CS584 – MACHINE LEARNING

FALL 2016

"Predict the overall ranking of a football player"

Group Members: ***Arati Bhat, Madhu Lokanath*** **and** ***Nithin Ashok***

Table of Contents

Predict the overall ranking of a football player

Group Members: ***Arati Bhat, Madhu Lokanath*** **and** ***Nithin Ashok***

Task

In this project, **Predict the overall ranking of a European Football players**, we are predicting the overall ranking of European Football players(which in our case is target variable) by considering the individual player statistics such as player's potential, agility, stamina and so on as the features(instances). So with the individual player's overall ranking, it will be easy for the bidders to choose the players for their team at the auction hence it is an interesting project to start with.

It is a regression problem, where we are basically building a supervised learning model to predict the overall ranking of football players. To train this model, we collected data from various websites which provides various kind of information about players and their statistics. Once the model is trained using the data collected above, the model is able to predict the overall ranking of any new upcoming players.

Dataset

We agreed that the regular-season data of European Football would be the most effective for building a machine-learning model that would model the correlation of a set of features into an overall ranking of everyone. That way, we could predict the overall ranking of any player in future given their set of features.

The information we have extracted explains a few key factors that affect the player's overall ranking, such as potential, crossing, finishing, short passing, free kick accuracy, ball control, sprint speed and so on.

Data source

As there's no lack of sites hosting data on football games, and we found them on http://www.football-data.co.uk/englandm.php and on kaggle website, which had the cumulative data on all the teams for many years.

It looks like a giant set of spreadsheets linked together on websites. Getting the data into the format we needed required a bit of filtering. But eventually we had a nice comma separated value (CSV) file with the 2008 to 2015 regular-season historical data. This had lot of null and missing values which we did preprocessing and there was some entries missing which we calculated and added manually.

Target variable

'Overall Ranking' of a player.

Features

In our project, the input features are the player's attributes. Here we selected 22 important features which affects the player's overall ranking (target variable). Those 22 features are as below:

1) acceleration
2) aggression
3) agility
4) balance
5) ball_control
6) crossing
7) curve
8) dribbling
9) finishing
10) free_kick_accuracy
11) jumping
12) long_passing
13) long_shots
14) penalties
15) potential
16) short_passing
17) shot_power
18) sliding_tackle
19) sprint_speed
20) stamina
21) standing_tackle
22) vision

Data size

Our dataset has 10849 rows and 23 columns where rows being players and column being features of the player.

Preprocessing

The data which we collected had multiple rows for each player and were whooping 183000 rows and hence we decided to take the aggregate of the data with respective to each player and hence ended up with one row for each player with cumulative performance details of each attribute of a player.

Visualization

Target

Overall rating's mean and variance:

mean	66.797304
std	6.226787

- Below is the class visualization which in our case is the overall rating.

