Analysis of NBA Players and Shot Prediction Using Random Forest and XGBoost Models

Maram Shikh Oughali
College of Computer and Information Sciences
Princess Nourah bint Abdulrahman University
Riyadh, Saudi Arabia
435200569@pnu.edu.sa

Mariah Bahloul
College of Computer and Information Sciences
Princess Nourah bint Abdulrahman University
Riyadh, Saudi Arabia
434200172@pnu.edu.sa

Sahar A. El_Rahman
College of Computer and Information Sciences
Princess Nourah bint Abdulrahman University
Riyadh, Saudi Arabia
Faculty of Engineering-Shoubra
Benha University
Cairo, Egypt
sahr_ar@yahoo.com

***Abstract*—This paper proposes a comparative study using machine learning algorithms to predict the shooting success by basketball players in the National Basketball Association (NBA). This work is focusing on analyzing NBA's regular session dataset, which will help NBA teams to prepare their play plan for future games based on the other team players' performance. For instance, how good is each player usually in shooting from different distance and what defense strategies they often use. In this work, Random Forest and XGBoost models are used for shooting prediction.**

Keywords—Random Forest, XGBoost, NBA, prediction, shooting, defenses

I. INTRODUCTION

New technology and statistics will change the way we understand basketball and will add new concepts into on-court play. SportVU is a camera tracking system and is one of the most useful innovations in the NBA. It is a camera system hung from the rafters that collects data at a rate of 25 times per second and follows the ball and every player on the court [1]. This kind of data helps to know more about everything in the court. From the position of each player, to the distance between players and how far they are from the basket, the timing of each and every shot made and the result of it. All of these detailed data and much more are collected from every game and stored in a huge database. And due to the significance of this technology, more and more teams are investing in it. Since this date can be analyzed and lead to a better understanding of the play patterns, moreover, it could reveal the weak and strength points of each player in their team or any other team that they are to play against. In this paper, the authors are highlighting some of the main features to be analyzed, features which may capture the interest of teams and coaches. Moreover, it would benefit them to improve their play plan depending on the team they are facing, to perform their best, and to bring out the best in each player according to his/her weak and strength points.

II. RELATED WORK

In [2], the researchers tackle the topic of predicting shots by using a Convolutional Neural Network (CNN) combined with a Feed Forward Network (FFN) to achieve a final accuracy of 61%. However, it is worth mentioning that the data used as input to build the model in this research paper is images. Although images data is slower to process, and more likely to provide more detailed data about shots, we can see that the accuracy wasn't very high compared to other models using less detailed data. However, the researchers concluded that the data that was used in the layers of their neural network in making predictions was using spatial data such as the location of the ball and the offensive and defensive players, which luckily we have similar attributes in our dataset, as the shot distance and the closest defender distance. While in [3], the researcher was aiming to explore the same shot logs dataset and to build a predictive model that would be useful for teams and coaches. The model he chose for training and testing the data to approach his goal of predicting a shot's success was boosting using XGBoost, and the results he achieved were an accuracy score of 62%, a 61% precision rate and 83% recall. In [4], the dataset used in this analysis is again the same shot logs dataset as the one we are using in our analytical study. The main aim 0f the work was shot success prediction where the maximum accuracy score of 62.2% using Decision Tree model. The model's accuracy was improved by conducting statistical analysis and reducing the number of features in the model, and the overall positive rate was greater than the false positive rate for all thresholds of probabilities that he worked with.

III. METHODOLOGY

The methodology depends on the following phases to predict the shooting result. To analyze the data and implement the models, we used python programing language employing it on Jupyter notebook provided by the Anaconda distribution.

A. Discovery

The National Basketball Association (NBA) is a men's professional basketball league in North America; composed of 30 teams. We decided to analysis the data of NBA because it is widely considered to be the premier men's professional basketball league in the world. In addition to the availability of a big dataset that contains the attributes that concern us in our study. The main goal of our study is predicting shot results, in addition to see the relation between the sot distance and the results.

- **Dataset:** The dataset this paper uses contain a total of 203591 shots from NBA 2014-2015 regular season.

B. Data Preparation

In big data projects, an important step is to prepare the data since the data may be collected with corrupted or missing values that need to be maintained in an appropriate manner. Usually, this phase consumes the longest time in the entire analytic life cycle.

Princess Nourah bint Abdulrahman University.

