NBA

Sahba Salarian

Feb. 2019

Introduction

The National Basketball Association (NBA) is a men's professional basketball league in North America. It is composed of 30 teams among which 29 teams are in the United States and 1 team in Canada. It is widely considered to be the premier men's professional basketball league in the world.

The NBA is considered one of the four major professional sports leagues in the United States and Canada. The NBA players are the world's best paid athletes by average annual salary per player, among which are Michael Jordan, Kobe Bryant, LeBron James, Kareem Abdul-Jabbar, etc. Considering the huge prevelage gained by these athletes, it is an interesting project to invetigate the career longevity of NBA players in the league based on their athlectic performance in the field.

In this project, the player's career length is devided into two categories of more or less than 5 years, associated with output values of 1 or 0, respectively.

Data Explanation

The data set provides information about field performance of each player, from 1980 to 2016. It consists of 1340 observations with 21 variables. The values for each variable except for the career longevity is calculated as mean per game during the associated rookie year.

The original source of the dataset is the official website for the National Bascketbal Association (www.NBA.com) and the current dataset was retrieved from the data.world repository for data analysis and data competition.

The list of all the variables are shown in Table 1.

Dataset Information

```
str(df)
```

```
## 'data.frame':    1340 obs. of  21 variables:
##  $ Name       : chr  "Brandon Ingram" "Andrew Harrison" "JaKarr Sampson" "Malik Sealy" ...
##  $ GP         : int  36 35 74 58 48 75 62 48 65 42 ...
##  $ MIN        : num  27.4 26.9 15.3 11.6 11.5 11.4 10.9 10.3 9.9 8.5 ...
##  $ PTS        : num  7.4 7.2 5.2 5.7 4.5 3.7 6.6 5.7 2.4 3.7 ...
##  $ FGM        : num  2.6 2 2 2.3 1.6 1.5 2.5 2.3 1 1.4 ...
##  $ FGA        : num  7.6 6.7 4.7 5.5 3 3.5 5.8 5.4 2.4 3.5 ...
##  $ FG.        : num  34.7 29.6 42.2 42.6 52.4 42.3 43.5 41.5 39.2 38.3 ...
##  $ X3P.Made   : num  0.5 0.7 0.4 0.1 0 0.3 0 0.4 0.1 0.1 ...
##  $ X3PA       : num  2.1 2.8 1.7 0.5 0.1 1.1 0.1 1.5 0.5 0.3 ...
##  $ X3P.       : num  25 23.5 24.4 22.6 0 32.5 50 30 23.3 21.4 ...
##  $ FTM        : num  1.6 2.6 0.9 0.9 1.3 0.4 1.5 0.7 0.4 1 ...
##  $ FTA        : num  2.3 3.4 1.3 1.3 1.9 0.5 1.8 0.8 0.5 1.4 ...
##  $ FT.        : num  69.9 76.5 67 68.9 67.4 73.2 81.1 87.5 71.4 67.8 ...
##  $ OREB       : num  0.7 0.5 0.5 1 1 0.2 0.5 0.8 0.2 0.4 ...
##  $ DREB       : num  3.4 2 1.7 0.9 1.5 0.7 1.4 0.9 0.6 0.7 ...
```

```
##  $ REB        : num  4.1 2.4 2.2 1.9 2.5 0.8 2 1.7 0.8 1.1 ...
##  $ AST        : num  1.9 3.7 1 0.8 0.3 1.8 0.6 0.2 2.3 0.3 ...
##  $ STL        : num  0.4 1.1 0.5 0.6 0.3 0.4 0.2 0.2 0.3 0.2 ...
##  $ BLK        : num  0.4 0.5 0.3 0.1 0.4 0 0.1 0.1 0 0 ...
##  $ TOV        : num  1.3 1.6 1 1 0.8 0.7 0.7 0.7 1.1 0.7 ...
##  $ TARGET_5Yrs: num  0 0 0 1 1 0 1 1 0 0 ...
```

The TARGET_5Yrs should be analyzed as a binary class, versus other variables for each athlete. The variables are demonstrated in Table 1.

Column Names	Explanation
name	ASCII subject name and recording number
Name	Name
GP	Games Played
MIN	Minutes Played
PTS	Points Per Game
FGM	Field Goals Made
FGA	Field Goals Attempts
FG.	Field Goals Percent
X3P.Made	3Points Made
X3PA	3Points Attempts
X3P.	3Points Attempts Percentage
FTM	Free Throw Made
FTA	Free throw Attempts
FT.	Free throw Percenrage
OREB	Offensive Rebounds
DREB	Defenive Rebounds
REB	Rebounds
AST	Assists
STL	Steals
BLK	Blocks
TOV	Turnovers
TARGET-5Yrs	Outcome=1(career length>=5 yrs),Outcome=0(career length<5)

