IJHSR
ijhighschoolresearch.org
RESEARCH ARTICLE

Comparing Three Machine Learning Models That Predict an NFL Player's Position Based on Height and Weight

Mohammad Arif

Queen Elizabeth's School, Queens Rd, Barnet EN5 4DQ; arif1221@yahoo.co.uk

ABSTRACT: This research paper uses machine learning algorithms to investigate the connection between football players' weight, height, and positional assignments in the National Football League (NFL). The paper compares three machine learning algorithms: K-Nearest Neighbors (KNN), Decision trees, and Random Forests. We further investigated the optimal parameters through hyperparameter tuning to compare our machine-learning models. We used performance metrics such as accuracy scores and raw confusion matrices to compare our models. Our results showed that KNN produced the highest accuracy. The findings shed light on the connection between NFL positional assignments and player physical characteristics. The findings can potentially improve team performance and success in the NFL by influencing player selection and development plans.

KEYWORDS: Machine Learning; K-Nearest Neighbor; Decision Trees; Random Forests.

Introduction

The National Football League (NFL) is at the forefront of professional American Football, and the sport's popularity is soaring. In the NFL, technology has been integrated in many aspects, from player tracking to injury rehabilitation and general statistical analysis.[1,3] This research paper delves into the intersection of technology and the NFL. We aim to identify the relationship between an NFL player's position and their height and weight. Beyond this, teams may need a strategic field for specific positions that will play a crucial role, and they will decide where a player will contribute the most effectively. These observations may often be seen and evident; however, using a machine learning (ML) model can help identify the role of specific positions. This may also reveal a genetic component that determines a player's success in a particular position, as their genes determine their height and can significantly impact their weight. KNN is an ML algorithm widely used in data mining and pattern recognition. It uses a proximity principle, where a data point is classified based on its nearest k neighbors. It was developed as a non-parametric alternative to discriminant analysis when probability density estimates are unreliable or difficult to obtain. The KNN algorithm was first introduced in an unpublished report by Fix and Hodges in 1951 and later formalized in 1967 by Cover and Hart, who showed that the classification error rate is bounded by twice the Bayes error rate for k=1 and $n \to \infty$.[4,5] Since then, there have been many developments in KNN classification, including augmented KNN techniques such as boosted fuzzy granule K-nearest neighbor,[6] and smallest modified K-nearest neighbor.[7] Decision trees are another ML algorithm used in classification and regression tasks. They create a hierarchical structure of decisions, creating internal nodes based on features and leaf nodes representing classes. Decision trees date back to the 1960s when they were introduced by J. Ross Quinlain.[8] Following the development of the Iterative Dichotomiser 3 algorithm, decision trees have become popular in various applications such as fraud detection and customer segmentation.[9,10] Random Forest is an ensemble learning algorithm combining multiple decision trees to form an accurate model. Importantly, each decision tree is formed independently before being combined through an average of their predictions. Currently, NFL teams determine a player's position by evaluating physical attributes, which include speed and agility, alongside their skillset and playing set. Random Forest was first introduced by Tin Kam Ho in the 1990s but gained popularity in the 2000s when it was improved.[11] Since then, its uses have been widespread, ranging from healthcare to finance.[12, 13]

Method

We optimized each model for our dataset to accurately compare all three machine learning algorithms.

To do this, we investigated the relationship between the number of neighbors(k) for the KNN algorithm and the accuracy of our ML model.

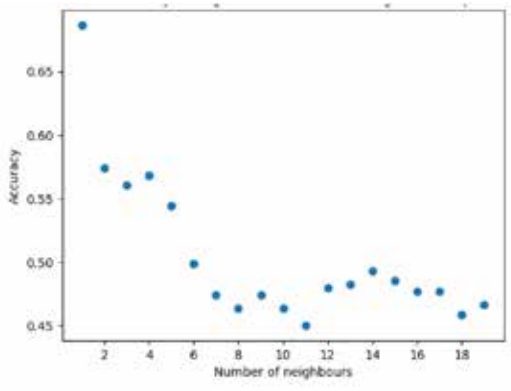

Figure 1: Graph that shows the accuracy of our KNN model as the number of neighbors changes. Here, generally, as the number of neighbors increases, the accuracy decreases.