978-1-5386-8125-1/19/$31.00 ©2019 IEEE

- **Cleaning:**
  - The original dataset included columns that are more than we need in our analysis, such as GAME_ID, MATCHUP, SHOT_NUMBER and CLOSEST_DEFENDER_PLAYER_ID. Hence, we reduced them to minimize the processing time effort and to facilitate our work. In Fig. 1, we can see the descriptive statistics summarizing the columns that matter to us in our analytical study after reducing them and removing the unnecessary columns. The columns that interest our research include FINAL_MARGIN, PERIOD, SHOT_CLOCK, DRIBBLES, TOUCH_TIME, SHOT_DIST, PTS_TYPE, CLOSE_DEF_DIST, FGM and PTS.
  - In TOUCH_TIME column, we notice some outliers, as indicated in Fig. 2, in addition to negative values. We also notice some illegal values exceeding the maximum time limit of 24 seconds, all of those need to be removed.
  - In SHOT_CLOCK column, we notice a big number of null values, we filled those with the mean of the column.
  - Last step we performed was converting the distance measuring unit from feet to meters for better understanding of the results.
- **Visualization**: After the cleaning process we started to visualize our data to have a better understanding of the overall statistics, and to explore the factors that may interfere with shot result whether it is mad or missed.
  - First, a general look at the shot logs data as indicated in Fig. 3-a that represents the percentage of made and missed shots across the entire season, Fig. 3-b shows the distance that all players in general are most likely to shoot from, and Fig. 3-c indicate the relationship between the distance of the shot and whether it was made or not.

	FINAL_MARGIN	PERIOD	SHOT_CLOCK	DRIBBLES	TOUCH_TIME	SHOT_DIST	PTS_TYPE	CLOSE_DEF_DIST	FGM	PTS
count	200590.000000	200590.000000	194997.000000	200590.000000	200590.000000	200590.000000	200590.000000	200590.000000	200590.000000	200590.000000
mean	0.021931	2.498698	12.448271	1.997087	2.720929	13.608178	2.269208	4.136613	0.449948	0.994209
std	13.667853	1.142783	6.755382	3.429842	3.025442	8.919466	0.443549	2.756629	0.497386	1.133479
min	-53.000000	1.000000	0.000000	0.000000	-192.000000	0.000000	2.000000	0.000000	0.000000	0.000000
25%	-9.000000	1.000000	6.200000	0.000000	0.900000	4.700000	2.000000	2.300000	0.000000	0.000000
50%	-1.000000	2.000000	12.300000	1.000000	1.600000	13.600000	2.000000	3.700000	0.000000	0.000000
75%	9.000000	3.000000	16.600000	2.000000	3.600000	22.600000	3.000000	5.300000	1.000000	2.000000
max	53.000000	7.000000	24.000000	32.000000	26.400000	47.400000	3.000000	53.200000	2.000000	6.000000

Fig. 1. Dataset after reducing the columns.

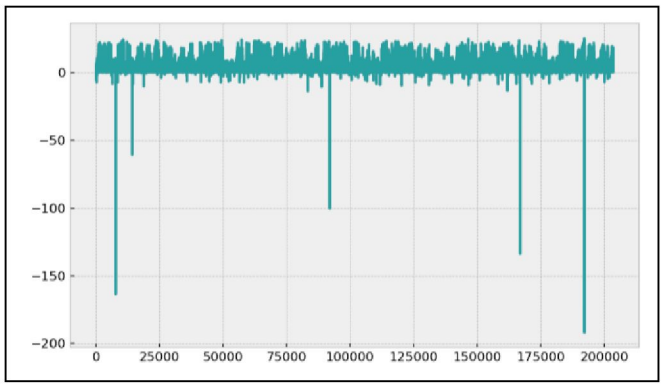
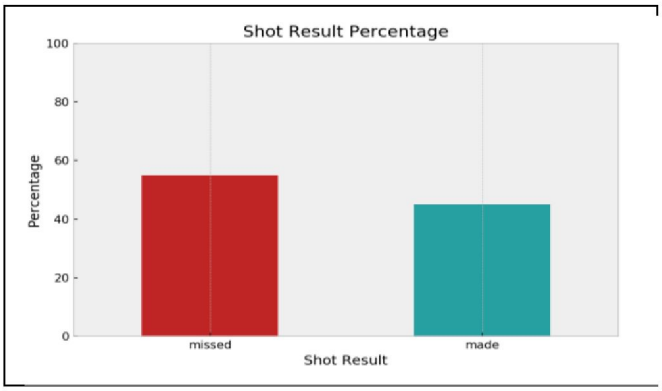