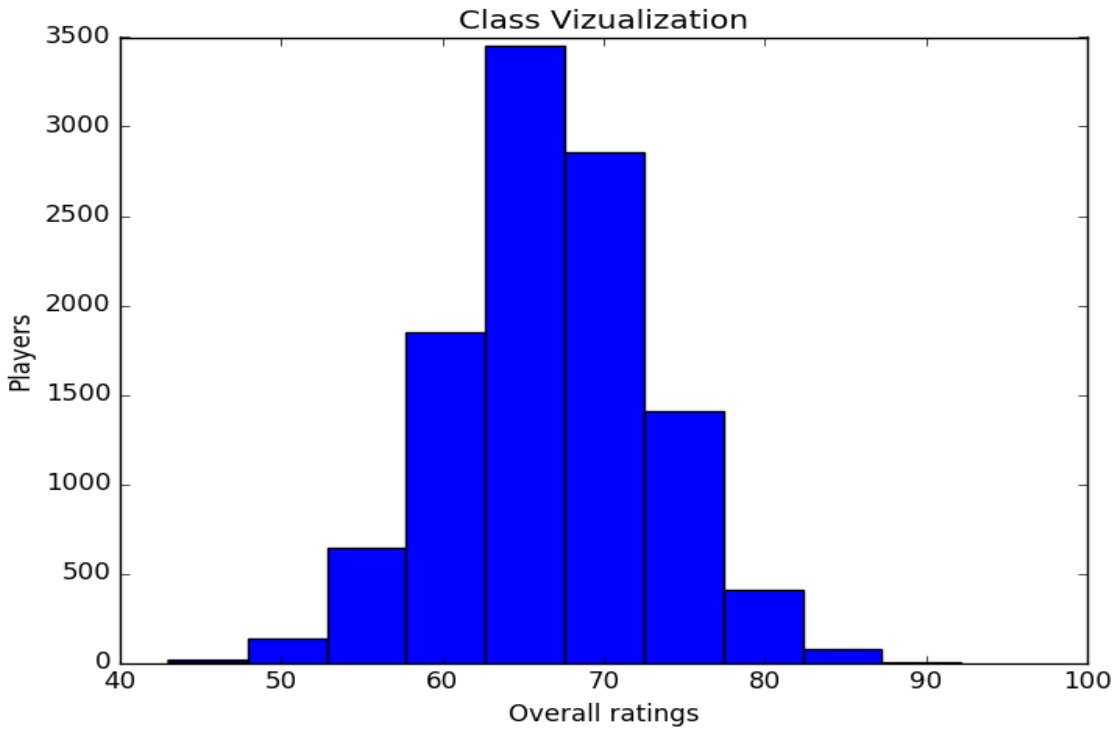
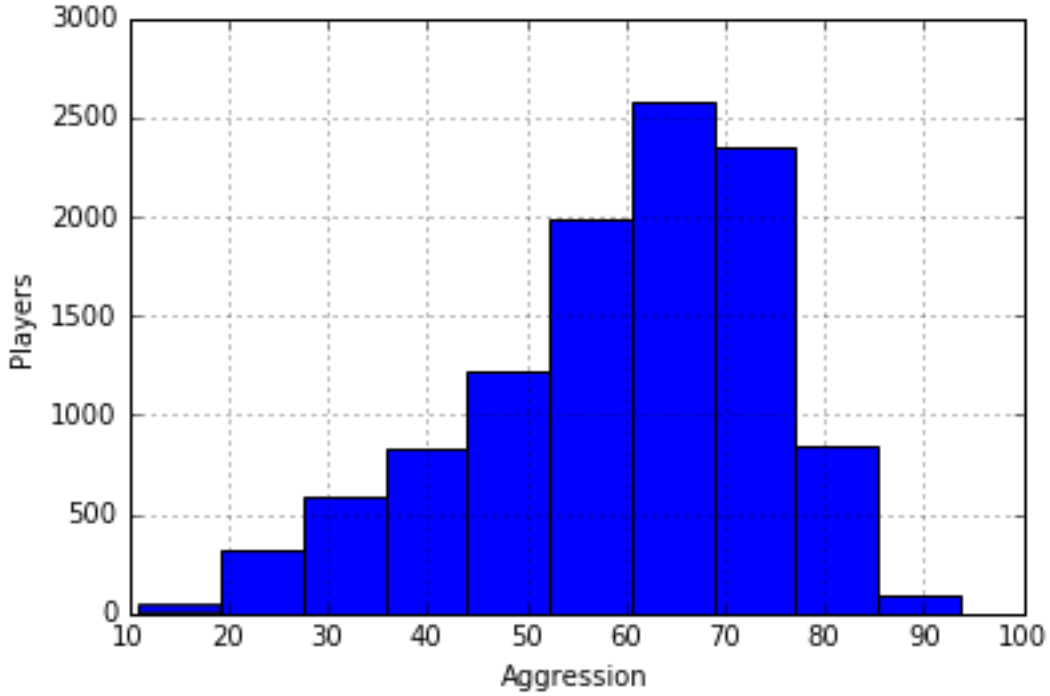
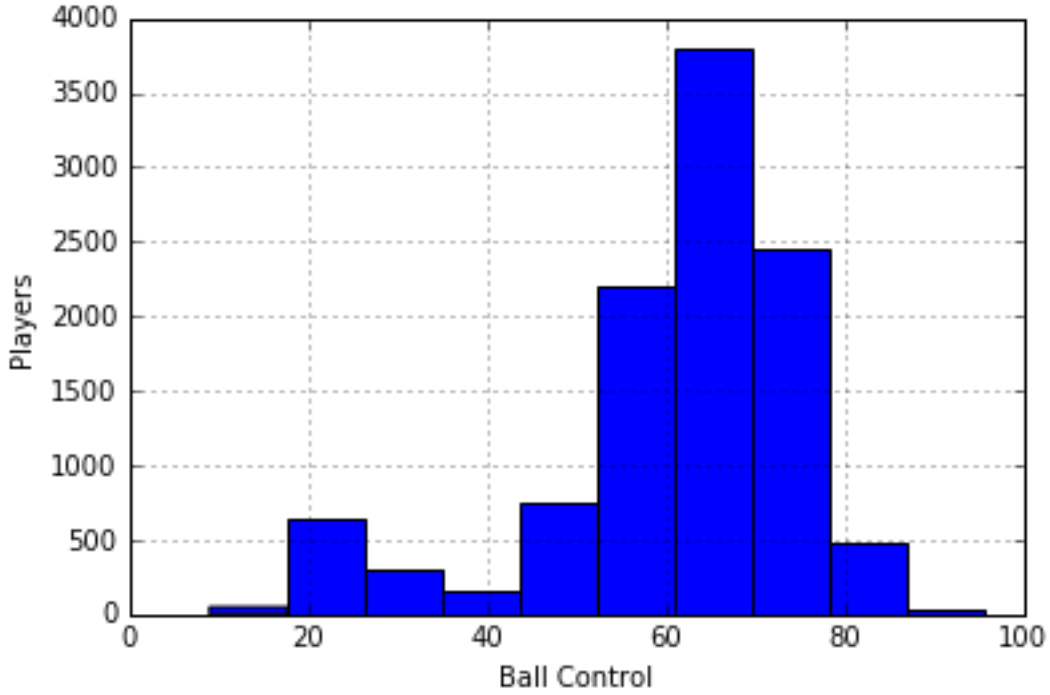