DOI: 10.36838/v6i11.3

Figure 1 shows that the model's accuracy decreases for our specific dataset as the number of neighbors increases. When k is set to a low value, the algorithm only considers a small number of neighbors to make predictions. This can mean the model is sensitive to noise in the data but risks overfitting. For our data set, we can investigate the reasoning for this outcome: From Figure 2, we can see a graphical representation of all the data. It is evident that many points are close together, so increasing the number of neighbors increases the risk of irrelevant information.

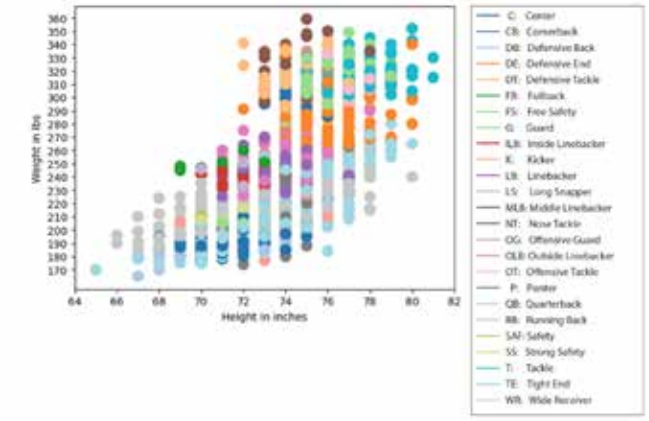

Figure 2: Plot of all players in our dataset on a height and weight axis, with color-coded positions. This shows how certain positions are grouped together and have similar attributes.

When k is set to a high value, a larger number of neighbors are used to make decisions that can reduce the influence of individual points, which are noisy. However, they need to consider neighbors from other classes. It also faces the problem that some classes are smaller than others, which may mean they are less likely to be classified correctly.

We aimed to do the same for the decision tree model, investigating how accuracy changed with max depth.

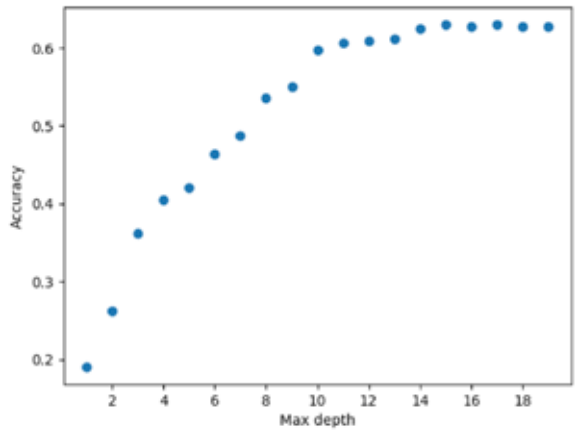

Figure 3: Graph to show how our decision tree model's accuracy changes as max depth changes. As max depth increases, accuracy increases.

The results from Figure 3 indicate that as maximum depth increases, the model's accuracy improves, but only up to a certain point. Beyond this point, increasing the maximum depth does not lead to a significant improvement and may even cause overfitting. As the depth increased, the initial increase in accuracy can be attributed to the decision tree aiming to capture more complex relationships within the data. As the depth increases, the model can create more "branches," allowing it to fit the data better. This leads to improved accuracy and higher performance.

Finally, we investigate the relationship between n_estimators (a parameter for the random forest model) and accuracy.

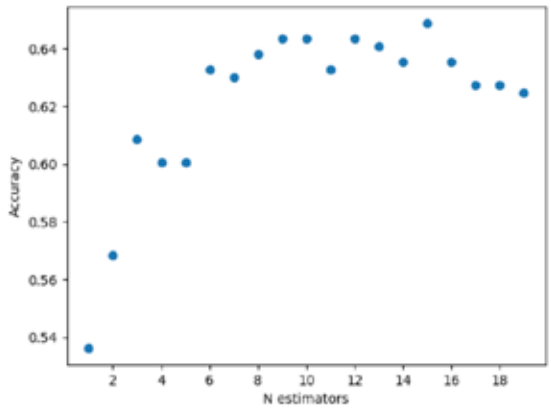

Figure 4: Graph to show how accuracy changes as the n_estimators change for our random forest model. Here, generally, as n estimators increase, accuracy increases.

