the previous linear regression and stepwise regression models
f. Since this is a black box model, we cannot get an idea about the significance of the different attributes in predicting the variance of target column

E. Reg_GK_Model4: Boosted tree regression

a. Select Analyze -> Modeling -> Partition -> Method -> Boosted tree. Although we varied the different parameters like Number of layers, learning rate among others and model complexity, we decided on following values for the parameters:

Figure 61

Specifications			
Target Column:	overall_rating	Number of training rows:	308
Validation Column:	ValidationColForGK	Number of validation rows:	334
Number of Layers:	50	Number of test rows:	255
Splits Per Tree:	3		
Learning Rate:	0.1		

b. Evaluation Metrics, Column Contributions, and Error Distributions:

Figure 62

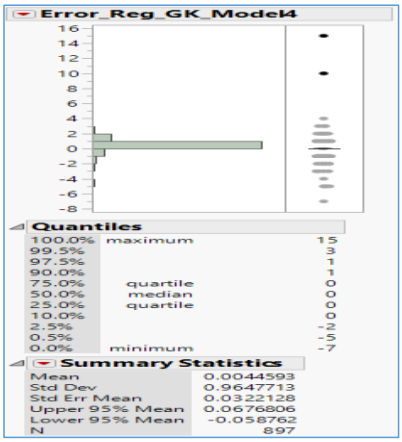
Overall Statistics

	RSquare	RMSE	N
Training	0.977	0.9866952	308
Validation	0.968	1.1907472	334
Test	0.953	1.3997209	255

Cumulative Validation

Column Contributions

Term	Number of Splits	SS	Portion
gk_diving	29	23833.9553	0.3604
gk_positioning	40	16420.4443	0.2483
gk_handling	31	13385.0297	0.2024
gk_reflexes	31	11959.1887	0.1808
gk_kicking	19	537.378114	0.0081



Observations:

- The R-square value on test data (Figure 62) is highest when compared to all the other models. Also, the R-square value obtained in case of Boosted Tree is almost equivalent to the value achieved via neural Network Model
- The contribution of the attributes in explaining the variance in target variable can be better ascertained than in case of neural networks
- Number of instances of test data related to underestimation: 2
- Number of instances of test data related to overestimation: 0
- Total cost of underestimation: $450,000
- Total cost of overestimation:

The work presented here is our team's and our team's work alone.

Model Performance Comparison on Test Data:

Name	R-square	RMSE	Max. error	Mean error	Underestimation Instances	Cost of Underestimation ($)	Overestimation instances	Cost of Overestimation ($)	Complexity
Baseline			20	0.22	24	3.60 million	21	11.06 million	Minimum
Least squares	0.96	1.32	19	0.01	1	400000	0	0	Simple
Stepwise	0.96	1.33	19	0.01	1	400000	0	0	Most Simple
Neural Network	0.97	1.24	16	0.05	2	700000	0	0	Very Complex
Boosted tree	0.98	1.40	15	0	2	450000	0	0	Medium

Recommendations on Model Selection

a. Considering all the evaluation metrics mentioned above, the stepwise regression model can be recommended as the best model
b. However, as stated before, if the management is concerned only with the accuracy of prediction and not with the cost and complexity of the model then Boosted tree can be selected

The work presented here is our team's and our team's work alone.

CONCLUSION AND FUTURE SCOPE

The report described an in-depth analysis conducted on the data describing the playing skills of European soccer footballers. The objective of this exercise was to empower the managements of the competing teams in making better and well-informed decisions on which players are to be selected for the upcoming season. For ease of understanding, we have separated the classification and regression modeling datasets for both GKs and NonGKs.

There are multiple ways in which the team managements can gain by considering the predicted outcomes of the recommended models:

- The relations between variables depicted by exploratory data analysis can aid the management in finding previously unknown patterns in the data. This can also serve as a reference to the coaching faculty in understanding the specific areas of improvement for any footballer
- Using the stepwise linear regression model recommended for GKs, we can obtain a mathematical model relating the overall rating of a goalkeeper with his various skills. Also, the significance of the predictors can also be clearly understood
- In real world scenario, the actual values of the target column (overall_rating) might not be available. The Boosted Tree algorithm might be slightly complex as compared to linear regression; but in case NonGKs, the model evaluation matrix clearly highlights a decrease of approximately 99.33% in overall cost of misclassification if the management considers the predicted overall ratings for footballers instead of applying the naïve approach. This is of course, assuming the fact only the attributes present in the dataset affect a footballer's overall rating.
- The derived target variable for classification problem, "Potential_Classification" depicts whether a player has high or low potential of performing well in the upcoming season. Not only does the classification algorithms like Logistic Regression or Decision Trees achieve a much higher overall accuracy in predicting the target column as compared to the baseline models, but also the number of instances of misclassifying the actual values is significantly lower. These facts can significantly bolster the confidence of the management in making lucrative selections.

