**SYST 568: Applied Predictive Analytics**

**Final Report**

**FIFA 2019 Dataset: Predicting Player Wage**

**Team** **members:**

Ritika De

Jiyad ur Rehman

Alexei Sothoron

Zulay Melendez Valle

Ruichen Zhou

**Abstract**

Soccer players' wage is determined by a combination of different attributes, such as players' skills, previous matches performance, and reputation throughout their career. These characteristics allow managers of soccer clubs to determine the appropriate salary for players. Our project aims to predict soccer players' wage based on their attributes using statistical modeling techniques. Furthermore, identify which attributes are relevant to predict wage.

Firstly, the data is gathered, organized and explored, with the intent to identify the predictors and target variables. Secondly, the data is explored to find out more about the relationships among the variables (for example, their correlation coefficients), how the variables behave independently, and their respectively distributions (i.e. are the predictor skewed? Are there outliers? Is the data noisy?). Lastly, several statistical techniques are utilized to select the best predictors and different models (i.e. Linear Regression, Random Forest, Support Vector Machine), and their performance will be examined.

**Background**

For this class project FIFA 2019 complete player dataset was used that is available on Kaggle’s website. The link to the dataset is as follows: <https://www.kaggle.com/karangadiya/fifa19>. The downloaded csv file contains detailed attributed for every player that is registered in the latest edition of FIFA 2019 database, which was scrapped from <https://sofifa.com>. As this dataset has approximately 89 variables, including but not limited to skills rankings, wage, value, overall potential, etc., it can be used to analyze, predict, or solve a number of problems. However, this project attempts to predict wages of players, using several prediction modeling techniques.

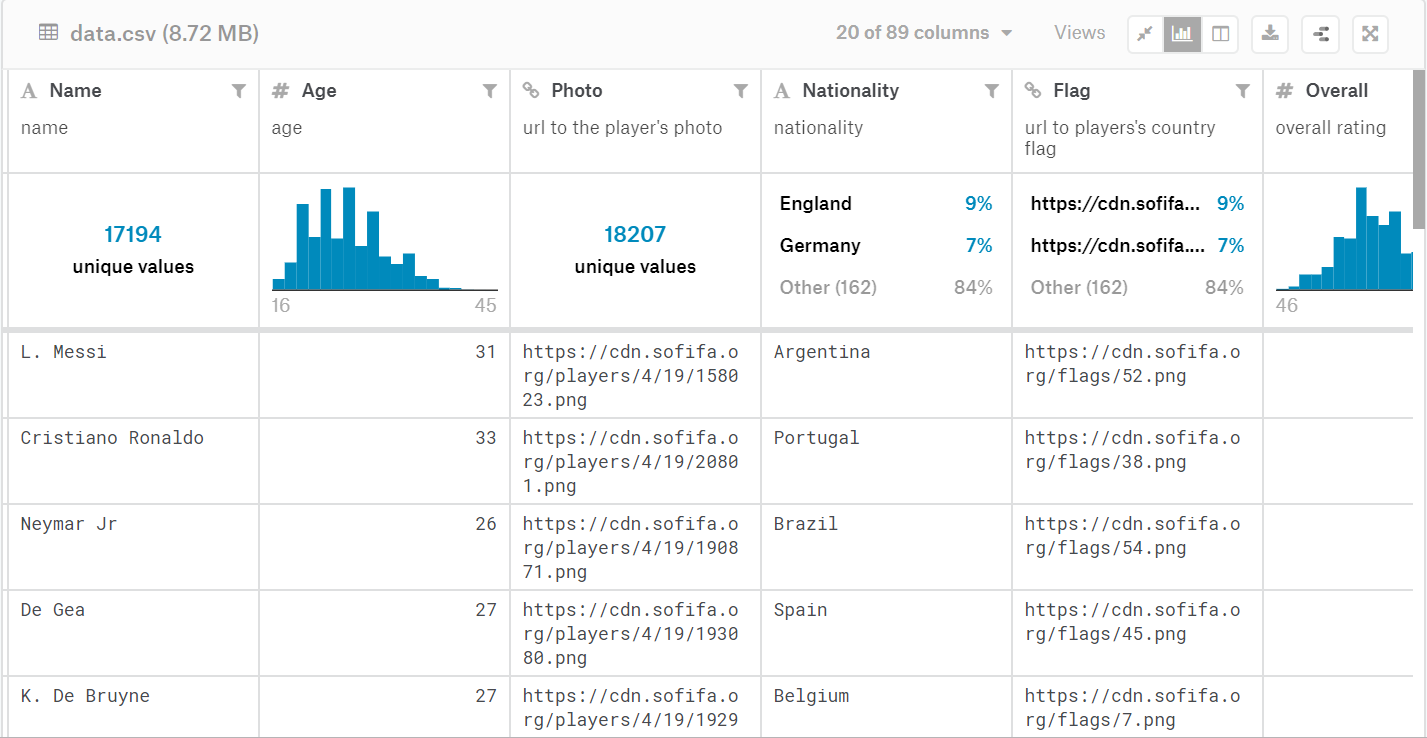
As in any sports franchise, the general manager (GM) seeks to predict the right price to offer a player given the player’s unique characteristics. A poor prediction would lead to a player being underpaid or overpaid; either way delivering negative results for the team in the long run. Optimizing wage for each prospective player gives the GM a superior team because they would not misallocate cash to underperformers, allowing them to spend more on star-players.

**Objective**

The object of this project is to identify a machine-learning model that can predict wage of a soccer player, precisely and effectively. The focus is towards the players, only, excluding the goalkeepers, and to recognize the features that should be considered by clubs when they look to hire new players for their upcoming season. Furthermore, using the prediction model, the clubs can get the expected wage they would have to pay for their new players based on their skill set.

**Dataset**

As mentioned previously, the data is downloaded from Kaggle, and is titled ‘FIFA 19 complete player dataset’. The comma separated variable (csv) file contains 89 columns, and 18207 rows. Figure 1 shows a glance of the dataset*.*

**

*figure 1. Dataset Preview.*

Following is the description of some attributes presented in the dataset.

* **ID:** Player ID (unique)
* **Name:** Player name
* **Age:** Player age
* **Photo:** Player current picture
* **Nationality:** Player nationality (where he come from)
* **Flag:** Player's country flag
* **Overall:** Overall rating. Determine the quality and the feature of a player's technical skills, behaviours and performance on the pitch. Player Attributes are rated from 0 to 99. [1]
* **Potential:** potential rating
* **Club:** Player current club
* **Value:** Player current market value
* **Wage:** Player current wage
* **Preferred Foot**: left/right
* **International Reputation:** rating on a scale of 5

1Market value is based on a combination of factors such as squad status, age, talent, league factor, premium position, adaptability, depreciation and so on [2].