Figure 4 indicates that the model's accuracy generally increases as the number of estimators increases. However, similarly to that of the decision tree model, a point of diminishing returns can be seen, where further increasing the number of estimators does not lead to any improvement and can instead be seen to decrease accuracy, potentially due to overfitting. Initially, the model performs better as estimators increase, allowing it to capture a broader range of patterns. This is due to the voting mechanism undertaken by random forest, which ultimately improves the model's ability to make predictions.

Data Collection

In this study, we utilized a publicly available data set named "CSV file containing height and weight data on all the players on 2013 NFL teams" obtained from GitHub.[14] This dataset is a collection of all players on the 2013 NFL teams, stating their position, height, weight, date, and team. Here, we extracted three columns: Position, height, and weight.

The data was randomly sampled without replacement. This was done to ensure no data points were duplicated across the training and testing set.

Software used:

We used Python to build and train this ML model. The scikit-learn (sklearn) library was used due to its extensive data processing and evaluation tools. The matplotlib library was also utilized to illustrate the many diagrams and make the results more interpretable.

Data split:

To compare the three machine learning models accurately, we aimed to keep the test and train split the same for all three at 80/20. This meant the model would learn on 80% of the data while would be tested on the remaining 20%.

DOI: 10.36838/v6i11.3

Model Training:

KNN, Decision tree, and random forests were trained on the training set. They were each implemented using their specific sklearn method.

After training the methods, the performance of each model was evaluated using the test set. Various evaluation methods were used to compare the ML methods to assess the model's performance. This involved using confusion matrices as well as accuracy scores. Accuracy scores provide an overall measure of the model's accuracy. At the same time, the confusion matrix allows for more detailed information stating the number of true negatives, true positives, false negatives, and false positives, and it is also used to calculate the accuracy.

Results and Discussion

Applying three machine learning algorithms has yielded insightful results in identifying trends between players' physical attributes (height and weight) and their positions in the NFL.

It is also important to consider the accuracy of a random model. With 25 classes, we can estimate the probability of random success to be 4%. It is important to understand the effectiveness of the machine learning models, as all three methods produce accuracies far greater than this performance metric.

In our comparison, we utilized two averages: macro and weighted average. The macro average can be defined as aggregating the performance metrics recall, precision, and f-1 score and taking their average. It provides equal importance to each class. Conversely, the weighted average assigns weights to each class before averaging the performance metrics. It does this by weighting each class proportionately to its support (frequency in the dataset).

KNN:

We used Python to build and train this ML model. The scikit-learn (sklearn) library was used due to its extensive data processing and evaluation

Table 1: Accuracy metrics for all positions when using the KNN model. This shows an overall accuracy of 69%.

Accuracy	0.6863270777479			
	Precision	Recall	F1-Score	Support
C	0.55	0.40	0.46	15
CB	0.77	0.69	0.73	29
DB	0.64	0.60	0.62	15
DE	0.75	0.80	0.77	30
DT	0.50	0.74	0.60	19
FB	1.00	0.71	0.83	7
FS	0.82	0.64	0.72	14
G	0.71	0.45	0.56	22
ILB	0.67	0.57	0.62	7
K	0.64	0.78	0.70	9
LB	0.50	0.64	0.56	14
LS	0.83	0.83	0.83	6
MLB	0.62	1.00	0.77	5
NT	1.00	0.40	0.57	5
OG	0.00	0.00	0.00	1
OLB	0.79	0.79	0.79	28
OT	0.33	0.33	0.33	3
P	0.40	0.33	0.36	6
QB	0.67	0.71	0.69	17
RB	0.61	0.83	0.70	23
SS	0.60	0.60	0.60	10
T	0.58	0.68	0.62	22
TE	1.00	0.96	0.98	25
WR	0.74	0.63	0.68	41
Accuracy			0.69	373
Macro Avg	0.65	0.63	0.63	373
Weighted Avg	0.70	0.69	0.69	373