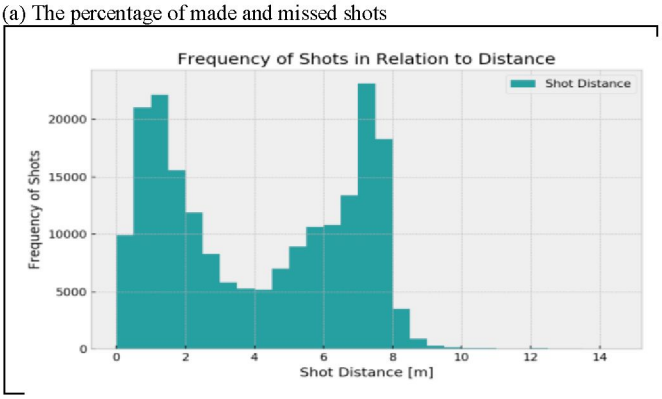
Fig. 2. TOUCH_TIME column outliers.

- Now let's look from the same perspective but at some more specific data. Let us try to see how the distance affects the shot result of the famous players: Stephen Curry (Fig. 4-a), LeBron James (Fig. 4-b), and the difference of free-throw and three-point percentages for both players, Curry and LeBron is indicated in Fig. 4-c.

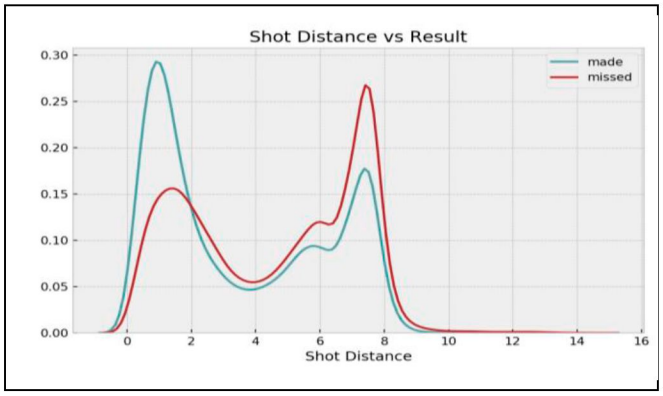
so, we noticed that the defender's distance has an impact on the result of the shot, the further the defender the more possible it is for a shot to be made. In addition, we saw that most successful shots are made at the end of each shot clock round with a percentage of almost 50% of times.



(a) The percentage of made and missed shots



(b) The distance that all players are most shoot from



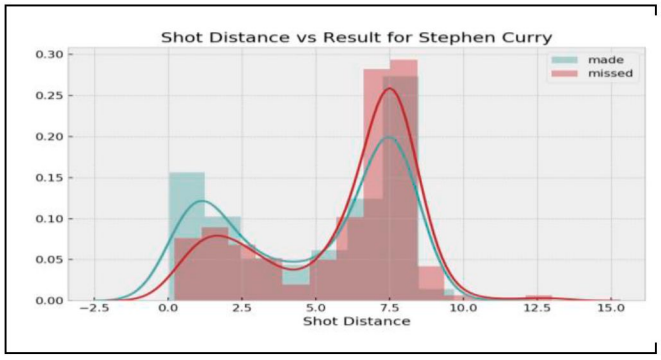
(c) The relationship between the distance & shot result

Fig. 3." A general look at the shot logs data

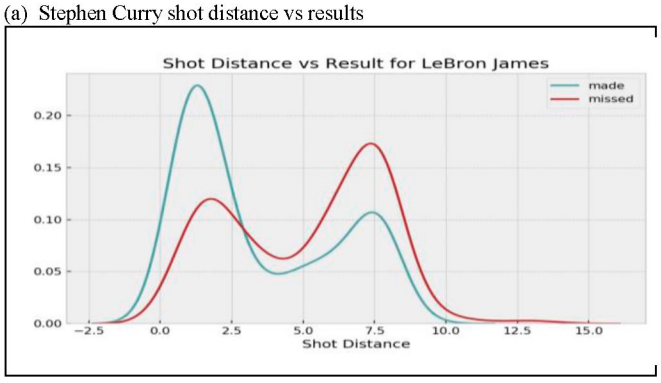
C. *Model Planning and Building*

In these phases, the techniques and workflow that are followed for model scoring are determined, and the dataset is developed for training, testing and production purposes. There is an overlap between model planning and model building phases, wherefore we iterated back and forth between them before reaching the final analytics model.

