International Journal of Decision Support System Technology
Volume 14 • Issue 1

Machine Learning on Soccer Player Positions

Umberto Di Giacomo, Università degli Studi del Molise, Italy*
Francesco Mercaldo, Università degli Studi del Molise, Italy
Antonella Santone, Università degli Studi del Molise, Italy
Giovanni Capobianco, Università degli Studi del Molise, Italy
https://orcid.org/0000-0002-9888-3627

ABSTRACT

During the last few years, sports analytics has been growing rapidly. The main usage of this discipline is the prediction of soccer match results, even if it can be applied with interesting results in different areas, such as analysis based on the player position information. In this paper, the authors propose an approach aimed to recognize the player position in a soccer match, predicting the specific zone in which the player is located in a specific moment. Similar objectives have not yet been considered. The authors consider supervised machine learning techniques by considering a dataset obtained through video capturing and tracking system. The data analyzed refer to several professional soccer games captured at the Alfheim Stadium in Tromso, Norway. The approach can be used in real time in order to verify if a player is playing according to the guidelines of the coach. In the experimental analysis, three different types of classification have been performed (i.e., three different divisions of the field), reaching the best results with random tree algorithm.

KEYWORDS

Machine Learning, Performance Analysis, Soccer Analytics, Sport Analytics

1. INTRODUCTION AND RELATED WORK

Sport analytics refers to the use of data and advanced statistics, for example machine learning techniques, to measure performance with the aim to take informed decisions and gain a competitive sports advantage. In other words, sport analytics is the practice of applying mathematical and statistical principles to different sports, such as baseball (Dietrich et al., 2014), basketball (Jain and Kaur,2017) and hockey (Liu and Schulte,2018). In soccer, the most usage is about the prediction of results and the definition of strategies that can be used to win a game or to obtain an improvement of the team performances. Usually, the models constructed in these analysis are based on several aspects about the game, such as tactical, technical or physical information.

However, although each sport has its own characteristics, sport analytics uses the same basic methods and approaches as any other kind of data analysis and, when properly applied, can yield tremendous competitive advantages to a team or an individual player.

The analysis that can be performed with sport analytics is typically divided into two different parts: bio-mechanical and notational analysis (Hughes and Franks,2004). Both techniques involve the analysis and improvement of the sport performance giving good feedbacks to coaches and athletes.

DOI: 10.4018/IJDSST.286678
*Corresponding Author

Sport biomechanics is concerned with fine detail about individual sport techniques while, on the other hand, notational analysis is more concerned with gross movements or movement patterns in games or teams, and is primarily interested in strategy and tactics. These types of analysis are useful because, if we consider the well- chosen performance indicators to evaluate a specific game, it can be possible to highlight advantageous strategy or important aspect of team performance. In other words, they help coaches to identify good and bad performances of team member or the whole team. (Bartlett,2001).

As we have said before, one of the most widespread use of sport analytics, in soccer environment, is related to the prediction of soccer match results. In literature there are several works focused about the most important factors that influence the results of a game. In (Capobianco et al.,2019), the authors propose a new feature set aimed to model a soccer match. The set is related to characteristics obtainable not only at the end of the match, and it is used to predict the results of the match and the number of goal

scored by the team that won the game. In (Joseph et al.,2006), an approach based on Bayesian Networks to predict match results has been presented. The analysis showed that the Bayesian networks is generally superior to other techniques such as the MC4, a decision tree learner, naive Bayesian learner (NB), and k-nearest neighbor learner (KNN) for this domain in terms of predictive accuracy. Specifically, authors obtain an accuracy equal to 59% which outperformed other machine learning models i.e., 41.7% (obtained by the MC4 classification algorithm), 47.86% (with the NB algorithm) and 50.58% (with the KNN algorithm). A similar analysis has been proposed in (Liti et al.,2017), where the authors predict the outcome of soccer matches finished with a draw at the end of the first half using the information stored during the first part of the match; while, in (Razali et al.,2017) a Bayesian Network approach to predict the outcome of English Premier League matches has been constructed. In (Berrar and Dubitsky,2019) the authors suggest that a key factor in soccer match outcome prediction lies in the successful incorporation of domain knowledge into the machine learning modeling process.