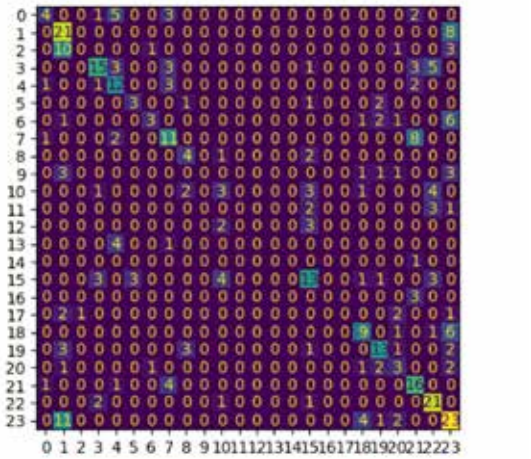

Figure 5: Confusion Matrix for KNN model. This correlates to the results in Table 1.

As shown by Table 1, the model's overall accuracy was 68.63% when we set the number of neighbors to 1. This indicated a moderately successful performance in accurately classifying the test data. Further analyzing the precision, recall, and F1-score (which used Figure 5) for each position allows a deeper understanding of the model's performance. Positions such as fullback (FB), tight end (TE), and middle linebacker (MLB) show a high precision, recall, and F1- score. This may suggest that a player's height and weight play a significant role in these positions. These players often engage in high-contact situations, where weight and height will offer the player an advantage in strength and reach. The clustering of these positions tells us that ML models can successfully use physical features (height and weight) to predict the general field location of a player.

On the other hand, positions such as offensive tackle (OT) and offensive guard (OG) present low scores in the three metrics, suggesting that additional factors may be of greater importance to their role in their position. Further investigation into the roles of OGs and OTs could aid the conclusion. We have also noticed that higher support positions generally have higher evaluation metrics. This shows how machine learning algorithms perform better with more input and are more accurate on average. Both macro average and weighted average provide aggregated metrics across the positions. The macro average was calculated as 0.63, indicating a decently good performance. The weighted average is 0.69, showing that positions with greater instances have a larger influence.

These results highlight the complexities of assigning players to positions based solely on height and weight. While these physical attributes play a role in determining a player's position, other factors, such as skill set and playing style, may also contribute. Integrating additional features such as speed or agility could enhance the model's performance.

Decision tree:

Table 2: Accuracy metrics for all positions when using the decision tree model. The overall accuracy is 63%.

Accuracy	0.6273458445040			
	Precision	Recall	F1-Score	Support
C	0.52	0.80	0.63	15
CB	0.52	0.90	0.66	29
DB	0.57	0.53	0.55	15
DE	0.74	0.77	0.75	30
DT	0.57	0.68	0.62	19
FB	0.67	0.86	0.75	7
FS	0.67	0.57	0.62	14
G	0.55	0.55	0.55	22
ILB	0.31	0.71	0.43	7
K	0.67	0.44	0.53	9
LB	0.46	0.79	0.58	14
LS	0.67	0.67	0.67	6
MLB	0.00	0.00	0.00	5
NT	1.00	0.60	0.75	5
OG	0.00	0.00	0.00	1
OLB	0.88	0.50	0.64	28
OT	0.50	0.33	0.40	3
P	0.29	0.33	0.31	6
QB	0.67	0.71	0.69	17
RB	0.94	0.65	0.77	23
SS	0.29	0.40	0.33	10
T	0.87	0.59	0.70	22
TE	1.00	0.88	0.94	25
WR	0.74	0.39	0.51	41
Accuracy			0.63	373
Macro Avg	0.59	0.57	0.56	373
Weighted Avg	0.67	0.63	0.63	373

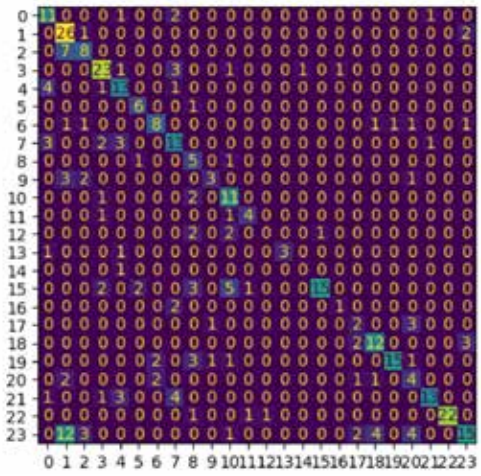

Figure 6: Confusion Matrix for our decision trees model. This correlates to the results in Table 2.