Table 1: Attribute Information

Data Engineering

In this stage the NA values of the data set are detedcted and the rows with such unknown or missing values are omitted from the data set. Moreover, the Name column is also removed from the dataset. 1329 observations and 20 variables are remaining as follows:

```
str(df)
```

```
## 'data.frame':    1329 obs. of  20 variables:
##  $ GP         : int  36 35 74 58 48 75 62 48 65 42 ...
##  $ MIN        : num  27.4 26.9 15.3 11.6 11.5 11.4 10.9 10.3 9.9 8.5 ...
##  $ PTS        : num  7.4 7.2 5.2 5.7 4.5 3.7 6.6 5.7 2.4 3.7 ...
##  $ FGM        : num  2.6 2 2 2.3 1.6 1.5 2.5 2.3 1 1.4 ...
##  $ FGA        : num  7.6 6.7 4.7 5.5 3 3.5 5.8 5.4 2.4 3.5 ...
##  $ FG.        : num  34.7 29.6 42.2 42.6 52.4 42.3 43.5 41.5 39.2 38.3 ...
##  $ X3P.Made   : num  0.5 0.7 0.4 0.1 0 0.3 0 0.4 0.1 0.1 ...
##  $ X3PA       : num  2.1 2.8 1.7 0.5 0.1 1.1 0.1 1.5 0.5 0.3 ...
##  $ X3P.       : num  25 23.5 24.4 22.6 0 32.5 50 30 23.3 21.4 ...
##  $ FTM        : num  1.6 2.6 0.9 0.9 1.3 0.4 1.5 0.7 0.4 1 ...
##  $ FTA        : num  2.3 3.4 1.3 1.3 1.9 0.5 1.8 0.8 0.5 1.4 ...
##  $ FT.        : num  69.9 76.5 67 68.9 67.4 73.2 81.1 87.5 71.4 67.8 ...
##  $ OREB       : num  0.7 0.5 0.5 1 1 0.2 0.5 0.8 0.2 0.4 ...
##  $ DREB       : num  3.4 2 1.7 0.9 1.5 0.7 1.4 0.9 0.6 0.7 ...
##  $ REB        : num  4.1 2.4 2.2 1.9 2.5 0.8 2 1.7 0.8 1.1 ...
##  $ AST        : num  1.9 3.7 1 0.8 0.3 1.8 0.6 0.2 2.3 0.3 ...
##  $ STL        : num  0.4 1.1 0.5 0.6 0.3 0.4 0.2 0.2 0.3 0.2 ...
##  $ BLK        : num  0.4 0.5 0.3 0.1 0.4 0 0.1 0.1 0 0 ...
##  $ TOV        : num  1.3 1.6 1 1 0.8 0.7 0.7 0.7 1.1 0.7 ...
##  $ TARGET_5Yrs: num  0 0 0 1 1 0 1 1 0 0 ...
```

Train & Test Split:

Just because a learning algorithm fits a data set well, does not guarantee that it is a good hypothesis. It could overfit and as a result the predictions on the other data set would be poor. The error of hypothesis as measured on the data set with which we trained the parameters will usually be lower than the errors on any other data set. So for the sake of better evaluation of the model and better prediction analysis the data set is randomly splitted into two separate datasets of "train" and "test", with 70% and 30% of the whole data, respectively. The train and test datasets respectively have 921 and 408 observatinos with 20 variables.

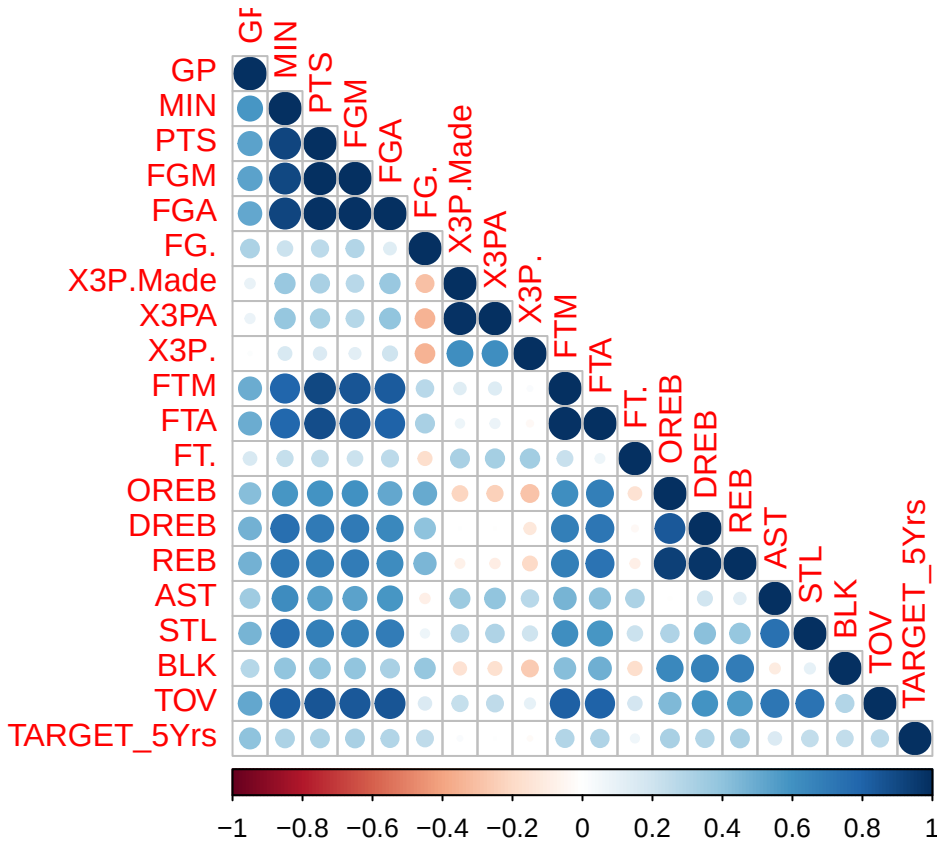

Figure 1: Correlation Matrix

Correlation Matrix

At the very first step, because of the importance and to get a better understanding of the dataset, correlation matrix for the predictors is plotted. The dark blue color between not-identical variables shown in Figure 1, demonstrates some non-negligible input feature correlations.