2 Wage “takes into account a number of factors including how good the player is (based on the total number of attributes points a player has), the age of the player, the amount paid to buy the player and how long the player has been at their current club as well as a small random modifier” [2]

* **Weak Foot:** rating on a scale of 5. A player's foot (left or right) that is weaker than their preferred foot. A player's attribute rated between 1 to 5 which specifies the shot power and ball control for the other foot of that player than his preferred foot [2].
* **Skill Moves**: rating on a scale of 5
* **Work Rate:** PlayerWork Rate is the rate of a player's behavior on the pitch in terms of attacking and defensive works. Attack work rate/defense work rate

For our analysis we decided to keep ‘Wage’ as our response variable and all other were supposed as predictors. The following part shares which predictors we kept and how we cleaned them to make them useful for our analysis and model.

**Data Cleaning & Preparation**

The raw dataset contained many columns that would have not been useful to our analysis and prediction model. For instance, the original file has row number, ID (i.e. unique id for every player), Photo, Flag, Club Logo (url to the club logo), etc., to name a few, were dropped.

Next, it was important to identify the appropriate response variable, as there were two options, namely, Wage and Value. For this project, Wage was selected. This column had zero values too, which could be because of incorrect data entry or were missing values, so it was decided to delete these rows. It was also noted that the alphabet ‘k’ was attached to the values that signified thousand, thus making the column non-numerical. It is because of this, that k’s were removed from Wage column. Similarly, ‘lbs’ was removed from Weight column.

Further, the remaining columns that were treated as predictors, were examined as a part of the data cleaning process. It was identified that there were a few rows that had null values in approximately 70% of the predictors. It made it impossible to use these records for analysis, and it was difficult to impute the values. This is the reason why it was best to delete these records. Another column that required cleaning was ‘Height’. It had values such as 5’11”. The approach to clean it was to convert the height to centimeters. This would make the values consistent and calculations easier to interpret.

Some of numeric predictors contained special characters, for instance 92+1. After exploring the discussion threads on Kaggle, it was discovered that ‘+1’signified the improvement of a player’s skills during the season. It was then decided to clean such columns by only keeping the number before the plus sign and deleting anything after the sign.

In addition to the existing predictors, two more columns were generated. Using the ANOVA test, it was determined that the ‘Nationality’ is an important variable, however it had more than 53 levels which would potentially be an issue when running models such as random forest, as the model can only take categorical variables with less than 53 levels. Thus, the newly generated column, named ‘Continent’, was based on the nationality column, using the “countrycode” library in R. Duration was another generated column that was the duration of the contract. This column was generated using Joined date and contract valid until. The null values in the duration column were filled with median values.

As stated in the project’s objective section, the team’s focus was the analysis of soccer players’ wage, and not goalkeepers. It is for this reason that the team decided to remove columns associated with goalkeepers. It was noted that a single prediction model would not be able to effectively predict wages of players and goalkeepers, as they require separate set of attributed.

After cleaning the noise, removing or imputing missing values, and keeping on the relevant features, the data set, resultantly, has 15,926 rows and 72 columns, including the target variable, Wage.

**Exploratory Data Analysis**

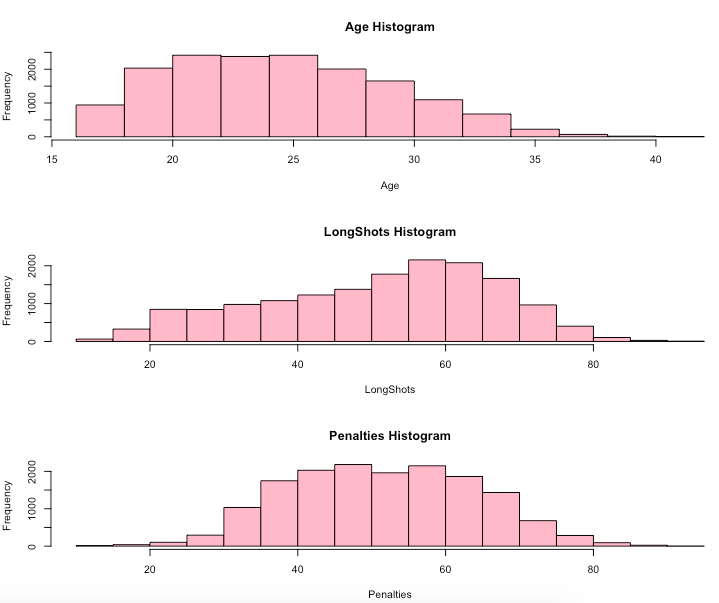
As the objective of the project is to predict wages for players based on their skill set, our first instinct was to analyze the independent variables’ relationship with the dependent variable (Wage). For this, numerical and categorical variables were separated and their relationships with Wage were analyzed.

First, the team explored the correlation coefficient with Wage. The team also plotted the pairwise correlation plot to make it easier to view the correlations, as seen in figure 2. As it can be noted from the correlation plot, a multicollinearity problem existed, that is, some of the independent variables were highly correlated with one another. Keeping these highly correlated independent variables in would potentially affect the interpretation of models’ results, making it difficult to understand. Thus, with the help of findCorrelation, which is a function in R that returns a vector of the data frame’s column numbers that should be removed in order to reduce the pairwise correlation **[1]**, a number of numeric independent variables were dropped from the data frame and would not be considered for further in the analysis of Wage. Some of these variables were, Overall, BallControl, LCM, CM etc.

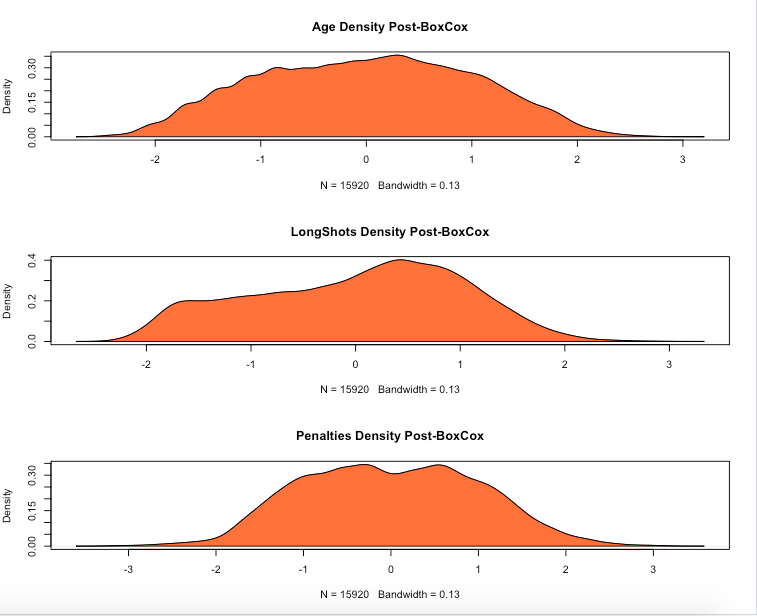
Categorical variables were separately analyzed using the aov test (it is an analysis-of-variance test). The hypothesis tests the equality of two or more group means. For this test, a high p-value suggests that the null hypothesis cannot be rejected. With these results, the team concluded to delete a few columns from the data frame, namely, ‘Preferred foot’ and ‘club category’.

After deleting some numeric and categorical predictors, the resultant dataset had 35 variables (including Wage), remaining. These variables were further analyzed for outliers, normality, and skewness.