In soccer, there are other types of work concerning specific aspects of the game or player performance analysis. For example, in (Fernandez and Cervone,2019) it has been presented a model that quantifies the expected outcome of a soccer possession at any time during the possession, driven by a fine-grained evaluation of the full spatio-temporal characteristics of the 22 players and the ball. In (Kharrat et al.,2017) the authors try to examine who are the best players in European football, and demonstrate how the players' ratings evolve over time, using plus-minus rating. Finally, in (Schultze and Wellbrock,2017) it has been proposed a weighted plus/minus metric to be used as an instrument to evaluate player performance.

In our work, we propose an approach to predict the player positions in a soccer match that can be used to verify, also in real-time, if a specific player observes the guidelines given by the coach. Additionally, this method can be used after the match, to analyze the behavior of the team and make considerations on several aspects to improve performances during the training; or to analyze the next opponent team in order to get some kind of information that can be used to obtain a strategical advantage before the match. Similar objectives have never been considered yet with our best knowledge. In this method, we

exploit supervised machine learning techniques by considering several classification algorithms to enforce the conclusion validity. In detail, the proposed method exploits features related to the relative positions of the ball in *x* and *y* axis, other features as, for instance, the player speed.

The paper proceeds as follows: Section 2 contains the proposed methodology for soccer player position detection; Section 3 presents the experimental results; in Section 4 conclusions and future research directions are presented.

2. METHOD

We propose a methodology to identify the zone in which a specific player is located, starting from other types of information such as the position of the ball and the direction toward the player are moving and looking at. It is important to know and describe the position of a player in terms of the ball position and body orientation in order to construct a model that can be used to predict, in each specific moment and tactical situation, in which zone of the field a player should be located in. Additionally, with this approach, when there is a negative event for a specific team, such as, for example, a goal conceded, the constructed model can be used to verify if a specific player or group of players were in a wrong position, respect than the coach instructions. The important point of this methodology is that it can be applied also to real-time data, extracted from a specific time window during a game, in order to verify, for example, if a player or group of them observe the coach guidelines.

Specifically, the method that we propose is depicted in Figure 1 and it is characterized by 4 different steps:

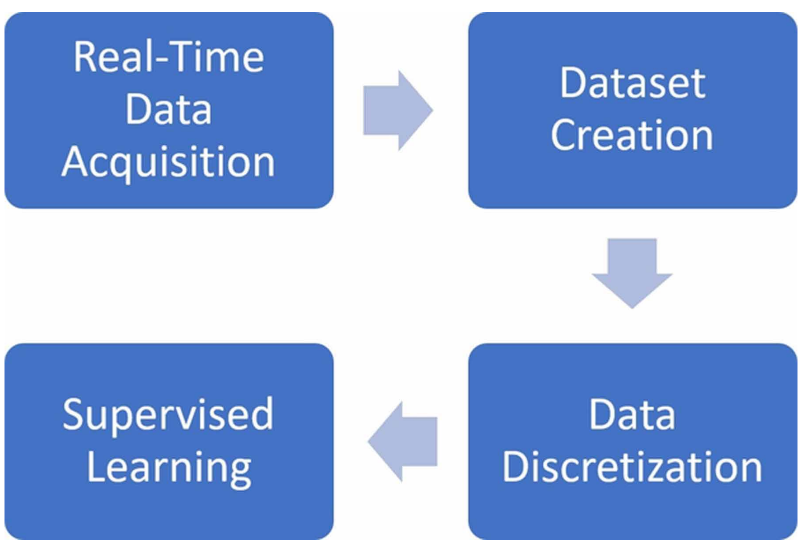
- Real-Time Data Acquisition
- Dataset Creation
- Data Discretization
- Supervised Learning

2.1 Real-Time Data Acquisition

The first step is about the data collection. The proposed methodology can consider, as source of information, video capturing method and advanced tracking system, like GPS system. These technologies allow to analyze real-time data, acquired, for example, in a specific time window - every 15 minutes.

Clearly, video capturing systems are widely used in sport environments all over the world, and a great effort has been put into building deep learning algorithms and computer systems for tracking object in videos, including sports. These types of systems allow us to collect a lot of data, but all types of algorithms and systems have their strengths and weaknesses. For example, video capturing

Figure 1. Methodology



systems are very sensitive to the lightning and environmental conditions, such as weather conditions, that are difficult to control, but, on the other hand, they provide a great amount of data to analyze.