The distribution and scatterplot matrix has been added to have a better understanding of the correlation between the features.

Regarding the correlated features

As illustrated in correlation and scatterplot matrices, some features are very correlated with each other. Different methods exist for omitting the collinearity, e.g. removing the set of correlated features and just keeping those more correlated with outcome or removing the features with lower variances. In this project the **caret** package has been used to find the best trained model based on automated training package in R.

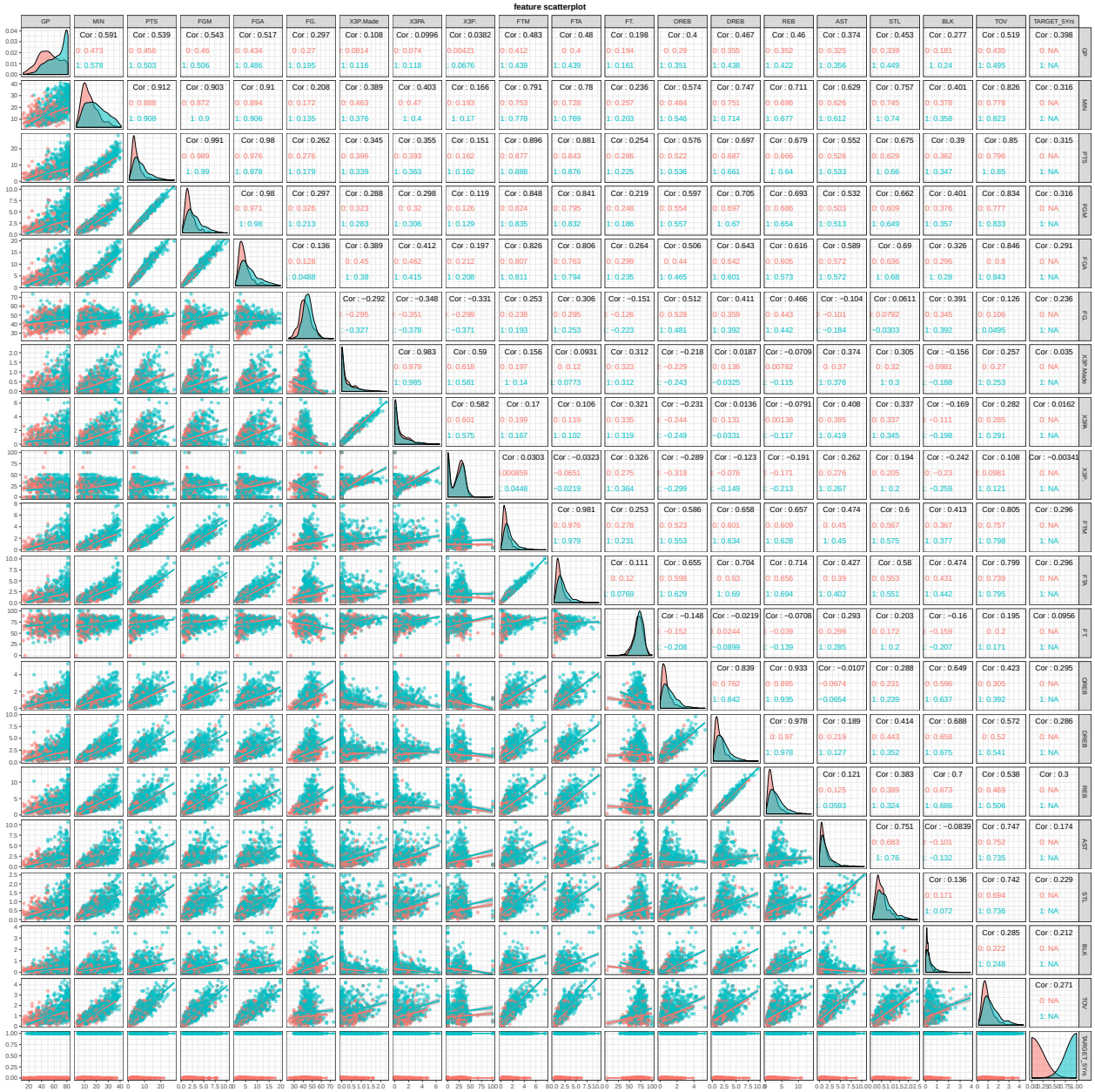

Figure 2: Scatterplot matrix for predictors

Fitting simple logistics regression model:

A simple logistic regression model with all inputs is fitted to the dataset, the summary of the fit, deviance, residual deviance and Chi-squared test analysis are shown in Table 2. As illustrated in Table 2, many features with high Pr(>Chi) are insignificant predictors.