As it can be seen in figure 2, a few of the numeric variables were skewed. With preprocess function in R, the variables were centered, scaled, and transformed using Box-cox for a more normal distribution. The transformed variables are shown in figure3.



*figure 2. Few of the skewed variables*

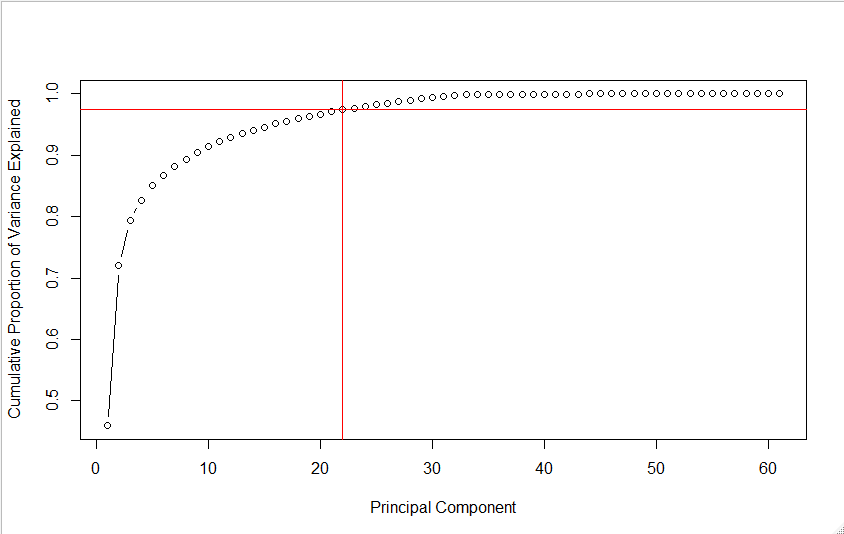


*figure 3. Post Centering, Scaling and Box-Cox Transformation*

` Wage, i.e., the chosen target variable, also was right skewed; hence it was decided to normalize it and apply Box-Cox transformation to it. This change was necessary for the linear regression model, however, for all other models, wage was not required to be transformed as they are non-linear prediction models.

**Principal Component Analysis (PCA)**

For the principal component analysis (PCA), the numerical variables were separated from categorical variables as PCA can only be applied to numerical variables. Later, PCA was applied using prcomp() function in R. With PCA applied to the variables, a chart was plotted to see how much variance is explained by different principal components. The plot is shows in figure 4:



*figure 4. Cumulative Proportion of Variance Explained vs. Principle Components.*

With the plot shown in figure 4, one can see that 22 principal components are enough to explain 97.5% of variance in the dataset; thus instead of 60 numerical variables, only 22 principal components. After the PCA transformation, the categorical variables were added back to the data frame. So, now, the data set contained 29, which included wage (the dependent variable), 22 principal components, and the rest were categorical variables.

Finally, the data was centered, scaled, and transformed using box-cox and PCA. Two datasets were derived to observe if the PCA datasets helped better predict wage, versus non-pca dataset (which was only transformed using Box-Cox). As mentioned earlier, PCA dataset has 29 variables, and non-pca dataset has 35 variables (inclusive of Wage in both).

**Data Models**

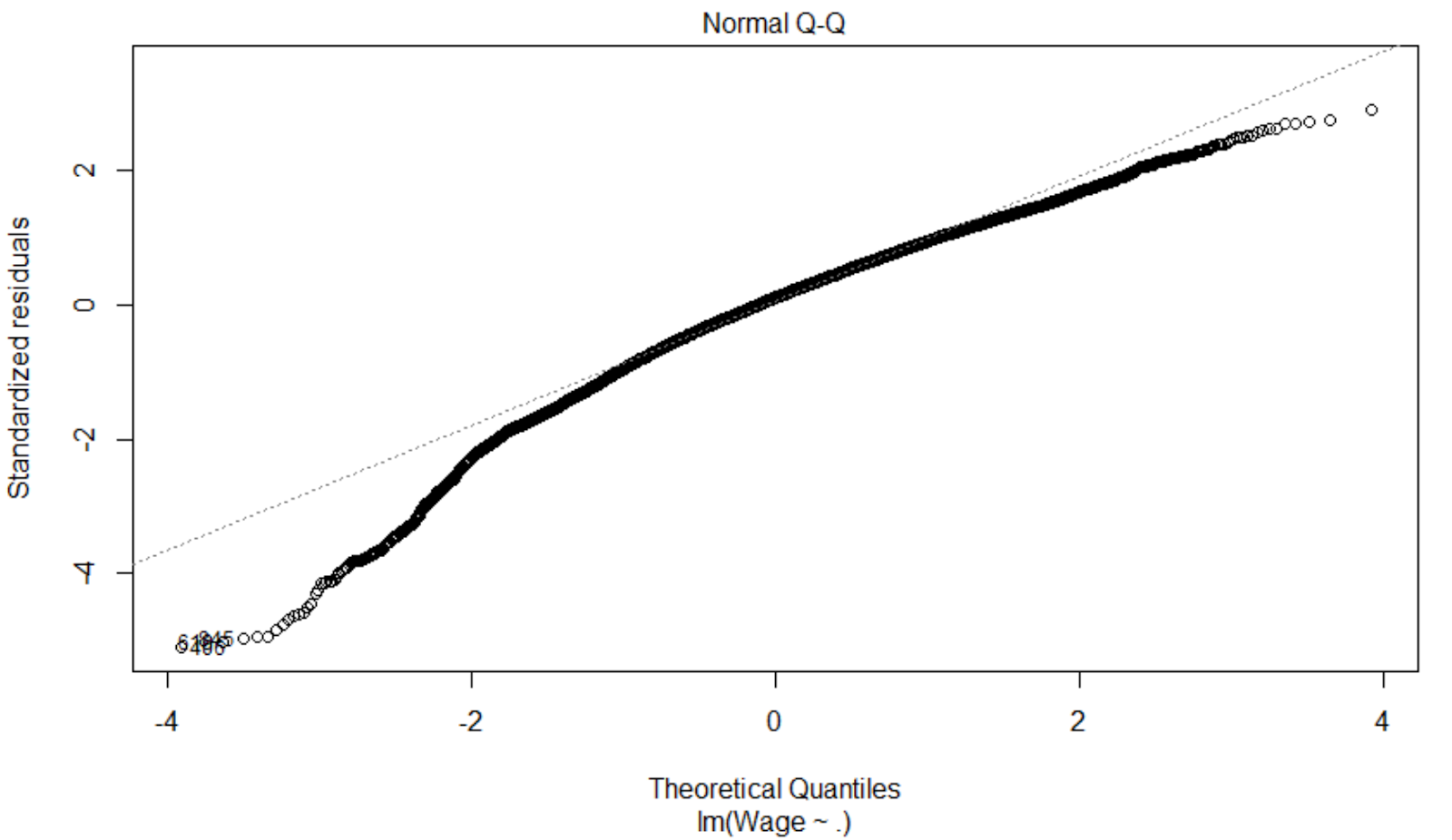
After cleaning, visualizing, processing, and transforming the dataset, some algorithms were identified as potential methods to reach the objective of this project: Predict soccer players appropriate wage. The following section discusses each of the models built, using PCA and non-PCA dataset, are discussed in detail, and their results are analyzed.