For these reasons, it could be useful to combine different data acquisition approaches, such as video capturing method and tracking systems, in order to obtain better results.

2.2 Dataset Creation

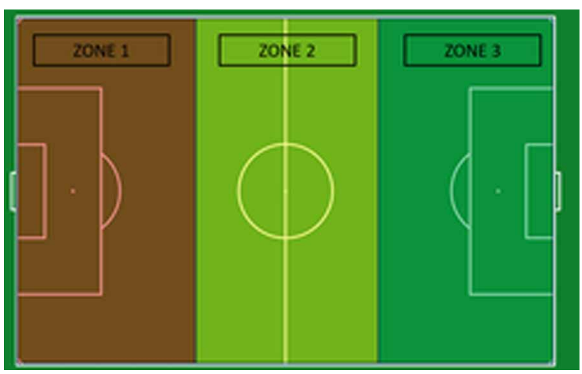
The second step of our methodology is about the dataset construction, that contains the information that we have to use to perform our analysis. In this phase, it can be used some algorithms, like normalization techniques, that allow us to do some operations on the data to get it more understandable. At the end, the dataset has to contain information about the position of the players involved in the game, the position of the ball and information about players' body orientation and movements.

2.3 Data Discretization

After the dataset creation, a discretization operation on player position variables (x player and y player) has been performed. Discretization is the process that allows to transform continuous variables, models or functions into a discrete form. To do this, a set of contiguous intervals (or bins) that go across the desired range has been created. After that, each evaluation of player positions, in x-axis and y-axis, is assigned to one of these intervals.

In soccer scenario, discretizing player position variables, means that we divide the pitch into *k* equal-width zones and we assign each evaluation to one of these zones (Figures 2 and 3 represent a discretization with k=3). We have to perform the discretization operation only on the variables that we want to classify (player position variables), in order to assign, to each player, a specific label which represents the zone occupied by that player.

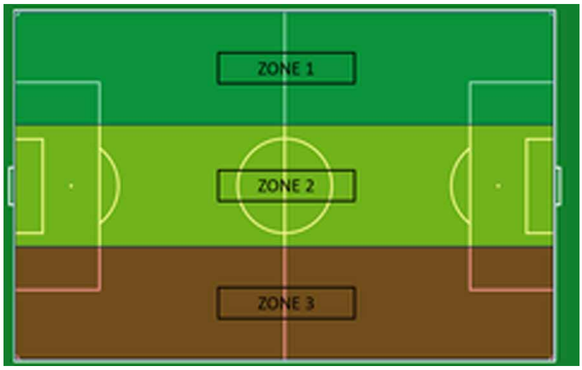
Figure 2. Division of the pitch on x-axis into three zones



2.4 Supervised Learning

In the fourth step, it has been performed a classification operation. Specifically, for this purpose, it can be used different supervised machine learning techniques. The variables that we want to predict are those related to the positions of the players (x position and y position). In our specific case, the classification consists in the assignment of a specific zone of the pitch to each player involved in the game, based on the predictors that we have at our disposal (ball position, heading, direction and speed). In such a way, we want to understand if there exist some predictors that are very informative and discriminant in order to explain the position of a player in the field.

Figure 3. Division of the pitch on y-axis into three zones



There are different types of classification algorithms that can be used for these purposes and, in the experimental results section, about our analysis, we have shown only those that are referred to the tree-based classification algorithms, since with these we obtain the best results.

3. EXPERIMENTAL ANALYSIS

3.1 Dataset

For our purposes, the dataset used has been constructed in (Pettersen et al.,2014). In this work, the authors have proposed a dataset of elite soccer player movements and ball position information. The dataset is captured at Alfheim Stadium, the home arena of Tromsø (Norway), during the match against Tottenham Hotspur. All the data refers to the home team. The player positions are measured at 20 Hz using the ZXY Sport Tracking system that is based on a two dimensional positional coordinate system, inside the stadium in which the game is played. This means, that we have 20 evaluations per second for each player of the home team, for both axes of the pitch (X and Y axes).

The reference system is composed by two-dimensional positional coordinates in which the positive *x*-axis points to the right with respect to the camera shooting, along the side of the field; while the positive *y*-axis points upwards, along the short edge of the field, as shown in Figure4.