Table 2: summary of logistic regression fit with all inputs

	Df	Deviance	Resid. Df	Resid. Dev	Pr(>Chi)
NULL			920	1212.97	
GP	1	153.73	919	1059.24	0.0000
MIN	1	12.93	918	1046.31	0.0003
PTS	1	8.61	917	1037.70	0.0033
FGM	1	1.69	916	1036.01	0.1932
FGA	1	9.19	915	1026.82	0.0024
FG.	1	2.72	914	1024.09	0.0989
X3P.Made	1	0.71	913	1023.39	0.4006
X3PA	1	8.13	912	1015.26	0.0044
X3P.	1	0.03	911	1015.23	0.8660
FTM	1	0.00	910	1015.23	0.9818
FTA	1	0.24	909	1014.99	0.6233
FT.	1	0.97	908	1014.02	0.3250
OREB	1	13.80	907	1000.22	0.0002
DREB	1	0.51	906	999.71	0.4770
REB	1	0.29	905	999.43	0.5926
AST	1	4.78	904	994.64	0.0287
STL	1	1.28	903	993.36	0.2578
BLK	1	7.88	902	985.48	0.0050
TOV	1	1.30	901	984.18	0.2544

Pseudo-R squared and mis-classification rate are also computed for the logistic fit, demonstrated in Table 3.

Table 3: simple logistic model accuracy for train dataset

Parametrs	R-squared	eta	accuracy
Values	30.0000000	0.2627579	73.7000000

Using caret for Training Machine Learning Models

The **caret** package in R, is a complete package used to train different machine learning algorithms. For applying the training procedure a trainControl(), cross validation method, shall be defined. The train cotrol for this project has been set on "bootstrap" with 25 as the number, the default setting in R, with TRUE probability class. Although there is a chance that bootstrap method gives underfit results in some cases, this becomes negligible when the observations are large enough. Since there are 1329 observations with 19 predictors in this project, the chance of underfit results can be neglected.

Use GLM to fit the model

The first model trained in this project is GLM method, applied on the train test by using the **caret::train()** function.

Generalized Linear Model

921 samples 19 predictor 2 classes: 'zero', 'one'

No pre-processing Resampling: Bootstrapped (25 reps) Summary of sample sizes: 921, 921, 921, 921, 921, 921, ... Resampling results:

Accuracy Kappa
0.7122912 0.3672358 over the train dataset. The fit coefficients are showin in Table 5.

Table 4: Trained GLM with all inputs-bootstrap

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-5.4359	1.5067	-3.61	0.0003
GP	0.0362	0.0058	6.28	0.0000
MIN	-0.0960	0.0400	-2.40	0.0165
PTS	-0.2990	1.0773	-0.28	0.7814
FGM	-0.0572	2.1562	-0.03	0.9788
FGA	0.4576	0.2833	1.62	0.1063
FG.	0.0456	0.0265	1.72	0.0859
X3P.Made	3.6777	1.5945	2.31	0.0211
X3PA	-1.2014	0.4905	-2.45	0.0143
X3P.	0.0046	0.0069	0.67	0.4999
FTM	0.1090	1.2393	0.09	0.9299
FTA	0.1414	0.5647	0.25	0.8022
FT.	0.0164	0.0123	1.34	0.1811
OREB	0.0823	1.6264	0.05	0.9597
DREB	-1.0950	1.6244	-0.67	0.5003
REB	0.9599	1.6195	0.59	0.5534
AST	0.3047	0.1353	2.25	0.0243
STL	0.5301	0.4051	1.31	0.1907
BLK	0.9393	0.3481	2.70	0.0070
TOV	-0.3746	0.3291	-1.14	0.2550

Based on the summary of the GLM model over all inputs, the features of PTS, FGM, FGA, X3P.,FTM, FTA,FT.,OREB, DRE, REB, STL and TOV, are shown insignificant.
Considering the consideration of insignificant features in this model the next investigated model is the AIC model to check the chance of feature removal.

The confusion matrix for GLM model is also represented in following table, Table 6. For the test dataset the accuracy is 65.93% and kappa value is 27.43%

Table 5: GLM Confusion Matrix

	zero	one
zero	83	59
one	80	186

Using glmStepAIC to fit the model.

over train data set. As it is illustrated in the trained AIC model summary table, Table 7. All the final features, the intercept, GP, MIN, FGA, X3P.Made, X3PA, FT., DREB, REB,AST and BLK are in the 95% significance interval.

Table 6: Fit coefficients for trained AIC-bootstrap

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-3.3579	0.6326	-5.31	0.0000
GP	0.0368	0.0056	6.52	0.0000
MIN	-0.0692	0.0368	-1.88	0.0600
FGA	0.1127	0.0619	1.82	0.0688
X3P.Made	3.4917	1.1526	3.03	0.0025
X3PA	-1.1867	0.4274	-2.78	0.0055
FT.	0.0144	0.0085	1.69	0.0909
DREB	-1.2947	0.3674	-3.52	0.0004
REB	1.0892	0.2698	4.04	0.0001
AST	0.2599	0.1036	2.51	0.0121
BLK	0.9135	0.3491	2.62	0.0089

Table 7: GLM Confusion Matrix

	zero	one
zero	83	59
one	80	186

Confusion Matrix for AIC model, Table 8, shows 65.93% accuracy for the test dataset with 27.43% kappa.