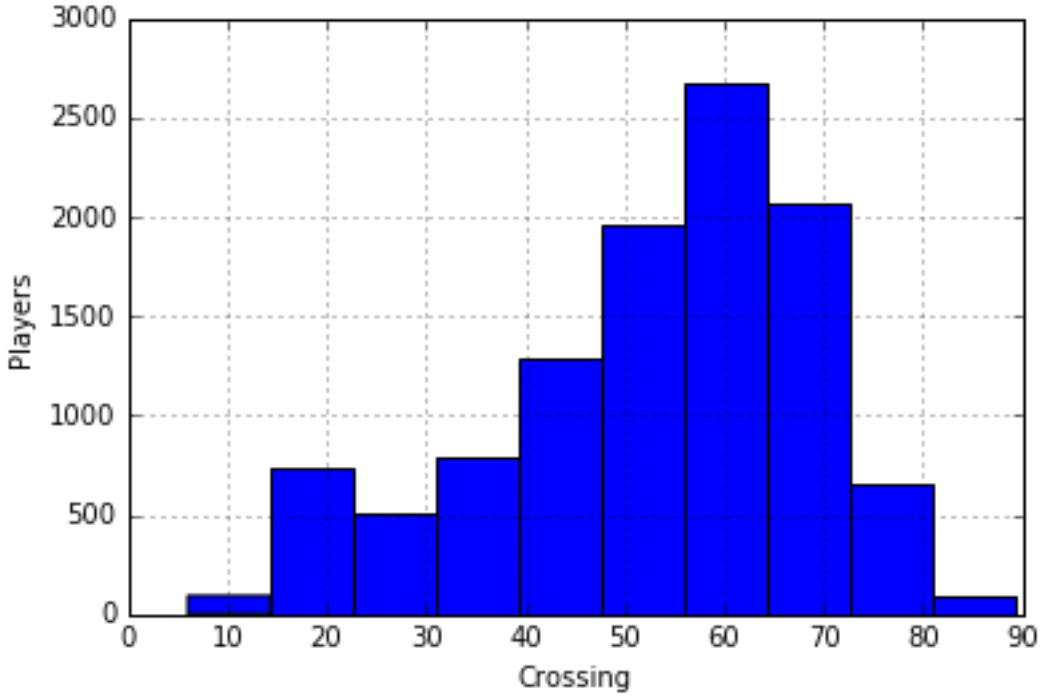
Fig. Histogram showing class distribution (overall rating).