**1) Linear Regression Models**

For linear regression, both versions of the dataset, i.e. with and without PCA, are used to find which one would perform better. As the most straightforward model in our model choices, we expect the result to be the baseline while comparing with all other models. While exploring the correlation coefficients of Wage with all other predictors, majority were in the range between 0.2 to 0.5, thus indicating that the correlation, even though positive, was not too strong; leading us to believe that the linear regression model to not perform very well on this data.

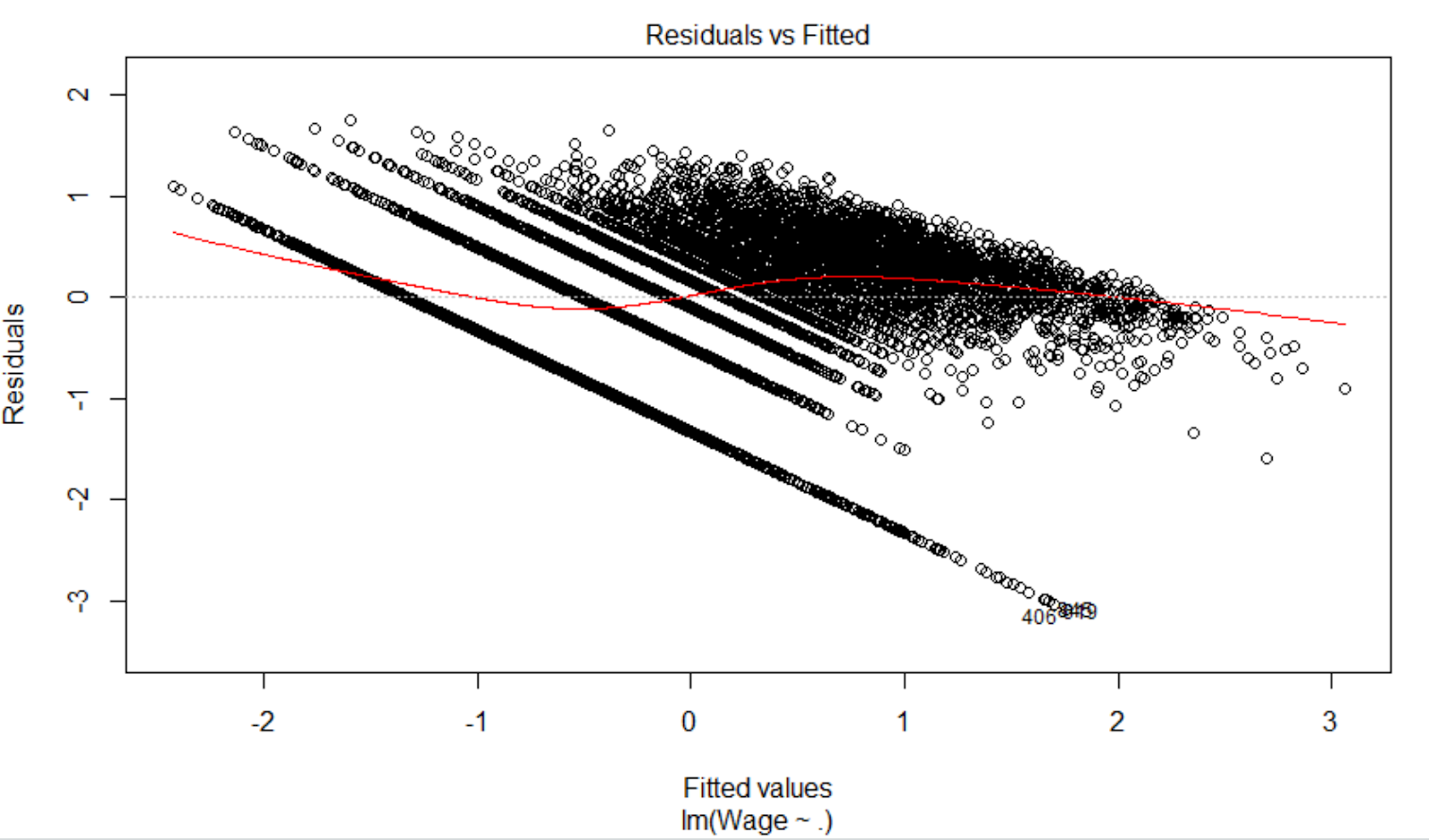
**a) Linear Regression with PCA**

We used lm function in R to perform the linear regression on the data with PCA. The Wage column was also transformed using Box-Cox, in addition to being centered and scaled. The results are shown in figures below:

******

*figure 5. Q-Q Plot for linear regression with PCA Data*

In the generated Q-Q plot, we can easily tell that the residuals have nonlinearity which means linear regression cannot explain the relationship between players’ wage and all the predictors (after PCA).

******

*figure 6. Residual-fit Plot for linear regression with PCA Data*

The residual-fit plot shows curved pattern, which violates the independence assumption. Also, the residual variance is higher for smaller values and decreases as predicted wage increases. Thus the constant variance assumption is violated. Results of both plots indicate that we should not stop on the regression model. The table below shows the RMSE of the model:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Train RMSE** | **Test RMSE** | **Normalized RMSE** |
| Linear Regression with PCA | 0.603967 | 0.6066876 | 0.1656875 |

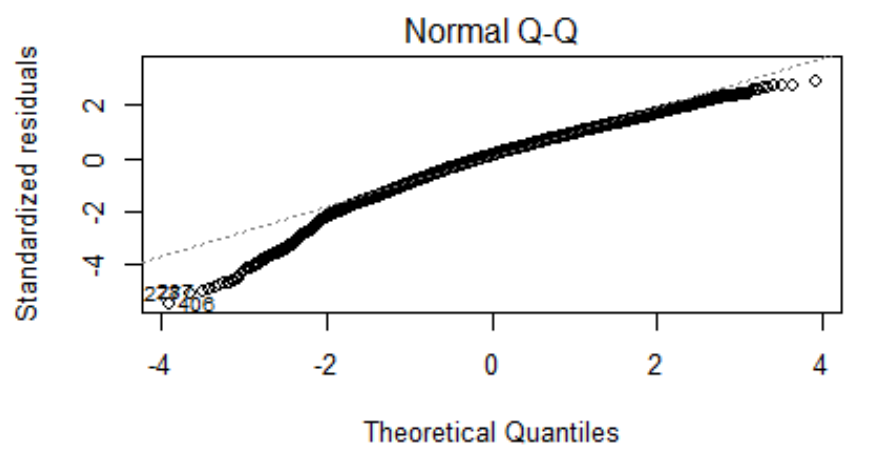
*Table 1. Linear Regression Accuracy on PCA dataset.*

The RMSE here may seem to be the smallest, however, the range of Wage column is -1.328 to 2.333, and thus making the RMSE’s not as impressively low. The normalized RMSE is 0.166, takes into account the range of the dependent variable to make the RMSE scale-free.