Support Vector Machine method

In this section svm model is used combined with **caret::train** function. since there exist a chnce of linear decision boundary for our dataset we consider SVM fit with both linear and radial kernels and investigate the results.

SVM with Linear kernel

Linear kernel is considered for svm fit with R's default train control method, boostrap and TRUE probability and scaling. Scaling is an important factor in SVM since normalization of variables leads to much uniform and stable computations.

```
## Support Vector Machines with Linear Kernel
##
## 921 samples
##  19 predictor
##   2 classes: 'zero', 'one'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 921, 921, 921, 921, 921, 921, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.7124764  0.3562906
##
## Tuning parameter 'C' was held constant at a value of 1
```

As mentioned the accuracy with linear kernel in SVM fit, over the trained set is 71.25% with kappa 35.36%. Confusion matrix for the model also shows the accuracy of 66.91% for test data set with kappa 28.31%.

Table 8: SVM with linear kernel Confusion Matrix

	zero	one
zero	77	49
one	86	196

SVM with radial kernel:

The default and usual kernel for SVM fit is the gaussian or radial kernel. SVM fit with radial kernel is also investigate din this project.

```
## Length  Class   Mode
##      1   ksvm     S4
```

```
## Support Vector Machines with Radial Basis Function Kernel
##
## 921 samples
##  19 predictor
##   2 classes: 'zero', 'one'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 921, 921, 921, 921, 921, 921, ...
## Resampling results across tuning parameters:
##
##   C     Accuracy   Kappa
##   0.25  0.7139483  0.3579176
##   0.50  0.7124723  0.3545496
##   1.00  0.7098730  0.3513038
##
## Tuning parameter 'sigma' was held constant at a value of 0.06461464
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were sigma = 0.06461464 and C = 0.25.
```

The model gives accuracy of 71.39% with kappa 35.79% for train test. The tunning features, sigma and cost based on the mentioned settings were set on 0.065 and 0.25, respectively. Creating the confusion matrix for the test set, gives 66.67% accuracy with kappa 26.29%.

Table 9: SVM with radial kernel Confusion Matrix

	zero	one
zero	67	40
one	96	205

Comparison between the different trained models:

The comparison is made between the trained GLM, AIC, SVM with linear kernel and SVM with radial kernel fits, by using **resample()** function in R. The summary of the comparison is illustrated as following.

```
##
## Call:
## summary.resamples(object = comparison)
##
## Models: GLM, AIC, SVMLinear, SVMRadial
## Number of resamples: 25
##
## Accuracy
##                Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## GLM       0.6549708 0.6991150 0.7102273 0.7122912 0.7230321 0.7706422    0
## AIC       0.6638177 0.6880223 0.7002967 0.7076136 0.7272727 0.7584098    0
## SVMLinear 0.6696697 0.6980057 0.7105263 0.7124764 0.7238372 0.7553517    0
## SVMRadial 0.6696697 0.6970588 0.7109145 0.7139483 0.7254902 0.7675841    0
##
## Kappa
##                Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## GLM       0.2636035 0.3465347 0.3619019 0.3672358 0.3831521 0.4933584    0
## AIC       0.2670418 0.3265982 0.3531209 0.3573470 0.3851695 0.4700597    0
## SVMLinear 0.2696198 0.3283564 0.3495401 0.3562906 0.3985590 0.4305864    0
## SVMRadial 0.2592218 0.3358366 0.3526248 0.3579176 0.3789183 0.4764914    0
```

ROC for the different fitted models

As another evaluation method for finding the best fit, the ROC curve has been drawn for all the fitted models in this project and the AUC values are calculated for each fit. AUC value which is the area under the ROC curve, can be interpreted as the probability that a random chosen instance from Y=1 population will have a higher score than a randomly chosen instance from Y=0 population. This parameter provides a measure for investigating the goodness of prediction.

First we have the simple logistic model with full features, it has AUC value of 0.732. The ROC for simple logistic curve is shown in Figure 3.

ROC for the GLM model trained by caret package, illustrated in Figure 4, presents similar AUC value to the simple logistic regression fit, meaning that the two models can be considered equal.

ROC for the fit with AIC model, differs in AUC value, Figure 5 shows AUC=0.738 for this model. A slightly better prediction for the test detaset.

By applying the SVM model with linear kernel the area under the curve has improved to AUC=0.742, illustrated in Figure 6.

And finally the SVM model with radial fit has AUC=0.718 over the test dataset, which is lower than the SVM model with linear kernel.

Comparing the ROC curves and more specificly the AUC values, it is conculed that SVM fit with linear kernel is the best model among other analyzed fits which has AUC value of 74.2 %.