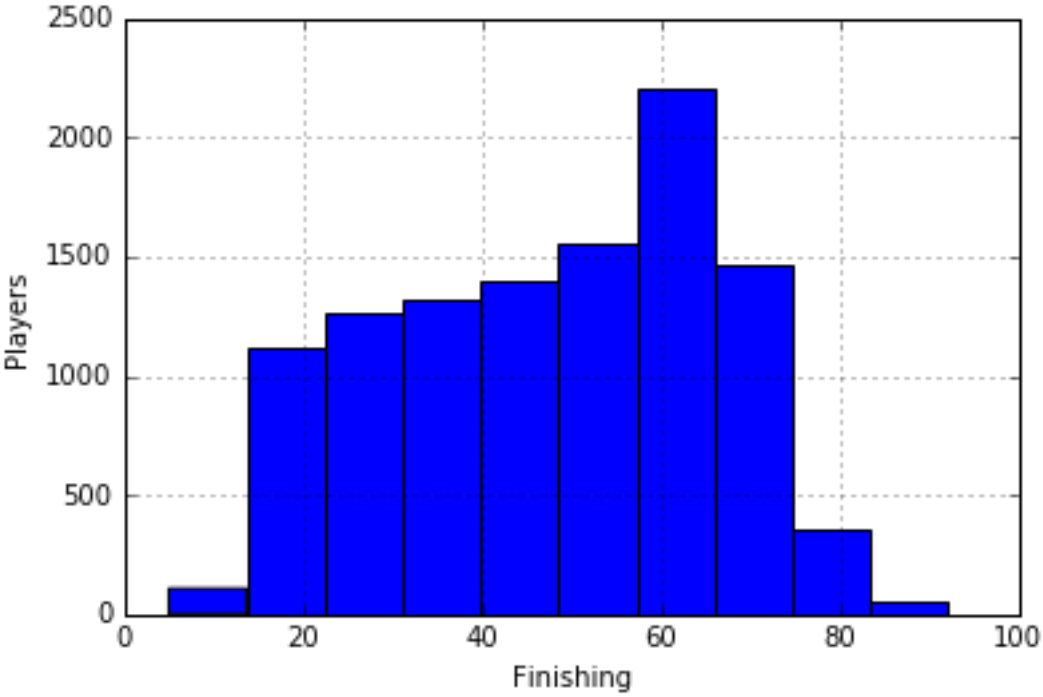
Features

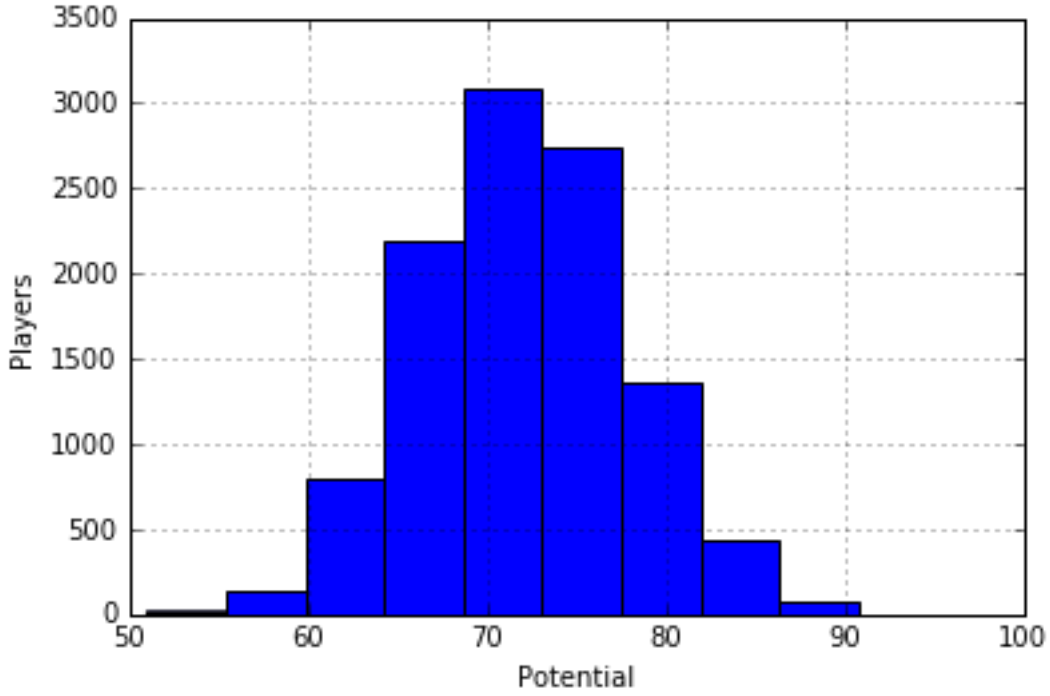
We have 22 features in our data and below are the visualization of some of the top 10 features.