The position (0, 0) is located in the lower-left corner of the image captured by the camera. The soccer pitch is 105 *mt* × 68 *mt* of dimension so, the values for *x position* and *y position* are, respectively, in the range of $0 \leq .\ x \leq .105$ and $0 \leq .y \leq .68$.

The variables referred to the players are represented in Figure 5.

About the ball position, instead, the information has been extracted manually by the researchers from video analysis. Merging the ball and players information, and removing some useless variables, we have obtained a dataset with approximately 495.000 observations and 7 variables, that are represented in Table 1.

3.2 Classification Analysis

In this section we present the results obtained.

For our analysis, we have considered three different discretization of the variables that we want to classify, x position and y position of each player. Specifically, the two featureshave been discretized into 3, 4 and 5 labels. Each label represents a single zone with which we divide the field. So, if we consider a 3-label discretization, we are considering the pitch divided into three equal-width zones. Our goal is to assign the position of each player, in terms of x-axis and y-axis, to a specific zone.

Figure 4. Pitch reference system

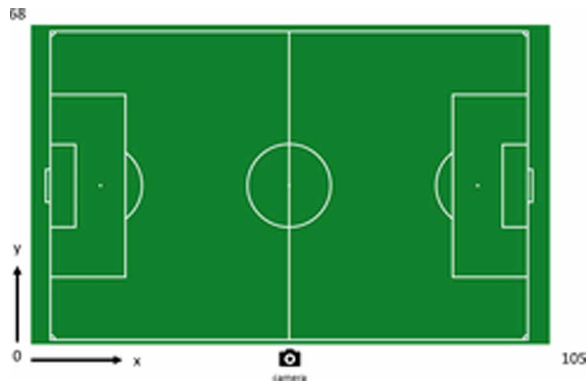


Figure 5. Samples from the 20 Hz ZXY sensor traces

```
'timestamp','tag_id','x_pos','y_pos','heading','direction','energy','speed','total_distance'
   ...
'2013-11-03 18:30:00.000612',31278,34.2361,49.366,2.2578,1.94857,3672.22,1.60798,3719.61
'2013-11-03 18:30:00.004524',31890,45.386,49.8209,0.980335,1.26641,5614.29,2.80983,4190.53
'2013-11-03 18:30:00.013407',0918,74.5904,71.048,-0.961152,0,2.37406,0,0.285215
'2013-11-03 18:30:00.015759',109,60.2843,57.3384,2.19912,1.22228,4584.61,8.14452,4565.93
'2013-11-03 18:30:00.023466',909,45.0113,54.7307,2.23514,2.27993,4170.35,1.76589,4070.6
   ...
```

In order to evaluate the classification that we have performed, five different metrics have been considered: false positive rate, precision, recall, F-Measure and Roc area. We have obtained, for each metrics, a value for each zone and an average value.

The precision is equals to the proportion of the instances that is predicted to be in Zone *X* and effectively belong to a specific Zone *X*, among all those which were assigned to that Zone. In other words, it is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved:

$$\text{Precision} = \frac{tp}{tp + fp}$$

Table 1. 3-Label discretization

Feature	Description	Info
F1	x_player	Relative position, in meters, of the player, on the x-axis.
F2	y_player	Relative position, in meters, of the player, on the y-axis.
F3	heading	Direction to the player is facing, in radians, where 0 is the direction of the y-axis.
F4	direction	Direction to the player is travelling, in radians, where 0 is the direction of the y-axis.
F5	speed	Player speed, in meters per seconds.
F6	x_ball	Relative position, in meters, of the ball, on the x-axis.
F7	y_ball	Relative position, in meters, of the ball, on the y-axis.

where tp indicates the numbers of true positives and fp indicates the numbers of false positives.

The recall has been computed as the proportion of the instances that is predicted to be in Zone X and effectively belong to a specific Zone X, among all the instances that truly belong to that class, i.e., how much part of the class was captured. So, it is the ratio of the number of relevant records retrieved to the total number of relevant records:

$$\text{Recall} = \frac{tp}{tp + fn}$$

where tp indicates the number of true positives and fn indicates the number of false negatives.