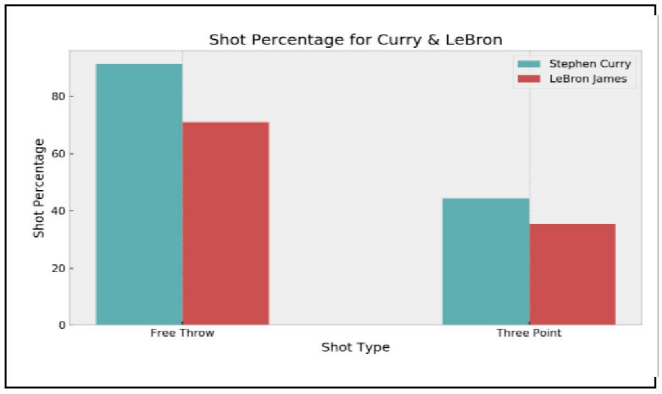
First of all, we explored the factors that may interfere with and effect the shot result if it is Made or Missed. Fig. 5 indicates that the defender's distance has an impact on the result of the shot, where Fig. 5-a shows the further the defender is the more possible it is for a shot to be made and Fig. 5-b shows that most successful shots are made at the end of each shot clock round with a percentage of almost 50% of times. Then, the models (Random Forest and XGBoost) are applied to predict the shot result (Made or Missed) using different attributes that actually have an impact on the shot result.



(a) Stephen Curry shot distance vs results

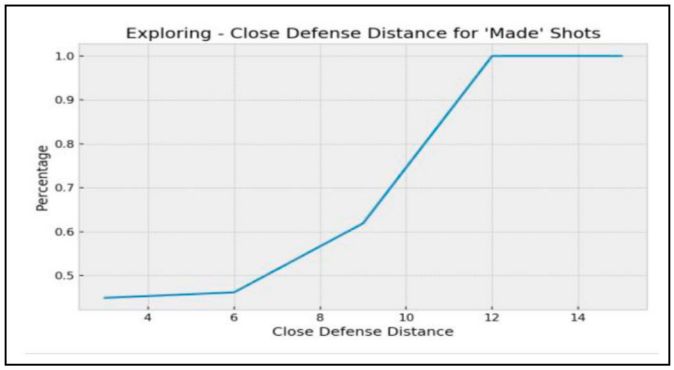


(b) LeBron James shot distance vs results

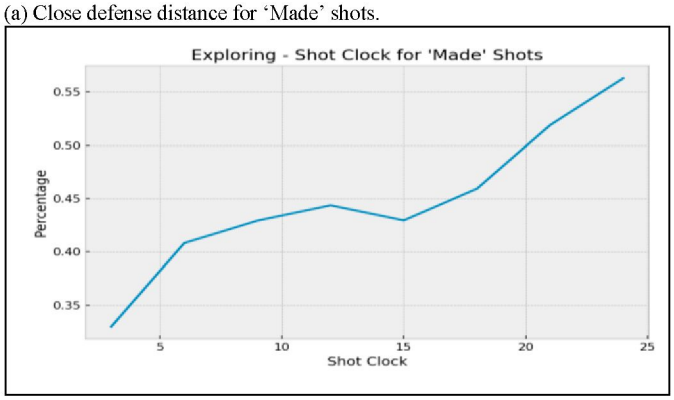


(c) Shot Percentage for Curry & LeBron

Fig. 4." The data visualization after cleaning process



(a) Close defense distance for 'Made' shots.



(b) Shot clock for 'Made' shots.

Fig. 5." Factors that may interfere with and effect the shot result

Random Forest Model

- It is a flexible, easy to use machine learning algorithm that produces, even without hyper-parameter tuning, a great result most of the time. It is also one of the most used algorithms, because of its simplicity and the fact that it can be used for both classification and regression tasks. Random Forest builds multiple decision trees and merges them together to get a more accurate and stable prediction, which makes them robust to over fitting. Another advantage of random forests is that they deal well with categorical data due to the nature of decision trees [5].
- The features used as predictors in our Random Forest model are the shot clock, the number of dribbles, the shot distance, and the closest defender's distance. The shot clock is an integer value that

refers to the exact second the shot was made during the 24 second period that an offense is given to shoot the ball. The dribbles attribute is the number of dribbles made by the shooter before he made the shot. The shot distance is the distance between the offense player and the basket when the shot is made, measured in meters. Finally, the closet defender's distance refers to the distance between the offender and the closest defender at the time of shooting; this distance is measured in meters as well.