Using decision trees, the machine learning model had an overall accuracy of 62.73%, as seen in Table 2, indicating a moderately successful performance. We set our model's max depth to 14 to achieve this accuracy.

An accurate evaluation of a model's performance capabilities is vital to derive valuable insights into the player position's specific requirements. Precision, recall, and F1-score measurements (calculated from Figure 6) provide excellent metrics with which to assess individual-level performance effectiveness across different player categories such as Fullback (FB), Defensive Tackle (DT), and Nose Tackle (NT). Our study showed that these groups recorded robust scores across all three indicators—implying highly precise classifications based solely on weight and height attributes. Similarly, within these three positions, we see a general position in the backline requiring a greater defensive job. For this reason, the ML has grouped these positions into a similar cluster regarding accuracy.

However, Middle Linebacker (MLB), Offensive Guard (OG), and Offensive Tackle (OT) displayed lower scores across the same three factors, highlighting greater complexity around assigning player roles in these particular categories. This could be attributed to the lower support, as the model has less data to train on and, therefore, is more likely to be less accurate.

It is plausible that technical playing skills or tactical elements play more significant roles in positioning players within these groups rather than merely relying on their biological traits alone. Improving our existing model for enhanced accuracy with respect to such complex situations will likely require incorporating additional parameters or domain-specific knowledge. Recognizing frequent instances' reliability in evaluating outcomes is vital- this explains how Cornerback (CB), Defensive End (DE), and Wide Receiver (WR) outperformed other groups across precision, recall, and F1-scoring mechanisms, making it evident that effective classification was done using solely height/weight qualities. With each position carrying an equal weighting, the macro-average score approximates around 0.56, revealing moderate performance among different positions. In contrast, the weight-based average score is slightly higher at around 0.63, reflecting greater accuracy from larger classes.

Random Forests:

Table 3: Accuracy metrics for all positions when using the random forests model. The overall accuracy is 65%.

Accuracy	0.6541554959785			
	Precision	Recall	F1-Score	Support
C	0.67	0.40	0.50	15
CB	0.57	0.83	0.68	29
DB	0.64	0.47	0.54	15
DE	0.77	0.77	0.77	30
DT	0.52	0.68	0.59	19
FB	0.83	0.71	0.77	7
FS	0.67	0.43	0.52	14
G	0.56	0.64	0.60	22
ILB	0.40	0.57	0.47	7
K	0.67	0.44	0.53	9
LB	0.56	0.64	0.60	14
LS	1.00	0.50	0.67	6
MLB	0.50	0.20	0.29	5
NT	1.00	0.80	0.89	5
OG	0.00	0.00	0.00	1
OLB	0.83	0.71	0.77	28
OT	0.00	0.00	0.00	3
P	0.25	0.17	0.20	6
QB	0.75	0.71	0.73	17
RB	0.74	0.87	0.80	23
SS	0.33	0.60	0.43	10
T	0.65	0.68	0.67	22
TE	0.86	0.96	0.91	25
WR	0.72	0.56	0.63	41
Accuracy			0.65	373
Macro Avg	0.60	0.56	0.56	373
Weighted Avg	0.67	0.65	0.65	373