The F-Measure is a measure of a test's accuracy. This score can be interpreted as a weighted average of the precision and recall:

$$\text{F-Measure} = 2\frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

The Roc Area is defined as the probability that a positive instance randomly chosen is classified above a randomly chosen.

The results, in the next section, are refereed to only 4 different type of tree-based algorithms:

- **J48**: is an open source Java implementation of the C4.5 decision tree algorithm that is used in data mining as a decision tree classifier which can be employed to generate a decision, based on a certain sample of data (univariate or multivariate predictors).
- **Rep Tree**: is a fast decision tree learner, based on C4.5 algorithm, and it can pro- duce classification (discrete outcome) or regression trees (continuous outcome). It builds a regression/decision tree using information gain/variance and prunes it using reduced-error pruning (with back-fitting).
- **Random Tree**: is an algorithm that constructs a tree that randomly selects at- tributes at each node. It performs no pruning.
- **Hoeffding Tree**: is an incremental, anytime decision tree induction algorithm that is capable of learning from massive data streams, assuming that the distribution generating examples does not change over time. Hoeffding trees exploit the fact that a small sample can often be enough to choose an optimal splitting attribute. This idea is supported mathematically by the Hoeffding bound, which quantifies the number of observations (in our case, examples) needed to estimate some statistics within a prescribed precision (in our case, the goodness of an attribute).

Clearly, We have tried to use also different algorithms but they are not mentioned because the results are very low.

In our approach, classification analysis means to construct a classifier in order to understand in which zone, among the k considered, a specific player is localized. The number of the zones depends on the type of discretization that we have performed on the position's player variable. Specifically, for the division of the pitch, we have started with a 3-zones partition, according to the traditional classification that is usually used in soccer to describe the general behavior of a team (*defensive, offensive or ultra-offensive team*). After that, we have also considered a 4-zones and 5-zones field divisions. The case of 2- zones division has not been considered since, for our purposes, it would be meaningless in statistical terms.

To deal with the absence of a very large designated test set that can be used to directly estimate the test error rate, that is the average error that results from using a statistical learning method to predict the response on a new observation, for the learning phase, it has been used a k -fold cross validation. This approach involves randomly dividing the set of observations into k groups, or folds, of approximately equal size. The first fold is treated as a validation set, and the model is constructed starting from the remaining k - 1 folds. The Mean Squared Error (MSE), is then computed on the observation in the held-out fold. This procedure is repeated k times; each time, a different group of observations is treated as a validation set. This process results in k estimates of the test error. The k -fold cross validation estimate is computed by averaging these values. Clearly, when we consider a classification problem, cross validation works at the same way, except that rather than using MSE to quantify test error, we instead use the number of misclassified observations. In our approach, we have used a k equals to 20.

We have evaluated the effectiveness of the classification method with the following procedure:

1. Build a training set $T \subset D$.
2. Build a testing set $T' = D \div T$.
3. Run the training phase on T .;
4. Apply the learned classifier to each element of T' ..

3.3 Results

As previously discussed, for each classification we considered 95% of the dataset as training dataset and 5% as testing dataset employing the full feature set.

For the experimental analysis and for each type of classification, are presented the results in relation to four different tree-based algorithm: J48, Rep Tree, Random Tree and Hoeffding Tree. The analysis has been performed also using other types of algorithm, that are not illustrated since the results are really low. For each type of discretization used in the analysis, the best results are obtained in correspondence of the Random Tree algorithm; while, the worst performance, among the algorithms used, is for Hoeffding Tree.

For what concern the 3-label classification, we obtain an accuracy of 0.954 for x position and 0.920 for y position. The algorithm with the worst performance is Hoeffding Tree with an accuracy of 0.781 for x position analysis and 0.668 for y position analysis.

In Figure 6 and Figure 7, it has been show the confusion matrices for Random Tree algorithm, with a 3 label discretization.

As we have said before, also for the 4-label classification, the algorithm that performs well is Random Tree with an accuracy of 0.936 for x position and 0.898 for y position; while, with the Hoeffding Tree, we obtain 0.678 (x-axis) and 0.583 (y-axis). The confusion matrices for Random Tree algorithm are set below (Figure 8 and Figure 9).