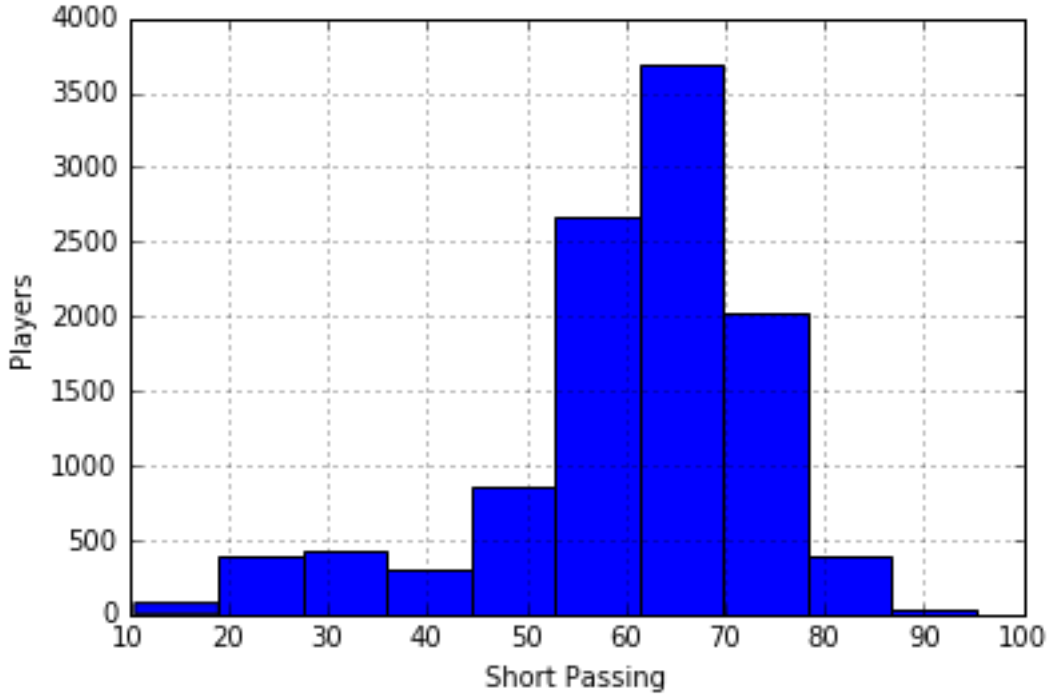


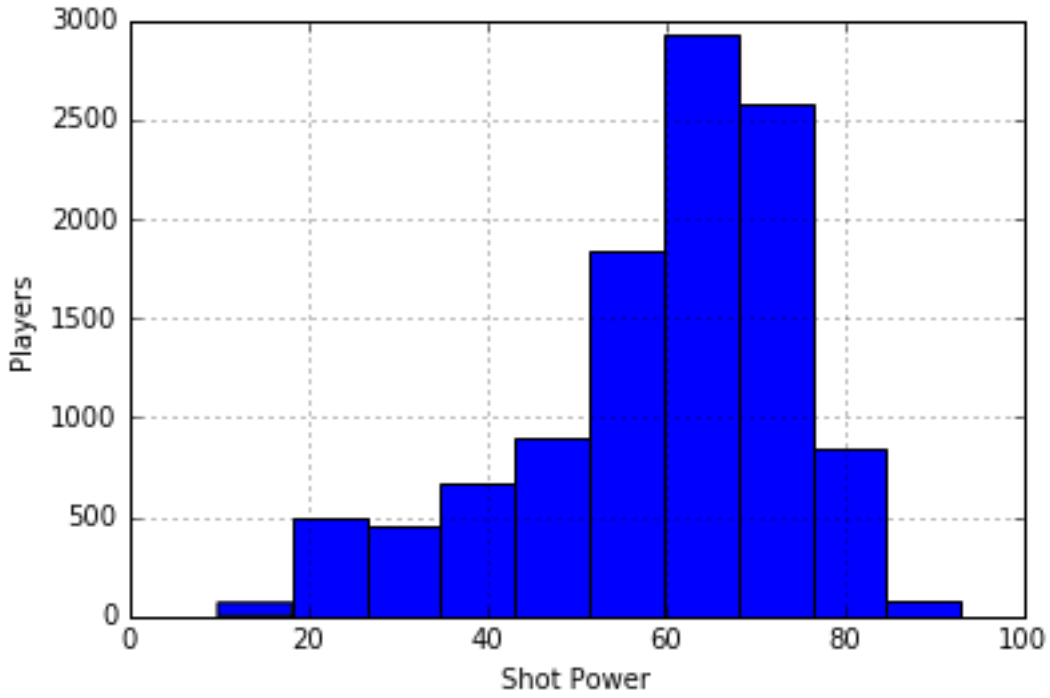


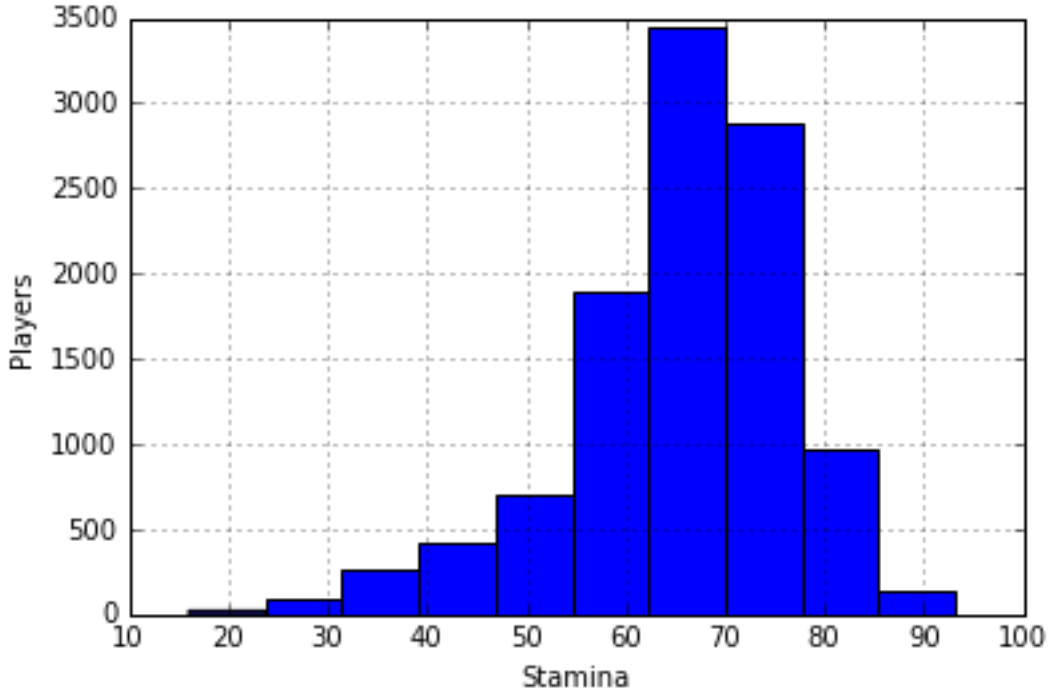

3000
2500
2000
1500
1000
500
0
Players
0
10
20
30
40
50
60
70
80
90
Crossing

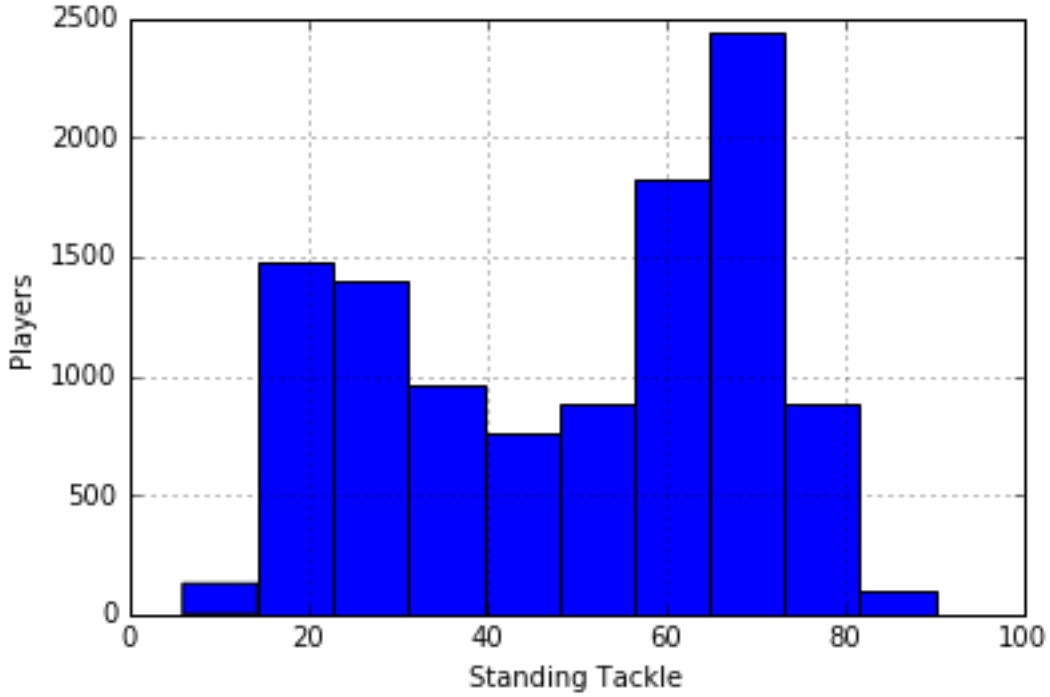

2500
2000
1500
1000
500
0
Players
0
20
40
60
80
100
Finishing

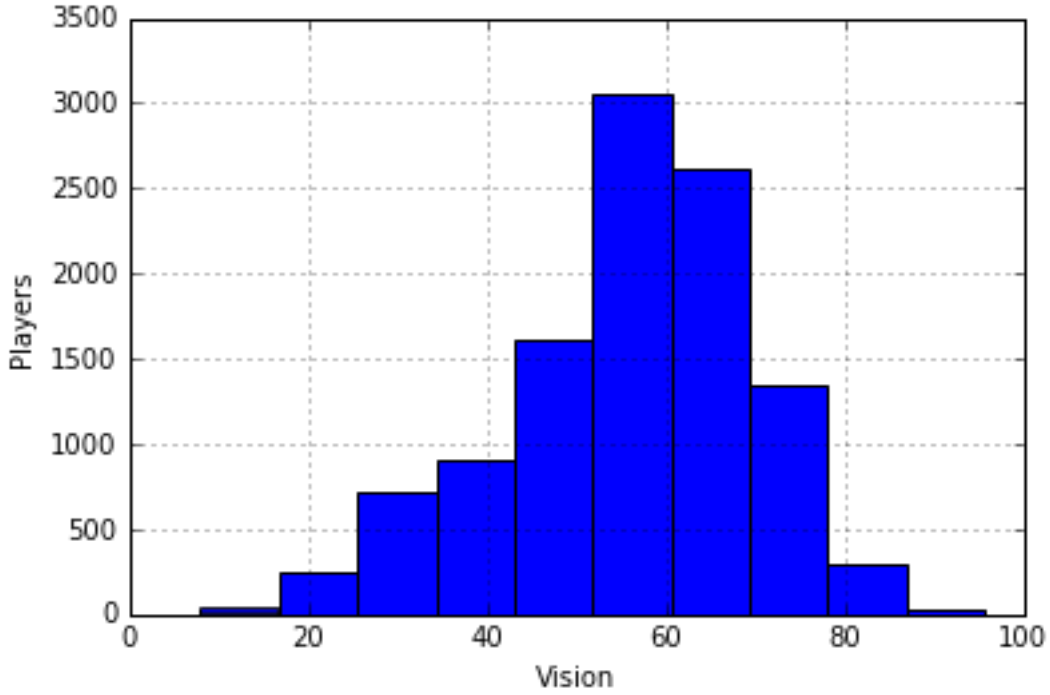

3500
3000
2500
2000
1500
1000
500
0
Players
50
60
70
80
90
100
Potential


4000
3500
3000
2500
2000
1500
1000
500
0
Players
10
20
30
40
50
60
70
80
90
100
Short Passing


3000
2500
2000
1500
1000
500
0
Players
0
20
40
60
80
100
Shot Power


3500
3000
2500
2000
1500
1000
500
0
Players
10
20
30
40
50
60
70
80
90
100
Stamina


2500
2000
1500
1000
500
0
Players
0
20
40
60
80
100
Standing Tackle


3500
3000
2500
2000
1500
1000
500
0
Players
0
20
40
60
80
100
Vision

Evaluation

Performance Measure

Since we were dealing with regression problem we chose the following evaluation metrics:

- cross_validation_score ,
- Accuracy, Mean
- Absolute Error,
- Mean Squared Error.

Classifiers

We chose some of the famous classifiers for regression:

- Linear Regression
- Lasso Regression
- Ridge Regression
- Random forest