DOI: 10.36838/v6i11.3

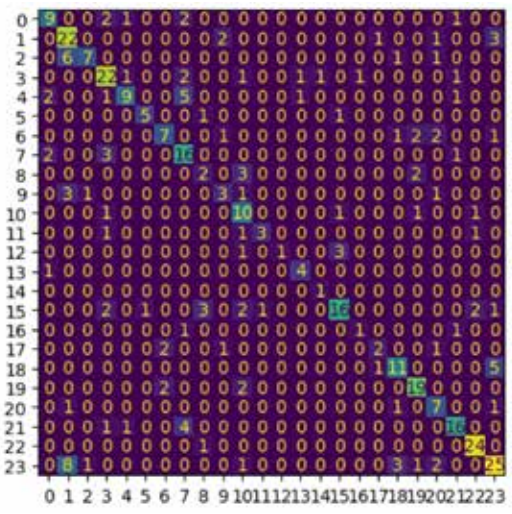

Figure 7: Confusion Matrix for our random forests model. This correlates to the results in Table 3.

When using random forests, our machine learning model's performance is given an overall accuracy of 65.42% (Table 3), indicating a moderate level of classification performance. This accuracy was achieved using an n_estimator value of 15. Evaluating the model's performance for specific positions requires examining its precision, recall, and F1-score (from Figure 7). Our study found favorable results for Fullback (FB), Defensive End (DE), and Nose Tackle (NT) with regard to high precision scores, along with excellent recall and F1 scores, suggesting accurate player classification based on weight and height attributes. Therefore, physical attributes are essential in differentiating between players in these positions. It is interesting to note that the positions were similar in the case of the decision trees, which shows how related the positions are to their physical attributes. In contrast, the Offensive Guard (OG), Offensive Tackle (OT), and Middle Linebacker (MLB) showed lower precision scores, implying challenges in precisely predicting specific player positions relying on only weight and height; however, their respective recall had stated frequent inconsistencies when attempting classification resulting in low F-1 score. We can acknowledge various factors affecting these, such as skillset playing style or positional needs, which may significantly affect positional assignments among these categories.

The macro average is around 0.56, indicating a moderate overall performance across different positions. The weighted average is slightly higher, around 0.65, reflecting the influence of positions with a larger number of instances.

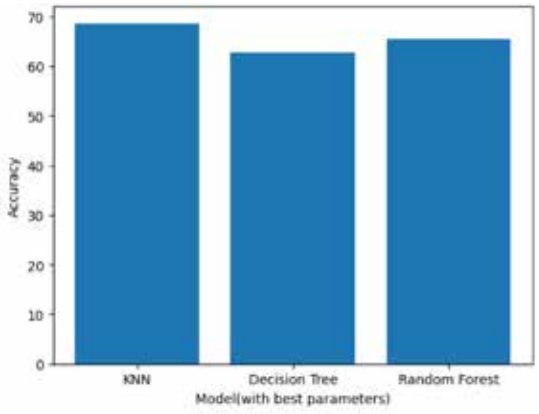

Figure 8: Bar chart showing our three models and their accuracies. KNN has the highest overall accuracy, followed by random forest and decision tree.

Among the three machine learning algorithms compared, the highest overall accuracy was achieved by KNN at 68.63%, as seen in Figure 8. However, all three models faced similar challenges in accurately predicting certain player positions based solely on height and weight attributes. Positions such as fullback (FB) showed consistent success across all models, indicating that these positions can be reasonably determined using these two factors alone. On the other hand, positions like offensive guard (OG), offensive tackle (OT), and middle linebacker (MLB) proved difficult to predict accurately, suggesting the need for additional parameters. This may be for multiple reasons, such as having low support (under 10), so a larger dataset should be used to reach a more accurate conclusion. Another reason may be that the role of these players is less reliant on height and weight. For example, the qualities of agility, speed, and a high football IQ may be more important as they allow the player to make quicker decisions that will result in a better outcome. While each model had its strengths and weaknesses for specific positions, it is evident that relying solely on height and weight presents limitations in assigning players to positions in the NFL.

We also notice that certain positions perform differently across the different models. For example, the Strong safety is predicted moderately well in KNN (0.60 precision score) while performing much worse in the decision trees and random forests model (0.29 and 0.33 precision score, respectively). As all models were trained and tested on the same data set, the disparity would occur due to the ML method. This may suggest that the height and weight of all the strong safeties are close together.