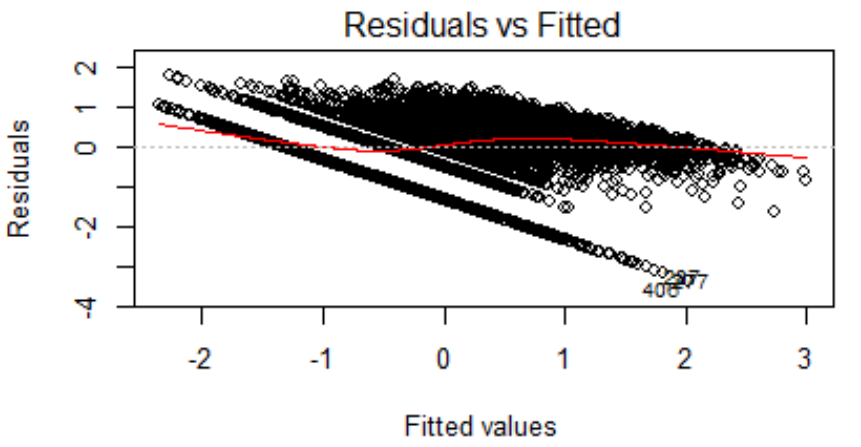
As noticed in the summary of this model, the R-square was 0.6365, which essentially suggests that about 63.65% of the variance of Wage can be explained by the predictors.

**b) Linear Regression without PCA**

Although both Q-Q and Residual-Fitted plot look the same as data with PCA, performance was still not good. See figures 7 and 8, and table 2.

**

*figure 7. Q-Q Plot for linear regression with non-PCA Data.*

**

*figure 8. Residual-Fitted Plot for linear regression (non-PCA).*

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Train RMSE** | **Test RMSE** | **Normalized RMSE** |
| Linear Regression without PCA | 0.6098339 | 0.6143307 | 0.1677749 |

*Table 2. Linear Regression Accuracy on non-PCA dataset*

In comparison, the model’s test RMSE, and normalized test RMSE have increased when non-PCA data was used. Additionally, the R-square was 0.6293, which tells that lesser amount of Wage’s variance can be explained by the predictors, as compared to 63.65% (as seen earlier).

**2) Support Vector Machine**

Support Vector Machine Regression with the e1071 library was used to predict the target variable using the principle components and categorical variables. For SVM training, we used 60% of the data. Using the tune() function in R, we were able to train the model over different cost {.001, .01, .1, 1, 5, 10, 100}and kernel {radial, polynomial, linear} values, each with its own 10-fold cross validation. To elaborate, for each pairing of cost and kernel, a 10-fold cross validation of the training set will be used to train and test a model. For each iteration, the best model, as measured by training error, is returned to the saved R object. Then the iteration continues to the next cost/kernel pair. At the end of the iterations, the R object allows us to retrieve the best parameter pairing as well as the best model for use on the test set. The final, best model had these parameters: Cost = 10 and Kernel = polynomial. The training MSE for this model is 169 with a standard deviation of 80. The test MSE was 220. There were 559 support vectors.

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Train RMSE** | **Test RMSE** | **Normalized RMSE** |
| SVM | 13 | 14.83 | 0.0354 |

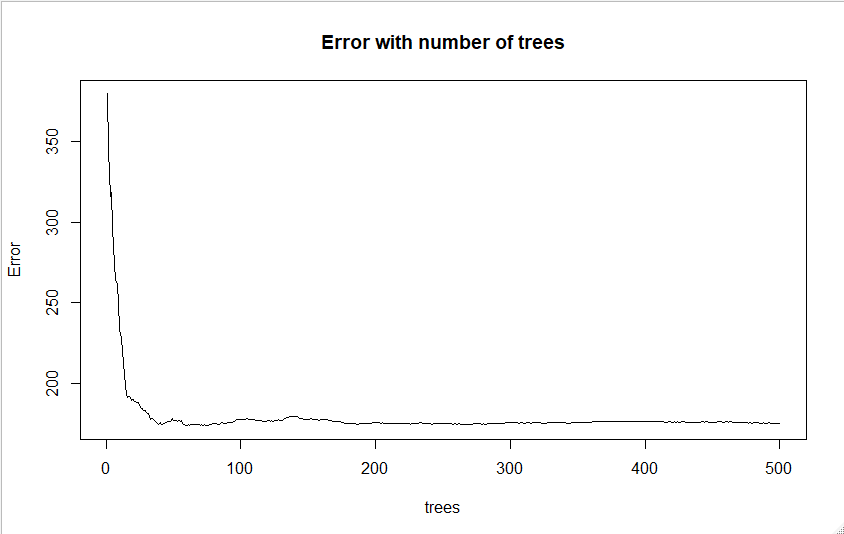
*Table 3. Accuracy results of Support Vector Machine*

**3) Random Forest**

The team wanted to try other non-linear regression models, random forest was our very first option. Based on the complexity of the dataset, we believed that a model like such would perform better.

**a) Random Forest with PCA**

First, the data set was divided into training and testing sets. The training set had 70% of the records and the remaining 30% of data was used for testing. Initially a random forest model was build using the default settings, that is, the trees were equal to 500 and mtry, which is usually the number of predictors divided by 3. This model took few minutes to develop on the training set and the following plot (figure 9) was produced to see the change in error with number of trees. It shows that about 200 trees are enough for this dataset.



*figure 9. Error versus number of trees (RF in PCA data)*

After this, we predicted the wage values using the test set and computed the results. Table 4 shares accuracy results of the training model and predictions, in detail:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Train RMSE** | **Test RMSE** | **Normalized RMSE** |
| Random Forest | 13.2308 | 10.81435 | 0.02677031 |

*Table 4. Random Forest Accuracy on PCA dataset, using default parameter values*

To further improve the results, we attempted to perform cross validation on random forest. The value for k was kept as 5 and the ntree parameter was changed to 200. This is because earlier it was noted that after two hundred trees the error did not change much. However, mtry was not tuned and left as the default value. The model took several minutes and the results were as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Train RMSE** | **Test RMSE** | **Normalized RMSE** |
| Random Forest (CV) | 12.61084 | 12.52065 | 0.02219973 |

*Table 5. Five-Fold Cross Validation on Random Forest Accuracy on PCA dataset (ntree=200)*

Normalized RMSE was used because the range of values in test dataset can be very different from the training set, using which the model was build. Normalized RMSE can be calculated by several methods, for instance, by dividing the RMSE value with mean or standard deviation; but here it was calculated by dividing RMSE (on test set) with the range (of the test set’s Wage). The closer train RMSE and test RMSE values suggest that the model behaves in the same manner for known and unknown values. This means that Random Forest with cross validation was better compared to the previous one. Even the normalized RMSE value is slightly improved for Random forest with cross validation, which suggest that Random forest model with the cross validation performed better.