Some of the Parameters which brought in a bit more accuracy are as follows:

- Random State- The Random State is useful to get the consistency in the model.
- N_estimators- Since Random Forest uses bagging, if the number of trees is too small the observations will be predicted only once hence we altered the number of trees to check the effect on the estimators.
- Max_depth- The Nodes are expanded until all leaves are pure.
- Fit_intercept- It is simple value at which the fitted line crosses y- axis, it is crucial to include the constant term in the model as it guarantees that our residuals have a mean of zero.

Evaluation Strategy

We did both Test-train split and Cross Validation.

Performance Results

Model	Parameters	Performance
Baseline	Default values	Mean squared error: 29.71
Linear regression	Default values: fit_intercept=True, normalize=False, copy_X=True, n_jobs=1	cross val score: 0.753417 Accuracy: 0.76 mean absoluter error: 2.378649 Mean squared error: 9.42
	fit_intercept=False, normalize=False, copy_X=True, n_jobs=1	cross val score: 0.751949 Accuracy: 0.76 mean absoluter error: 2.388687 Mean squared error: 9.49
Lasso regression	Default values: alpha=0.1, fit_intercept=True, normalize=False, precompute=False, copy_X=True, max_iter=1000, tol=0.0001, warm_start=False, positive=False, random_state=None, selection='cyclic'	cross val score: 0.753396 Accuracy: 0.76 mean absoluter error: 2.378728 Mean squared error: 9.43
	random_state=15	cross val score: 0.748504 Accuracy: 0.75 mean absoluter error: 2.405936 Baseline Mean squared error: 27.17 Mean squared error: 9.72
Ridge regression	Default values alpha=1.0, fit_intercept=True, normalize=False, copy_X=True, max_iter=None, tol=0.001,	cross val score: 0.753417 Accuracy: 0.76 mean absoluter error: 2.378649 Mean squared error: 9.42

	solver='auto', random_state=None	
Random forest	Default values n_estimators=10, criterion='mse', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='auto', max_leaf_nodes=None, min_impurity_split=1e-07, bootstrap=True, oob_score=False, n_jobs=1, random_state=None, verbose=0, warm_start=False	Accuracy: 0.91 cross val score: 0.893719 mean absoluter error: 1.422416 Mean squared error: 3.72
	n_estimators=100, max_depth=25, oob_score=True,random_state=20	Accuracy: 0.92 cross val score: 0.906862 mean absoluter error: 1.324897 Mean squared error: 3.20