Conclusion

In conclusion, each model performs at moderately high accuracy and performs similarly across most positions. Relying solely on height and weight presents limitations when assigning players to positions in the NFL. We can also not conclusively say that one model is outright better than the other, as they produced similar metric scores across the board, albeit KNN came out on top for this specific dataset.

Our models performed moderately well in positions where physical traits play a central role and less so in positions where agility, speed, and tactical skill may be more critical. Grouping

the highest predicted positions, the ML models performed well with positions located in the backline. These positions seem to be more reliant on height and weight.

References

1. Dutta, R.; Yurko, R.; Ventura, S. L. Unsupervised Methods for Identifying Pass Coverage among Defensive Backs with NFL Player Tracking Data. *Journal of Quantitative Analysis in Sports* **2020**, *16* (2), 143–161. https://doi.org/10.1515/jqas-2020-0017.
2. Görtz, S.; Matava, M. The University of the National Football League: How Technology, Injury Surveillance, and Health Care Have Improved the Safety of America's Game. *Journal of Knee Surgery* **2016**, *29* (05), 370–378. https://doi.org/10.1055/s-0036-1584313.
3. Stimel, D. A Statistical Analysis of NFL Quarterback Rating Variables. *Journal of Quantitative Analysis in Sports* **2009**, *5* (2). https://doi.org/10.2202/1559-0410.1166.
4. Silverman, B. W.; Jones, M. C. E. Fix and J.L. Hodges (1951): An Important Contribution to Nonparametric Discriminant Analysis and Density Estimation: Commentary on Fix and Hodges (1951). *International Statistical Review / Revue Internationale de Statistique* **1989**, *57* (3), 233–238. https:// doi.org/10.2307/1403796.
5. Cover, T.; Hart, P. Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory* **1967**, *13* (1), 21–27. https://doi.org/10.1109/tit.1967.1053964.
6. Li, W.; Chen, Y.; Song, Y. Boosted K-Nearest Neighbor Classifiers Based on Fuzzy Granules. *Knowledge-Based Systems* **2020**, *195*, 105606. https://doi.org/10.1016/j.knosys.2020.105606.
7. Ayyad, S. M.; Saleh, A. I.; Labib, L. M. Gene Expression Cancer Classification Using Modified K- Nearest Neighbors Technique. *Biosystems* **2019**, *176*, 41–51. https://doi.org/10.1016/ j.biosystems.2018.12.009.
8. Quinlan, J. R. Induction of Decision Trees. *Machine Learning* **1986**, *1* (1), 81–106. https://doi.org/ 10.1007/bf00116251.
9. Choi, E.; Go, W.; Lee, T. *A STUDY on FINANCIAL FRAUD DETECTION METHOD USING MACHINE LEARNING in MOBILE SMALL- AMOUNT PAYMENT SERVICE*; pp. 2321–1156. https://www.ijirts.org/ volume4issue5/IJIRTSV4I5004.pdf (accessed 2023-06-11).
10. Duncan, R. What Is the Right Organization Structure? Decision Tree Analysis Provides the Answer. *Organizational Dynamics* **1979**, *7* (3), 59–80. https://doi.org/10.1016/0090-2616(79)90027-5.
11. Tin Kam Ho. *Random decision forests*. IEEE Xplore. https://doi.org/10.1109/ICDAR.1995.598994.
12. Khalilia, M.; Chakraborty, S.; Popescu, M. Predicting Disease Risks from Highly Imbalanced Data Using Random Forest. *BMC Medical Informatics and Decision Making* **2011**, *11* (1). https://doi.org/ 10.1186/1472-6947-11-51.
13. Zou, Z. B.; Peng, H.; Luo, L. K. The Application of Random Forest in Finance. *Applied Mechanics and Materials* **2015**, *740*, 947–951. https://doi.org/10.4028/www.scientific.net/AMM.740.947.
14. Craig Booth. *CSV file containing height and weight data on all the players on 2013 NFL teams*. Github. https://gist.github.com/craigmbooth/5a9be04fe72d77fa3cff.

Author

Mohammad Arif is interested in computer science and math and how they link together to real-life applications. He also has interests in sports, competing in gymnastics, water polo, and swimming.

DOI: 10.36838/v6i11.3