- We began by analyzing the data a little more, trying to find which attributes had more impact on the shot result than others. We decided to try to train our first model using a subset of those attributes, so we built our Random Forest model by using the SHOT_CLOCK, DRIBBLES, SHOT_DIST, and CLOSE_DEF_DIST as our predictors, as explained earlier. Where the target was to find the SHOT_RESULT and determine whether it was made or not. In the process of preparing our model we converted the shot result to an integer number, 1 means the shot was made, and 0 for missed shots. The dataset was split to train and test data into 70% training and 30% for testing. After that, we fitted our model, predicted the values in our test set, and checked our model score.

XGBoost Model

- Boosting is a machine learning approach that takes multiple weak learners, which are classifiers that predict a result better than random guessing but not by much and combine them using weights into a strong learner. "The name XGBoost, though, actually refers to the engineering goal to push the limit of computations resources for boosted tree algorithms. Which is the reason why many people use XGBoost." It is a library that implements machine learning algorithms under the Gradient Boosting framework [6]. Training this model, we used the same predictors and the same training and testing sets that we used to train our Random Forest model, which included four features from the original dataset.

- We were tried to look for the most predictive features and studied how these can affect the accuracy of our predictions. We first started by extending our set of predictors to seven attributes. Then, we visualized the importance of each feature using the plot_importance() function provided by the XGBoost library (see Fig. 6). Afterwards, we decided to build our training and testing dataset using the four most predictive features which include: the shot distance, closest defender's distance, the shot clock, and lastly the touch time. The first three features were already explained and used in our previous models. However, a new addition to building our new model is the TOUCH_TIME, which measures the time the shooting player has possession over the ball prior to taking the shot.

- XGBoost (with parameter tuning): Along with using the four most predictive features as explained earlier, the GridSearchCV function from the sklearn toolkit was used as well in building the XGBoost model with parameter tuning, and resulted in better results than the previous models.

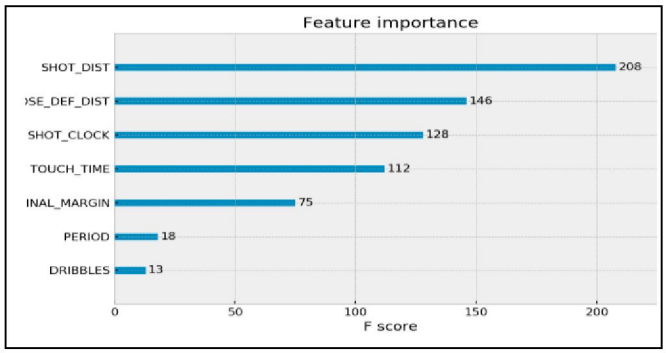


Fig. 6. Most predictive features

IV. Disscution and Results

The Random Forest and XGBoost models were assessed by confusion matrix, process time and accuracy. Random Forest model scored an accuracy of approximately 57% as it can be computed using the confusion matrix as well (see Table I). The accuracy of XGBoost was definitely better, with a score of 62%. Again, this can be computed easily from the confusion matrix by dividing the sum of true negatives and true positives on the total number of predictions (see Table II). Another metric for evaluating the accuracy of the models is plotting the ROC (Receiver Operating Characteristic) Curve and calculating the AUC (Area Under the Curve) for both Random Forest and XGBoost models as indicated in Fig. 7. The model that used XGBoost with parameter tuning achieved the best accuracy rate that approximately 68.4%, which is a very good rate, considering the limits on the available data features as well as the other unpredictable factors that may have a great effect on the shot result. The optimal parameters found when implementing the model were '1' estimator, learning rate '0.00001', max depth '3', and a minimum child node weight of '0.0001'.

Both Random Forest and XGBoost models are machine learning methods that are usually used for predicting (regression or classification) by building individual, simpler trees and combining their results. However, the main difference between those models is the way those trees are built, and how their results are combined. Random Forest uses a random sample of the data, which makes it a more robust model than a single decision tree. Using those random samples, it trains each tree independently. Typically, Random Forest has two parameters, the number of trees and the number of features at each node. On the other hand, the boosting algorithm used by the XGBoost model builds trees sequentially, where with each tree added, it helps correcting the errors in the previous tree. Usually, this model uses three parameters: number, depth of trees and the learning rate. Although in general, Random Forest model is less prone to overfitting and takes less training time compared to XGBoost because of the way each model builds its trees, XGBoost model usually scores better results, there also exist strategies to overcome the overfitting issue by using different combinations of parameters [7].