Keeping this in perspective, following are few steps which can be taken up in future by the analytics team:

- Instead of considering data from only the last played season, may be accumulate data for multiple seasons and observe the trends in ratings of footballers
- With adequate information, perform analysis on the footballers at different playing positions like striker, mid-field or defense
- The current dataset has comparatively fewer observations for goalkeepers. The analysis may be more effective if a greater quantity of data is available

ok- good suggestions

However, a model is only as good as its underlying assumptions and the data on which it is built. Furthermore, the effectiveness of the models considered is dependent on assumptions like the available data was fairly accurate, the considered threshold values are acceptable by the business and the predictors mentioned in the data are the only ones affecting the target variables.
However, some players just don't work well with this formula, meaning they end up getting rated much lower than their real-world performance would indicate. For example, a player is not a great dribbler, he can't strike the ball properly, his finishing is sometimes off and his shot power is not his strength as well. But the same player always finds the right spot on the pitch and scores a goal every-time. So, if you rate the player with this formula, he ends up with a low rating which doesn't make a lot of sense. This indicates that there is always a possible uncertainty associated with the predictions. Hence, we believe that the entire process of data analysis is a cyclic process which can be constantly improved through an optimal blend of business knowledge and statistics.

The work presented here is our team's and our team's work alone.

APPENDIX A – GLOSSARY

1. **Overfitting -** is a situation when the model incorporates patterns present in the sample data that are not present in the overall population. This means that the model can produce acceptable results for a given data set, but when the data set is changed results may not be reliable.

2. **Area Under the Curve (AUC – ROC) -** The ROC curve is the plot between sensitivity and (1-specificity). (1- specificity) is also known as false positive rate and sensitivity is also known as True Positive rate.

3. **Root Mean Squared Error (RMSE)** – The formula is given as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

4. **Split History-** is a two-dimensional graph plotted between number of splits and RSquare values obtained for each split of the decision tree model.

5. **Over-fitting-** is a situation when the model incorporates patterns present in the sample data that are not present in the overall population. This means that the model can produce acceptable results for a given data set but when the data set is changed, results may not be reliable.

6. **RSquare –** is the coefficient of determination denoted as R2 or r2 ???????. It is a number that indicates the proportion of the variance in the dependent variable that is predictable from the independent variable.

7. **Misclassification -** Classifying low potential player as a high potential player (Type I–False Positive). Also, Classifying high potential player as a low potential player (Type II – False Negative).

8. **Goodness of Fit -** describes measures used to test how well a model can fit the data set to produce accurate results. This is generally measured by looking at the difference between actual and predicted values of the model. If there is a small fit, then the model is a good fit.

9. **Boosting** – It is a machine learning ensemble meta-algorithm for primarily reducing bias, and variance in supervised learning, and a family of machine learning algorithms which convert weak learners to strong ones.

10. **True Positive Rate -** aka. sensitivity, hit rate, and recall, which is defined as TP/TP+FN. Intuitively this metric corresponds to the proportion of positive data points that are correctly considered as positive, with respect to all positive data points. In other words, the higher TPR, the fewer positive data points we will miss.

11. **False Positive Rate -** aka. fall-out, which is defined as FP/FP+TN. Intuitively this metric corresponds to the proportion of negative data points that are mistakenly considered as positive, with respect to all negative data points. In other words, the higher FPR, the more negative data points we will misclassified.