After performing the random forest on the complete dataset, we tried to remove the outliers. Although this was just to check how the model performs without the extreme values but in actual, we wanted a model that could deal with the data points that are away from the mean. The reason of this was that there are always few players whose wages are quite high compared to the rest of the players. For this model we deleted all the records that had wage values above 45.

The normalized RMSE suggest that the model did not perform that well as the earlier two models.

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Train RMSE** | **Test RMSE** | **Normalized RMSE** |
| Random Forest w/o wage > 45 | 5.447483 | 5.337619 | 0.1241884 |

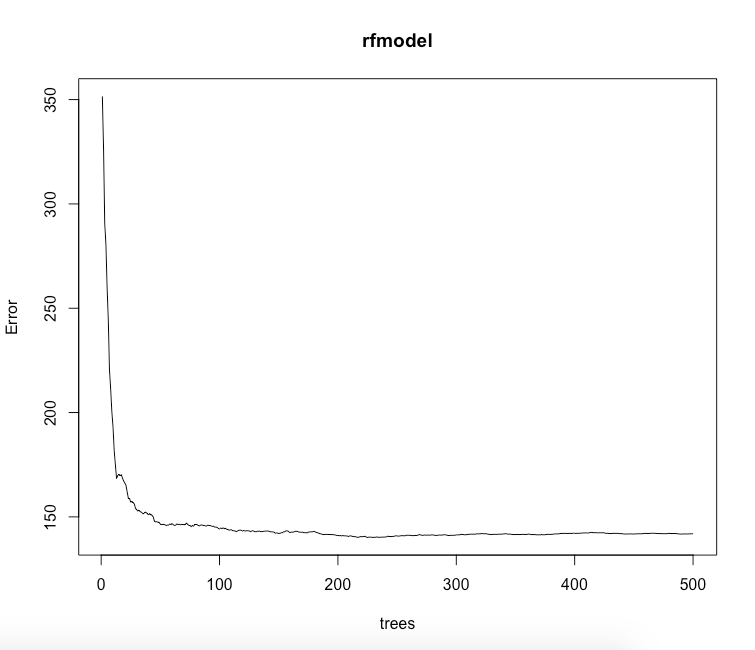
*Table 6. Random Forest Accuracy on PCA dataset (without outliers)*

As far as the random forest models, with PCA, are concerned, the 5-fold cross validation random forest model works best, as noted by the train, test and normalized RMSE.

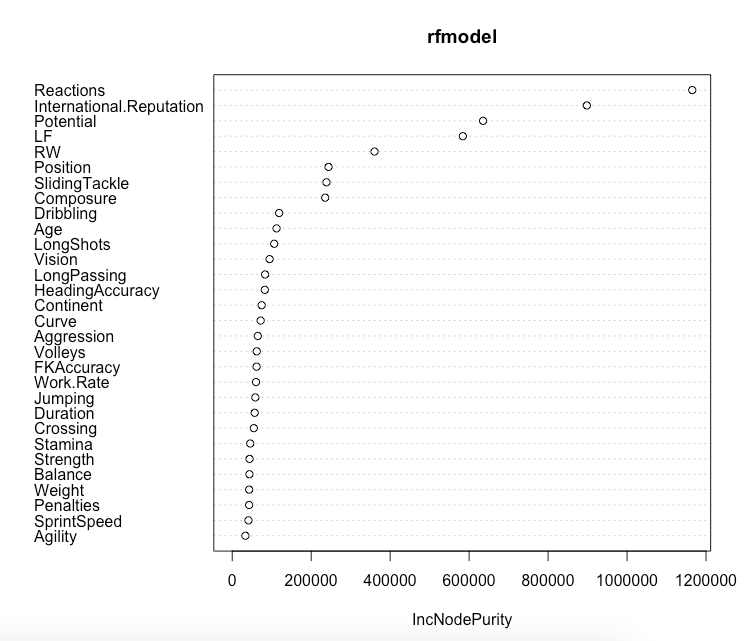
**b) Random Forest without PCA**

Now we wanted to observe the accuracy of the same models, but on data that has not been transformed using PCA. As in the previous section, the dataset was first split into 70% training and 30% test. The Wage was not transformed using Box-Cox for training or prediction model because random forest is known to handle outliers well. Later, building the random forest model using default settings, specifically the ntree equals to 500, and mtry is just the number of predictors divided by three. Because there are 35 predictors in the non-pca dataset (i.e. the numerical and categorical ones), the mtry is equal to 11. The model took roughly 3 to 5 minutes to run on the training dataset. After which the predict function was used to fit the model to the test dataset. The plot in figure 10 shows the change in error as the number of trees increase. It shows that about 200 trees would be enough for this dataset, as with was the case with PCA dataset.

The next graph, figure 10, shows the predictors and their respective node purity. Basically it demonstrates that reactions, international reputation, LF, RW, position, sliding tackle, composure, and dribbling are the predictors on which when the tree splits, the node’s impurity is reduced, significantly, compared to the other predictors used in the model.

****

*figure 10. Error versus number of trees (RF on non-PCA data)*

****

*figure 11. Predictors and their IncNodePurity(RF default parameters)*

Using RMSE (train and test) and normalized RMSE, the performances of this model were assessed. The results can be viewed in the following table:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Train RMSE** | **Test RMSE** | **Normalized RMSE** |
| Random Forest | 11.91268 | 9.633276 | 0.02384489 |

*Table 7. Random Forest Accuracy on non-pca dataset, using default parameters*

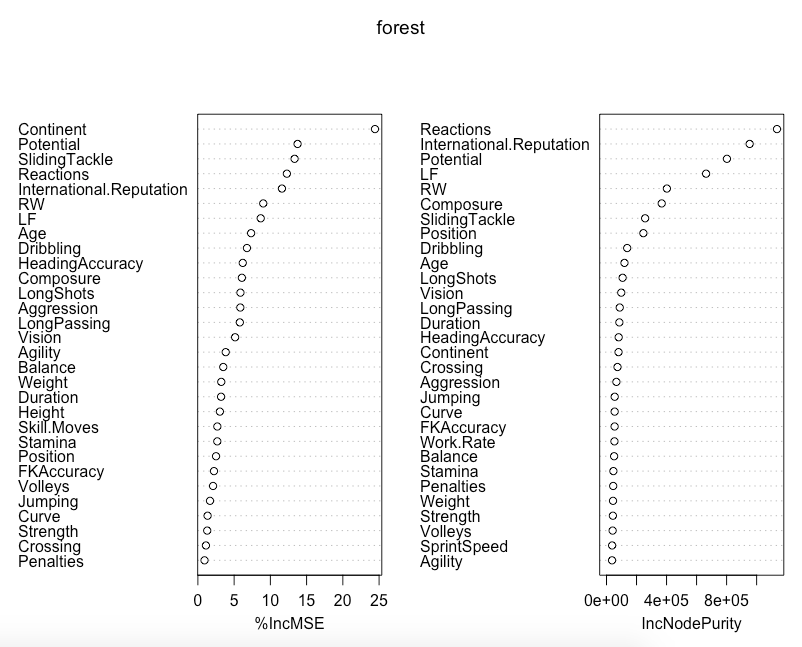
It was interesting to note that the non-pca model was performing better than PCA model. The training, test, and normalized RMSE’s were lower than ones shown in table 1 (earlier).