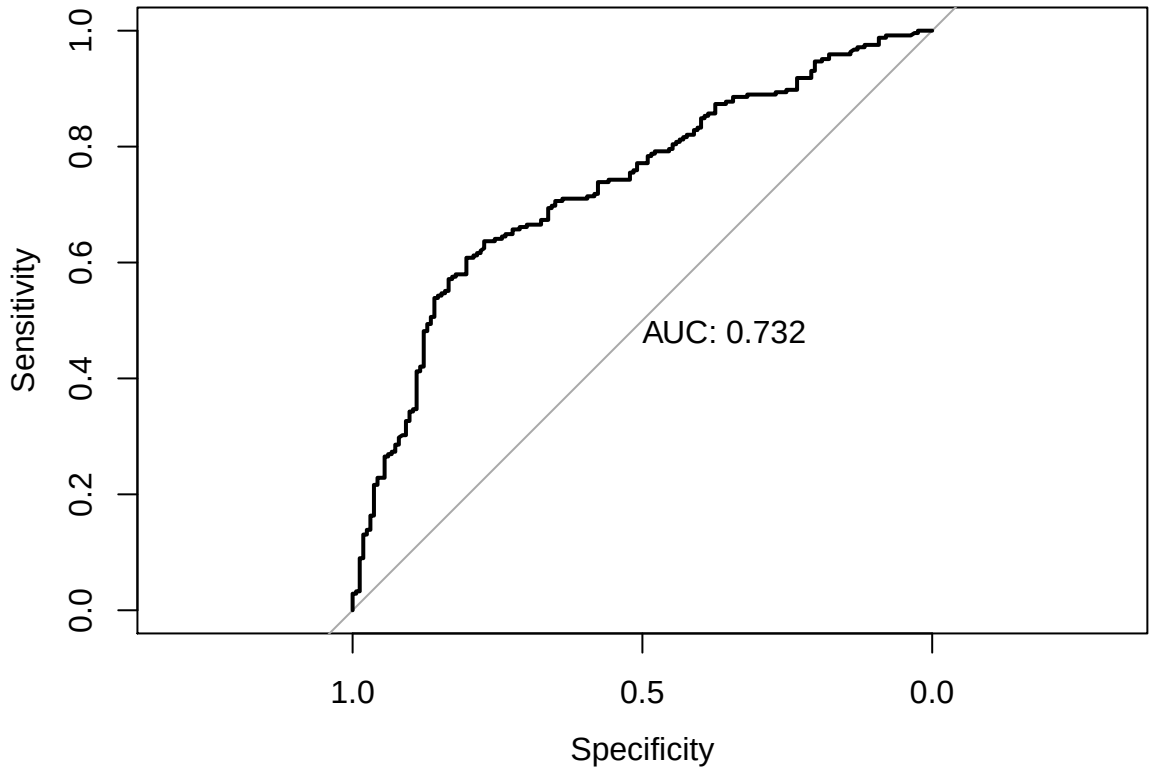

Figure 3: ROC for simple Logistic Fit

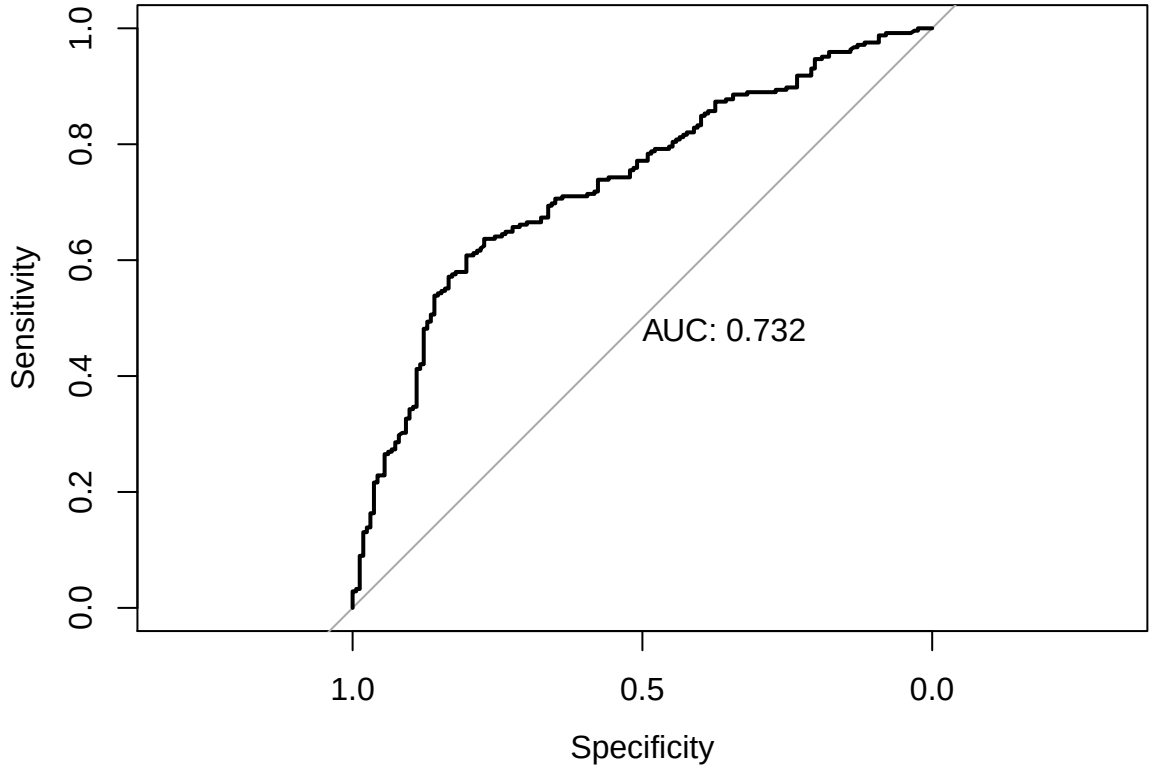

Figure 4: ROC for GLM

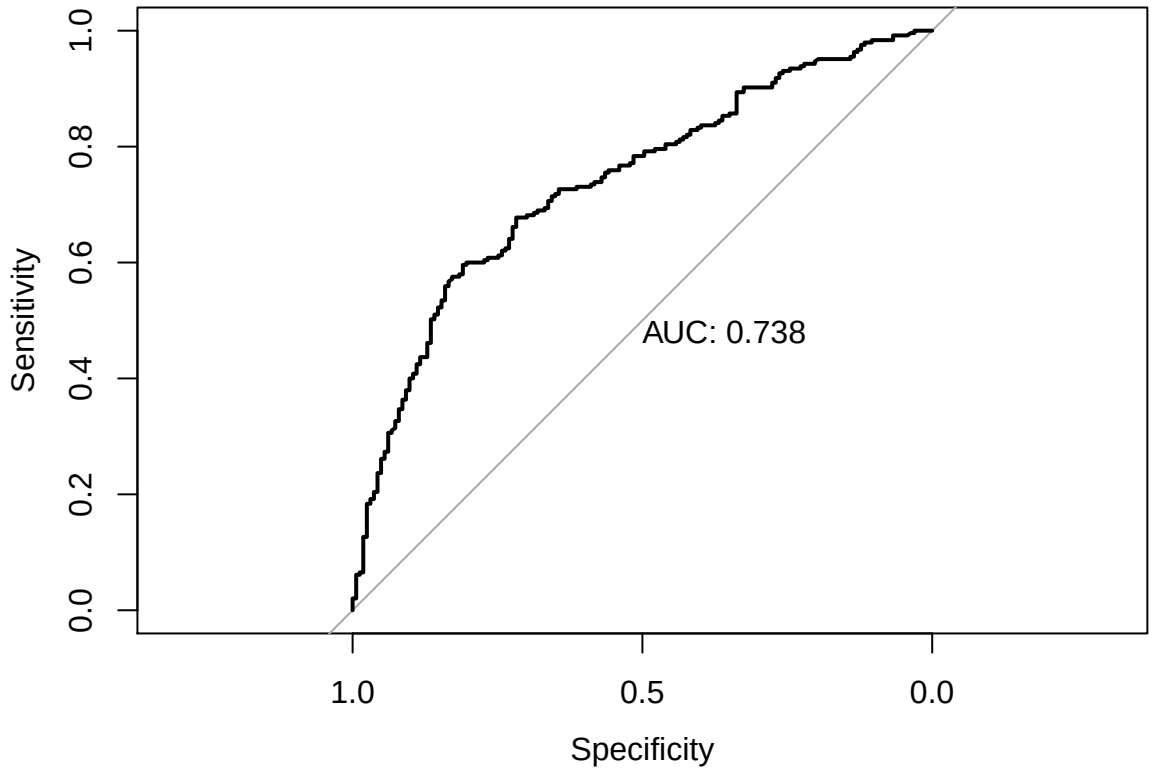

Figure 5: ROC for AIC

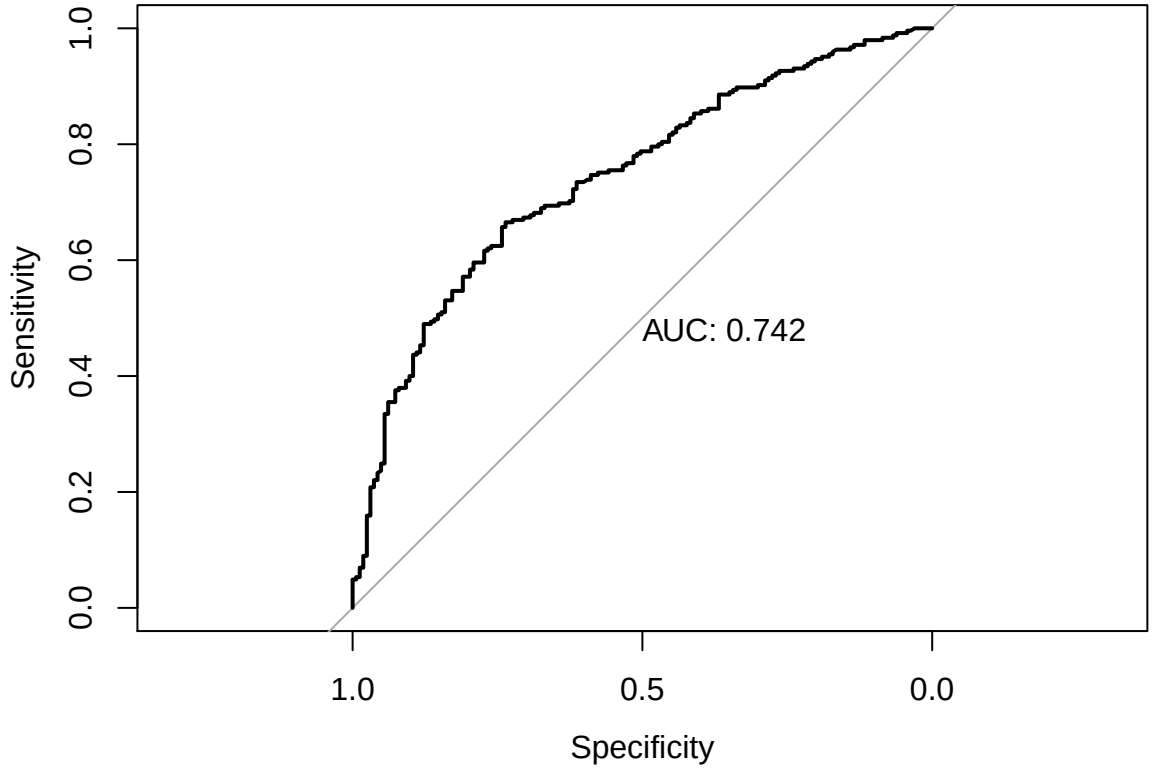

Figure 6: ROC for SVM with linear kernel

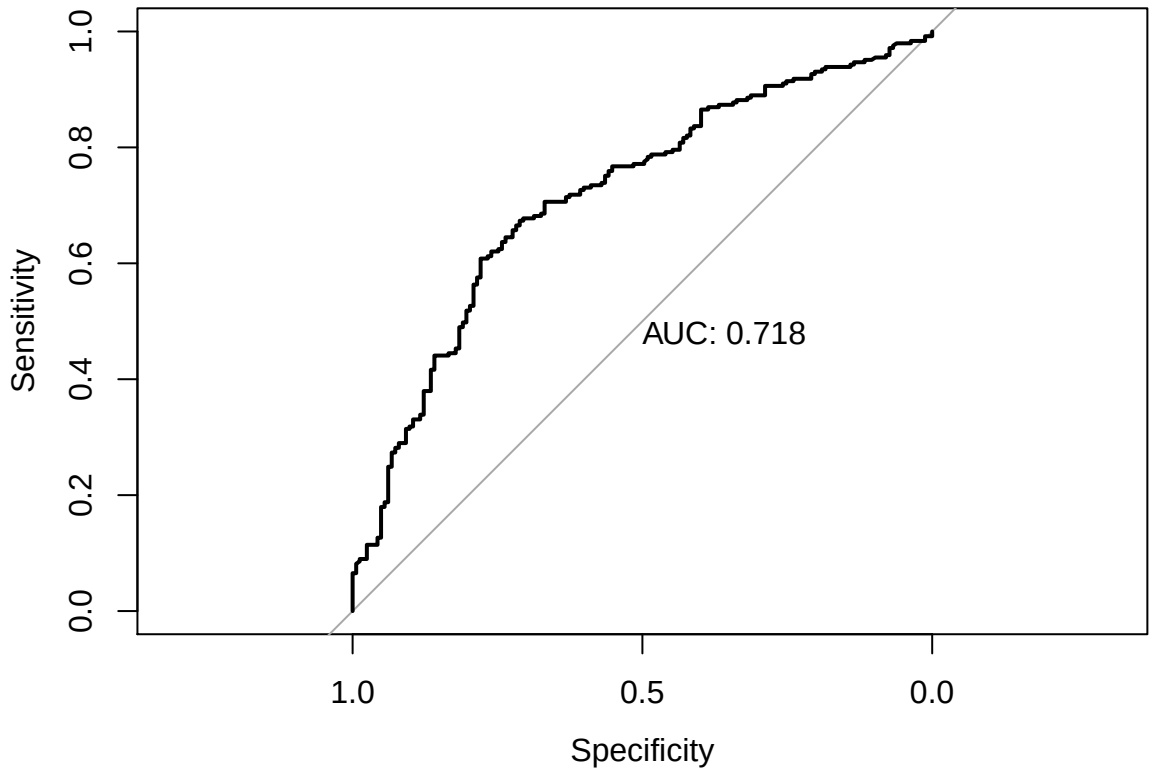

Figure 7: ROC for SVM with radial kernel

Conculsion:

The career longevity of NBA players, with respect to their athletic performance is provided in the analyzed dataset for this project. The outcome is a binary classification based on the years each player stayed in the league >= or < than 5 years. There were 1340 observations and 21 variables. The dataset has been splitted into two "train" and "test" separate dataset.

The Caret packge in R has been used over the train dataset for training the machine learning algorithms. The boostrap method has been used to control the training with 4 different methods of GLM, glmStepAIC, SVM with linear kernel and SVM with radial Kernel, creating different fits. The accuracy and kappa value are analyzed for all the fitted models. The best fits from the different mentioned training methods were then compared with each other, regarding the best prediction over the test dataset. The comparison has been made with resample() function from caret package in R and the ROC curves. The results show that the best fit among the investigated methods with bootstrap process, is SVM with linear kernel, with AUC value of 74.2%.