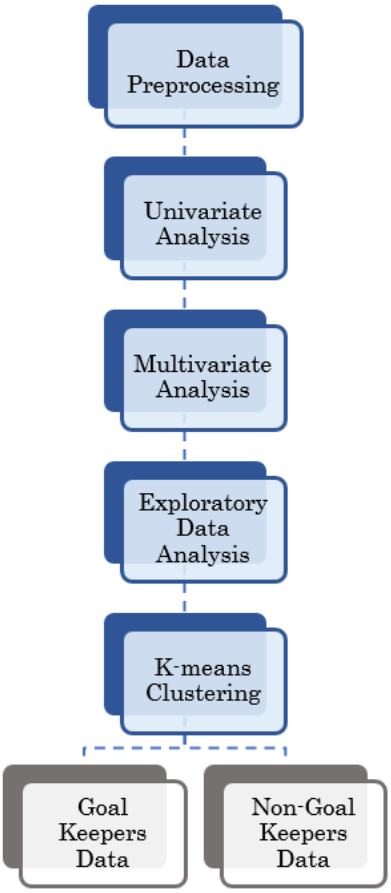
The work presented here is our team's and our team's work alone.

12. **Interquartile Range** - is a measure of variability, based on dividing a data set into quartiles. Quartiles divide a rank-ordered data set into four equal parts and are denoted by Q1, Q2, and Q3 respectively. The interquartile range is equal to Q3 minus Q1.

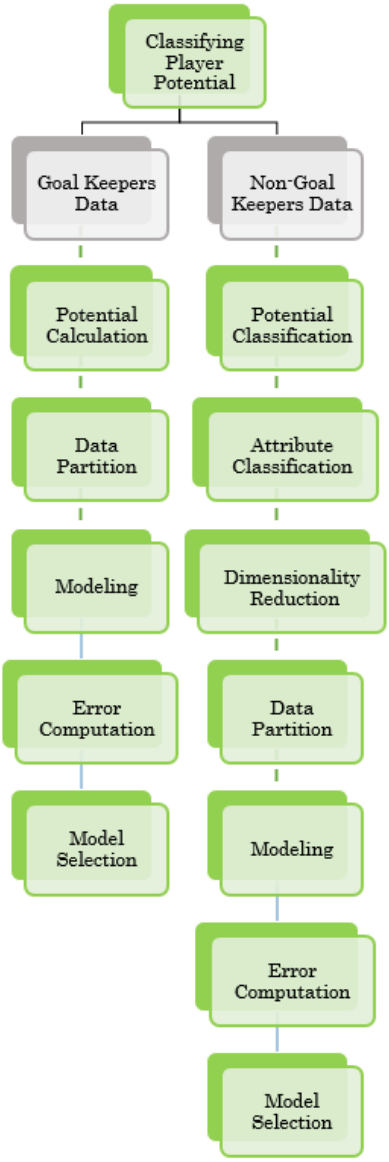
The work presented here is our team's and our team's work alone.

APPENDIX B – PROCESS OVERVIEW CHARTS

PRE-PROCESS OVERVIEW

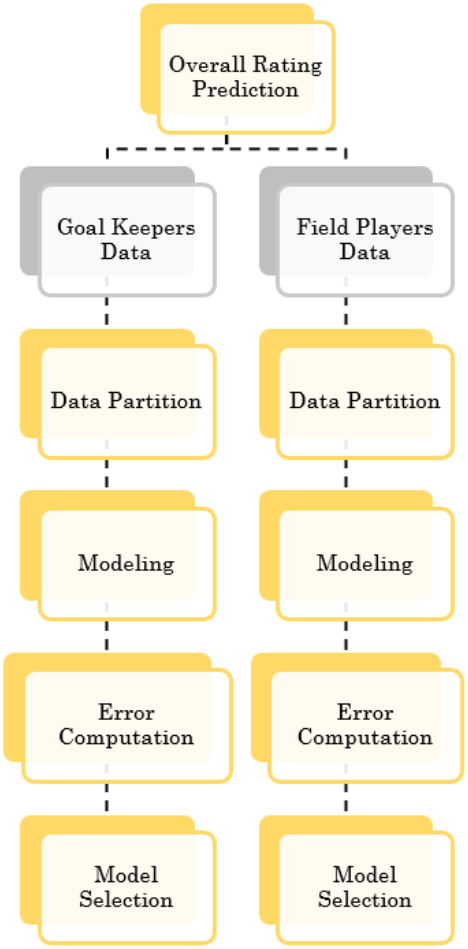


POTENTIAL CLASSIFICATION PROCESS OVERVIEW



The work presented here is our team's and our team's work alone.