To further explore if the accuracy improves, we again used a five-fold cross-validated random forest (cv-rf) model on the non-pca data. The parameters were tuned similarly to the previous cross-validated random forest model, with ntree set to 200 and mtry was left as default. It takes several minutes to run on the training dataset. Table 8 shows RMSE of this model.

|  |  |  |
| --- | --- | --- |
| **Method** | **Test RMSE** | **Normalized RMSE** |
| Random Forest (CV) | 11.313349 | 0.02005937 |

*Table 8. Five-Fold Cross Validation on Random Forest Accuracy on non-PCA dataset (ntree=200)*

Again, the cross-validated random forest model on non-pca data performed better than the one on pca data. The normalized RMSE has dropped even lower, by 0.02, which not a significant improvement.



*figure 12. Predictors and their %IncMSE & IncNodePurity( CV-RF default parameters)*

Similar to the results shows in figure 11, when the important variables of cv-rf non-pca model were plotted (figure 12), it can be seen that the same predictors (reactions, international reputation, LF, RW, position, sliding tackle, and composure) are reducing the node’s impurity.

**4) Gradient Boosting**

As the Random Forest models outperformed all the other models the team built using the PCA and non-pca datasets, we decided to try Gradient Boosting Model (GBM) with non-pca dataset.nThe models build using Non-pca data had lower RMSE, and thus this dataset was chosen to build the GBM model.

First, with the default parameters of the gbm function in R were used. This means that the parameters of interaction depth was set to1, shrinkage was set to 0.1, n.trees=100, and distribution was selected to be Gaussian. The accuracy results of the model were as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Train RMSE** | **Test RMSE** | **Normalized RMSE** |
| GBM | 11.4057563 | 11.8666908 | 0.02104023 |

*Table 9. GBM on non-PCA (with default parameters)*

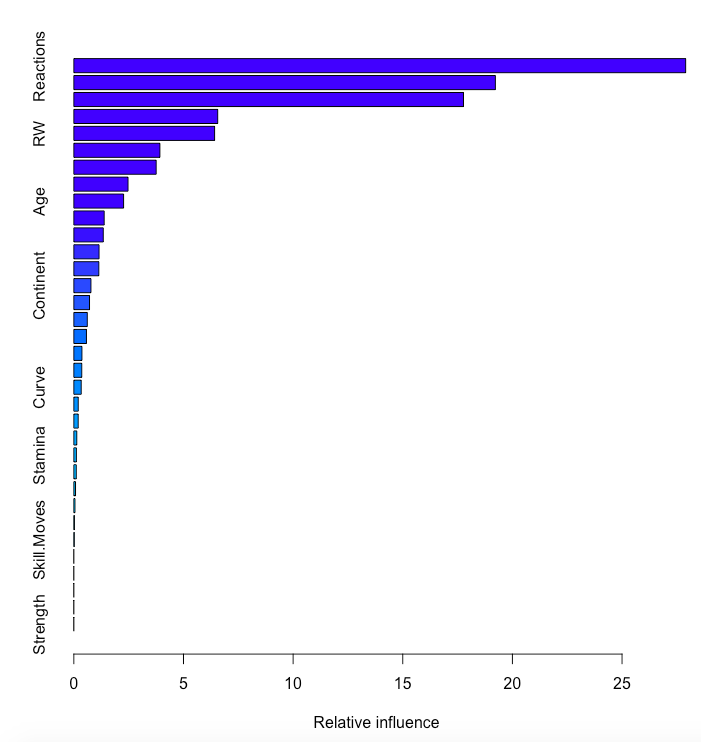
It was expected for the GBM model to perform better than the random forest model. However, the RMSE’s of the first GBM model were higher than that of the 5-fold cv-rf (non-pca data).

Later, after tuning the parameters of GBM, the training RMSE did reduce to 9.70 (approximately), which was much lower than the initial 11.41. The n.trees was increased to 300, interaction depth was 2, and shrinkage was 0.05.

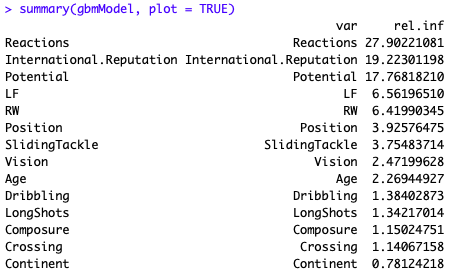
|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Train RMSE** | **Test RMSE** | **Normalized RMSE** |
| GBM | 9.7049918 | 11.2602592 | 0.019965 |

*Table 10. GBM on non-PCA data (with tuned parameters)*

However the test RMSE and the normalized test RMSE did not change by much (as seen in table 10). This could be because GBM tends to over-fit on the training dataset, hence causing the training RMSE to decrease. But, the test RMSE was not that much lower.



*figure 13. Predictors and their Relative Influence.*



*figure 14. Predictors and their Relative Influence (summary of GBM).*

As seen in figure 13 and 14, it can be seen that reactions, International.Reputaion, potential, LF, and RW have the most relative influence on the GBM model, than any of the other independent variable. As one of the objectives of this project was to identify the most important predictors for soccer player wage in FIFA 2019 dataset were, it is safe to assume that the aforementioned variables should definitely be the part of every analysis when trying to predict wage. This is because these were the same variables that even the random forest models determined to increase node purity the most.

**Model Comparison Summary**

After building several machine learning models on this dataset, observing their results, and measuring their respective accuracies, the team thinks that FIFA dataset does not require a PCA transformation, and that a random forest model with 5-fold cross validation would be the best option in models to predict the wage of a particular soccer player based on his/her skill set and attributes; specifically reactions, international reputation, potential, LF, and RW.

**Lesson Learned**

A very important lesson learned during the coarse of this project was thatdata preparation and processing are the most crucial, time-consuming, and rewarding part of any data analysis project. Real datasets are usually incomplete, skewed, noisy and full of outliers, requiring plenty of cleaning, and pre-processing. Such datasets cannot be used to fit a statistical learning model, if used in its raw state.  Therefore, ample time should be spent on data exploration, analysis and preparation before building machine-learning models on the data. The FIFA 2019 dataset preprocessing was the most important step to get an understanding the predictors’ behavior and helped shape our expectations of the kind of results to expect when using the prediction models. Besides taking most of the team’s efforts, this phase of the project helped us determine the most suitable methods for this dataset.

Furthermore, different statistical techniques were used to standardize the variables, and reduce the feature space in order to work with most significant predictors. The Box-Cox approach was used to centralized, scaled and normalized the cleaned data, while PCA was performed for dimensionality reduction as well. When working on this part, it was determined that predictors such as nationality and contract start and end dates were relevant and correlated to our response variable, however had to be changed in a way so that particular models could be fit on them (nationality was mapped to continents, and duration column was calculated using start and end dates of contracts). Making it evident that the number of features should be reduced only after data exploration.