Finally, with the 5-label classification we have results that are very similar to the previous

discretization. Specifically, we obtain 0.924 and 0.893 in relation to, respectively, x and y axis, with Random Tree. The worst performance is, even in this case, for the Hoeffding Tree algorithm with an accuracy of 0.621 (x-axis) and 0.534 (y-axis). The confusion matrices for Random Tree algorithm are set below (Figure 10 and Figure 11).

For all the analysis performed the level of accuracy of the other two algorithms, J48 and Rep Tree, is slightly lower with respect to the best algorithm, Random Tree. Another aspect of the results obtained is that, for each level of discretization, the number of FP and FN is the same, since the precision and accuracy take the same value for each algorithm.

Starting from the results obtained, we can conclude that we are able, with this method- ology, to identify the position of a player with a good level of accuracy. This is important because the coach can verify if a specific player, in a certain game situation, respect his guidelines in terms of position. This can be useful when the coach wants to obtain a general comprehension of the position of each player, in every game situation, based on the ball position.

Figure 6. Confusion matrix for Random Tree Algorithm in x-axis position prediction (k=3)

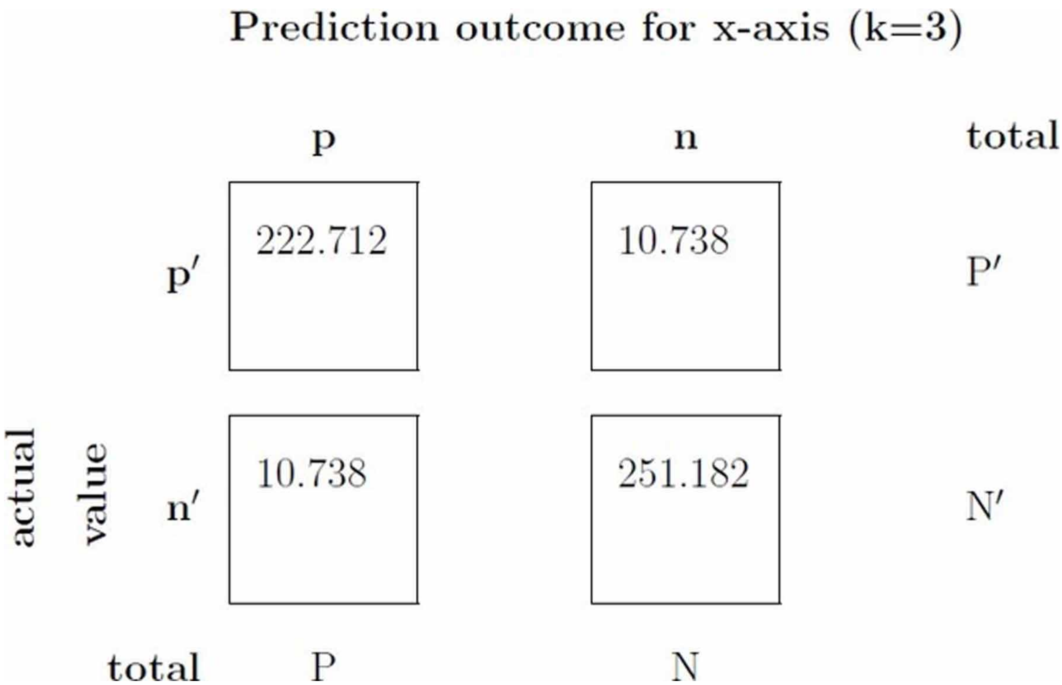


Figure 7. Confusion matrix for Random Tree Algorithm in y-axis position prediction (k=3)