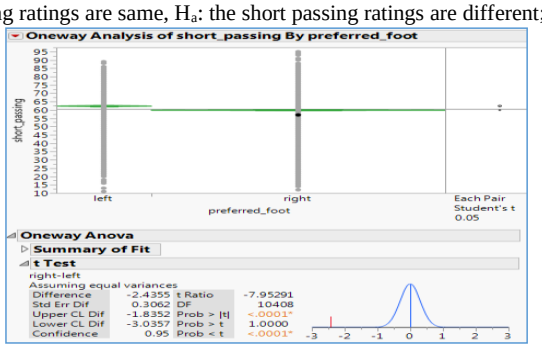
RATING PREDICTION PROCESS OVERVIEW



The work presented here is our team's and our team's work alone.

APPENDIX C – ADDITIONAL INSIGHTS

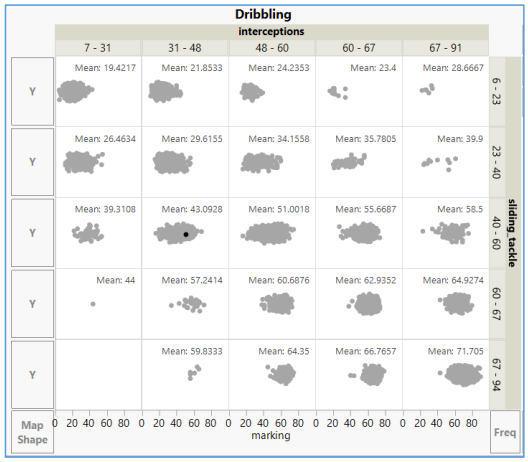
a. H_o: the short passing ratings are same, H_a: the short passing ratings are different;



b. Observation: In this paired-t test, since the Prob > |t| (< 0.0001) is less than the considered significance level, we can reject the claim.

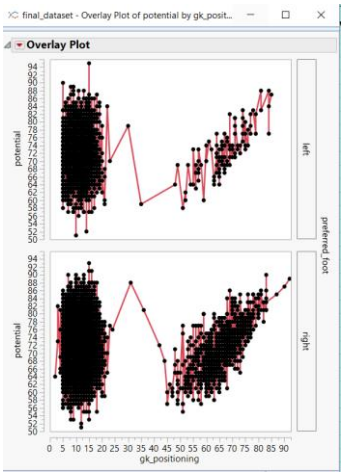
c. The number of preferred left and right foot players have their reflexes quotient between the range shown below with respect to overall potential. The histograms on the plot confirms that the distribution for potential is nearly uniform.

d. We can infer from the below graph that dribbling variables like Interceptions, Sliding_Tackle and Marking are related to each other (i.e) if a person is skillful in sliding tackle, he is likely to perform well in marking as well as interceptions. Also, we can notice that the bin for left extreme column is empty i.e. there are no players binned in 67-94 for sliding tackle,7-31 for interceptions and marking.



Overlay Plot of Potential by gk_positioning

The work presented here is our team's and our team's work alone.



The number of players with their gk_positioning skills in the range of 20-45 is very minimal when compared to other ranges.

The work presented here is our team's and our team's work alone.

APPENDIX D – DATA DICTIONARY

Data Dictionary.jmp

Multivariate_Analysis.xltx

The work presented here is our team's and our team's work alone.

REFERENCES

European Soccer Database. Retrieved October 28, 2016, from Kaggle https://www.kaggle.com/hugomathien/soccer

Analytics Vidhya. Learn Everything about Analytics Blog. Retrieved Oct 2016, from https://www.analyticsvidhya.com/

Players and team's attributes from EA Sports FIFA games. Retrieved Nov 2016, from http://sofifa.com/ :

Most expensive football transfers. Retrieved Nov 2016, from https://en.wikipedia.org/wiki/List_of_most_expensive_association_football_transfers

Salaries of players. Retrieved Nov 2016, from www.totalsportek.com.

Best Paid Goalkeepers. Retrieved Nov 2016, from http://www.telegraph.co.uk/sport/football/11696669/The-worlds-top-10-best-paid-goalkeepers-including-Petr-Cech.html?frame=3352403

Official FIFA Ratings. Retrieved Nov 2016 from https://www.easports.com/fifa/ratings

The work presented here is our team's and our team's work alone.