Finally, modeling, predicting, and tuning parameters all together were the interesting part of this project. Assessing and selecting the optimal model parameters should be considered was challenging to some extent. In this section we learned how some models even after tuning the parameter do not perform perfectly or as expected. Such case was presented when performing Gradient Boosting Model (gbm) as it tends to over-fit the training data. Tuning to obtain the best model becomes challenging even with models such as SVM, which can be very time consuming as the model can take up to 24 hours to run, and reach the optimal results.

In conclusion, it was an interesting dataset. When initially downloaded, the team had different assumptions, for example, we expected the linear regression model to be one of the better performing model, and that PCA dataset would produce the best results. However, once we explored the data and the variables’ relationships with one another and Wage, we soon realized that we have to change our assumptions and experiment with several models to get the optimal results.

**Reference**

[1] FIFA 19 Encyclopedia. “Player Attributes.” by FIFA • January 9, 2019 Accessed May 1, 2019. [Online]. Available: https://www.fifplay.com/encyclopedia/player-attributes/

[2] Game Guide. "Footballer Values and Wages.” by FIFA • August 4, 2019 Accessed May 1, 2019. [Online]. Available: <https://isokron.atlassian.net/wiki/spaces/90MF/pages/1377523/Footballer+Values+and+Wages>

[3] RDocumentation. "FindCorrelation: Determine Highly Correlated Variables.” by R• March 15, 2017 Accessed April 1, 2019. [Online]. Available: <https://www.rdocumentation.org/packages/caret/versions/6.0-84/topics/findCorrelation>

[4] M. Kuhn. Contributions from Jed Wing, Steve Weston, Andre Williams, Chris Keefer, Allan

Engelhardt, Tony Cooper, Zachary Mayer, Brenton Kenkel, the R Core Team, Michael Benesty,  Reynald Lescarbeau, Andrew Ziem, Luca Scrucca, Yuan Tang, Can Candan and Tyler Hunt.

(2018). caret: “Classification and Regression Training. R package version 6.0-81.”

[https://CRAN.R-project.org/package=caret](https://cran.r-project.org/package=caret).

[5] A. Liaw and M. Wiener (2002). “Classification and Regression by randomForest”. R News 2(3), 18--22.

[6] T.  Davies, “Book of R,” A First Course in Programming and Statistics, 1st Edition Ed. United States of America. No Starch Press.

[7] R. Ji, “Lecture 2: Data Pre-Processing and PCA”, Department of System Engineering and Operations Research. George Mason University, 2019.

[8] R. Ji, “Lecture 7: Bagging, Random Forest and Boosting”, Department of System Engineering and Operations Research. George Mason University, 2019.

[9] R. Ji, “Lecture 4: Cross-Validation, Model Selection, and Regularization”, Department of System Engineering and Operations Research. George Mason University, 2019.

[10] M. Kuhn and K.l Johnson (2018). AppliedPredictiveModeling: Functions and Data Sets for  “Applied Predictive Modeling”. R package version 1.1-7. [https://CRAN.R-project.org/package=AppliedPredictiveModeling](https://cran.r-project.org/package=AppliedPredictiveModeling)

[11] J. Gareth, D. Witten, T. Hastie,  & R. Tibshirani, (2013). “*An introduction to statistical learning: With applications in R.”* Springer edition 2014, CA.

[12] David Meyer, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel and Friedrich Leisch (2019). e1071: Misc Functions of

 the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. R package version 1.7-1.

 https://CRAN.R-project.org/package=e1071

**Appendix- R code**

**Steps of Running “PlayersWageAnalysisP1.R”:**

This file should prior to all the other R files shared:

Also set the working directory and all files should be in the same directory, you may need to change the loading path in the R-codes.

Run code from

1. Line 1 till line 269 for basic data preprocessing and PCA
2. Line 263 to line 261 for creating a .txt file of the data
3. Line 277 to line 284 clears the environment and loads the data from newly created .txt file
4. Line 287 to line 307 deletes columns that were not required for PCA and then creates another .txt file
5. Line 313 loads the updated data for the PCA file
6. Line 319 to line 329 divides the data into train and test set
7. Line 334 to lone 387 runs simple random forest and calculates RMSE and normalized RMSE
8. Line 392 to line 454 initially clears the environment and then loads the data for random forest with cross validation. Then calculates RMSE and normalized RMSE
9. Line 459 to line 505 clears the environment then loads the data and performs random forest to data that had wage less than 45 and check how the model performs

The other R scripts can be run in any order.

**Steps for running “Fifa-Linear Regression.R”:**

This code contains the models for linear regression on PCA and non-PCA data

Run code from:

For linear regression on PCA data:

1. Line 4 to 14 to load PCA data, remove categorical variables, transforming wage
2. Lines 16 to 30 to split the data into training and test sets, building mode, and gathering results.

For linear regression on non-PCA data:

1. Lines 36 to 47 to clear environment, load non-PCA data file, and transforming wage for linear regression.
2. Lines 50 to 77 to set seed, split data, build run run model, and gather results and plots. Further clearing the environment again.

**Steps for running “SVM PCA Final.R”:**

Run the code as follows:

1. Line 1 to 27, load PCA data, and split dataset into train and test
2. Lines 30 to 42 are creating samples to get understanding of which parameters are best for final training.
3. Lines 46 to 58 contain 1st tune, 2nd tune, 3rd tune, and full tuning of training data
4. Lines 60 to 78, prediction model on test data using best model, and accuracy results of the model (normalized RMSE calculated manually).

**Steps for running “Fifa correlation, RF, GBM.R”:**

Run the code as follows:

1. Lines 1 to 13 for loading data, and exploring the columns.
2. Lines 16 to 36 to select the numeric variables in dataset to form a data frame, checking for correlation coefficients and plotting them.
3. Lines 38 to 48 to use identify which columns to delete
4. Lines 51 to 63, creating updated file with selective numeric variables (that were not dropped after findCorrelation) and categorical variables (that passed the ANOVA test). Creating new .txt file named “fifa5.txt” (however commented out to avoid creating many files of same name).
5. Lines 67 to 163 for loading non-PCA data file (“fifa5.txt”), plotting histograms, boxplots, density plots, skewness, and checking for near zero variance in pre-transformed data, transforming data using preprocess function (center, scale, and box-cox), and again plotting histograms, boxplots, and density plots after transformation and normalization.
6. Lines 165 to 204 to run random forest with default parameters on training and test sets. Also get the RMSE’s and normalized RMSE
7. Lines 205 to 242 for Cross Validation Random Forest, and its respective results (RMSE scores)
8. Lines 245 to 276 to explore and transform wage column (includes histogram, boxplot, density plots pre and post processing)
9. Lines 279 to 342 for default and tuned Gradient Boosting model (with their RMSE’s, normalized RMSE’s)

Run “**Fifa imp categorical vars.R**” file on R to see the aov() test results

Run “**Fifa raw EDA.R”** file to see the exploratory data analysis on raw data. Lines 189-190 perform transformation.