TABLE I. " CONFUSION MATRIX FOR RANDOM FOREST

n=60,944	Random Forest		
	Predicted P	***Predicted N***	
Actual P	23,703	9,834	33,537
Actual N	16,182	11,225	27,407
	39,885	21,059	

TABLE II. " CONFUSION MATRIX FOR XGBOOST

n=60,944	XGBoost		
	Predicted P	***Predicted N***	
Actual P	28,465	5,072	33,537
Actual N	18,031	9,376	27,407
	46,496	14,448	

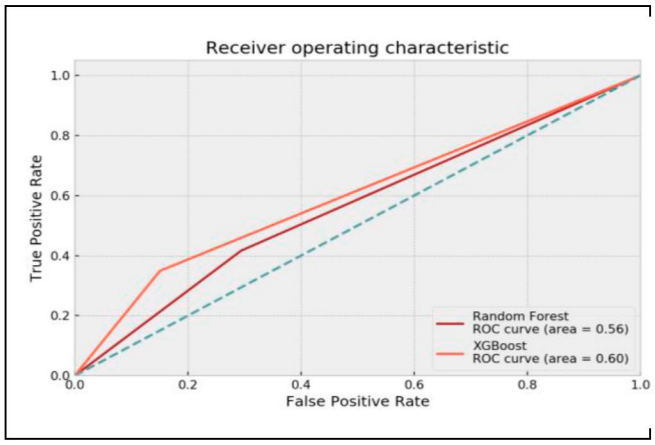


Fig. 7." ROC Curve for Random Forest & XGBoost models

TABLE III. " SHOT PREDICTION ACCURACY

Method	***Accuracy***
Random Forest Model	0.57
XGBoost (without parameter tuning)	0.60
XGBoost (parameter tuning)	0.68

V." CONCLUSION

In this work Random Forest and XGBoost models were used and the best performing model was definitely XGBoost as it scored the highest accuracy rate. However, Random Forest model was a very good and suitable classifier for this kind of dataset as well. Given the challenges when analyzing behavioral data, and the limited features provided by the dataset, an accuracy of approximately 60% is actually still a good result. Achieving 80-90% accuracy seems unrealistic given the given the complexity of the process of shooting, and the many different unmeasured and unpredicted factors that might affect a human when he is attempting to shoot. Be it an emotional factor, being slightly off balance, or simply having a small movement out of place, can affect the outcome of a shot.

ACKNOWLEDGMENT

The author would like to thank all the participants involved into this work at College of Computer and Information Sciences and Princess Nourah bint Abdulrahman University.

REFERENCES

[1]" STATS. (2018). Football Player Tracking | Football Tracking System | STATS. [online] Available at: https://www.stats.com/sportvu-football/ [Accessed 6 Nov. 2018].

[2]" M. Harmon, P. Lucey, and D. Klabjan. Predicting shot making in basketball learnt from adversarial multiagent trajectories 2016

[3]" Kaggle.com. (2018). Shot Analysis & Field Goal prediction - XGBoost | Kaggle. [online] Available at: https://www.kaggle.com/grejsegura/shot-analysis-field-goal-prediction-xgboost [Accessed 23 Nov. 2018].

[4]" GitHub. (2018). achafamo/NBA-Shot-Prediction. [online] Available at: https://github.com/achafamo/NBA-Shot-Prediction/blob/master/Final%20Report.ipynb [Accessed 23 Nov. 2018].

[5]" Towards Data Science. (2018). The Random Forest Algorithm – Towards Data Science. [online] Available at: https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd [Accessed 14 Nov. 2018].

[6]" Xgboost.readthedocs.io. (2018). XGBoost Documentation — xgboost 0.81 documentation. [online] Available at: https://xgboost.readthedocs.io/en/latest/ [Accessed 29 Nov. 2018].

[7]" Quora. (2015). What are the advantages/disadvantages of using Gradient Boosting over Random Forests? [online] Available at: https://www.quora.com/What-are-the-advantages-disadvantages-of-using-Gradient-Boosting-over-Random-Forests [Accessed 11 Feb. 2019]