Top Features

- Potential
- Ball Control
- Standing Tackle

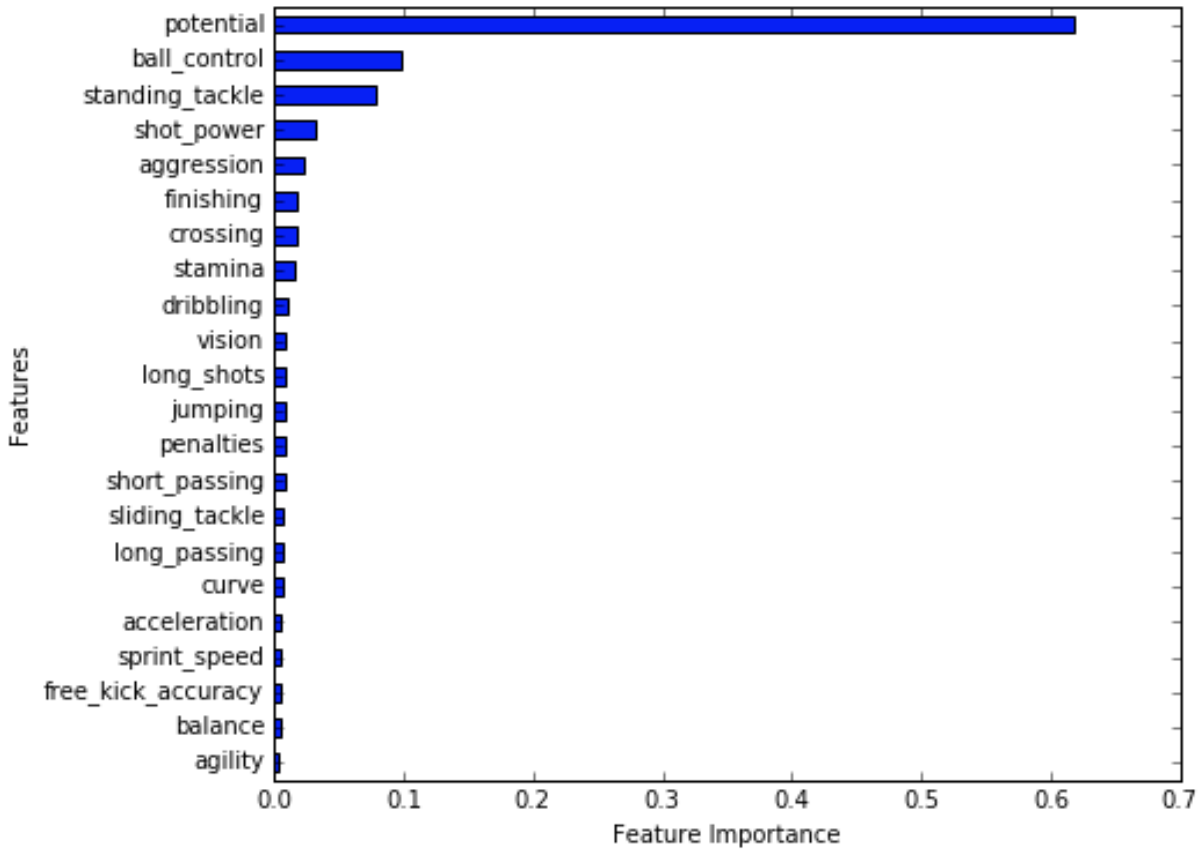


Fig. Top features in our data

Discussion

We started with linear regression as a model to test and train our data set and then with lasso and then ridge and finally with random forest regressor and corresponding results can be seen in the above table under performance results.

As expected, random forest regressor performed well as expected with an accuracy of 92%, which is highest compared among others. Thus, we used random forest regressor to test and train our dataset finally which gave us maximum accuracy.

Interesting/Unexpected Results

Sl No	Input	Output
1	All features as 0.0	48.5
2	All features as 100.0	88.8

- If a new player with no data available is given 0.0 in features the model gives an overall ranking as 48.5
- If a new player with no data available is given 100.0 in features the model gives an overall ranking as 88.8

Contributions of Each Group Member

- Phase1 Data Exploration
  This phase includes collection of data from websites, analysis of the collected data, pre-processing the raw data, handling the missing values.
  All of us collectively did the above mentioned tasks in this phase
- Phase 2: Evaluation
  In this phase, we chose 4 different models to evaluate the performance viz. linear regression, lasso regression, ridge regression and finally random forest.
  Here we divided the task among us-
  - Nithin : linear regression and did baseline evaluation
  - Arati : ridge and lasso regression and mean_abs_error, mean_squared_error
  - Madhu : random forest and cross validation
- Then we all worked to get important features through decision tree regressor and documentation.

Conclusion

Hence from our model given an input as array of values between 0-100 our model will estimate the overall ranking which can be used by many online bidders to choose a player for bidding.

References

1) www.Kaggle.com
2) www.scikit-learn.org