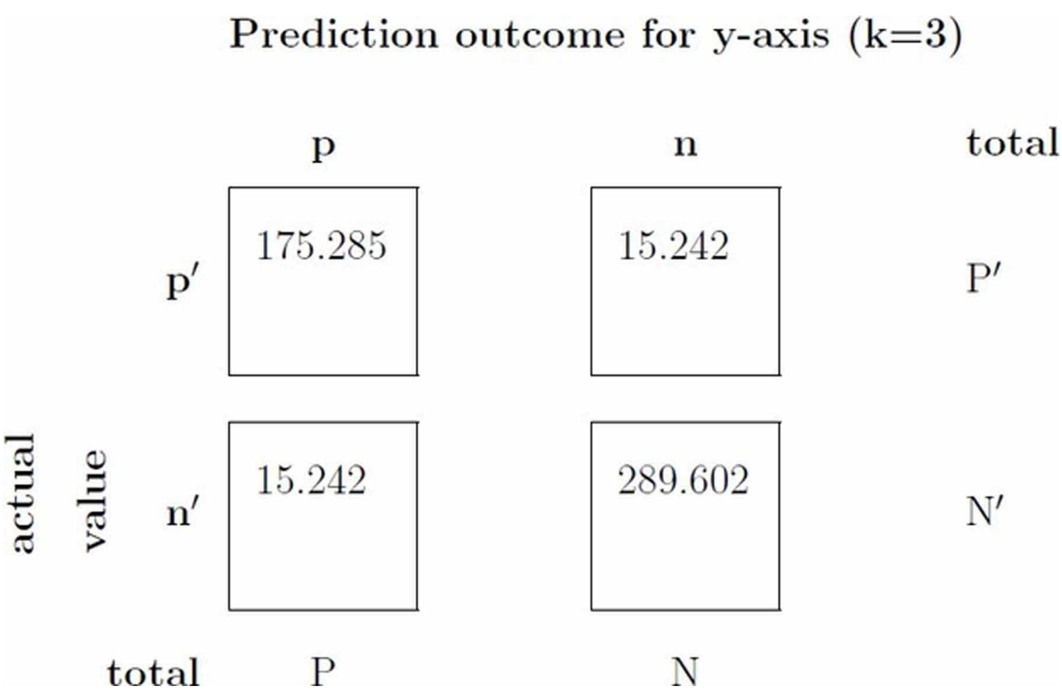


Figure 8. Confusion matrix for Random Tree Algorithm in x-axis position prediction (k=4)

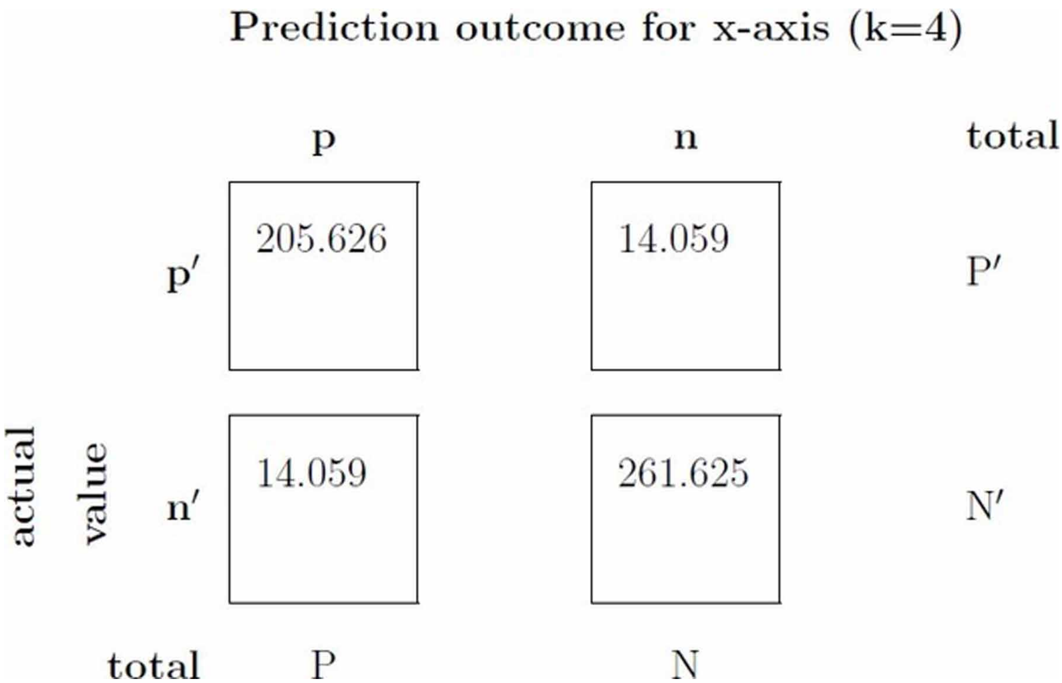


Figure 9. Confusion matrix for Random Tree Algorithm in y-axis position prediction (k=4)

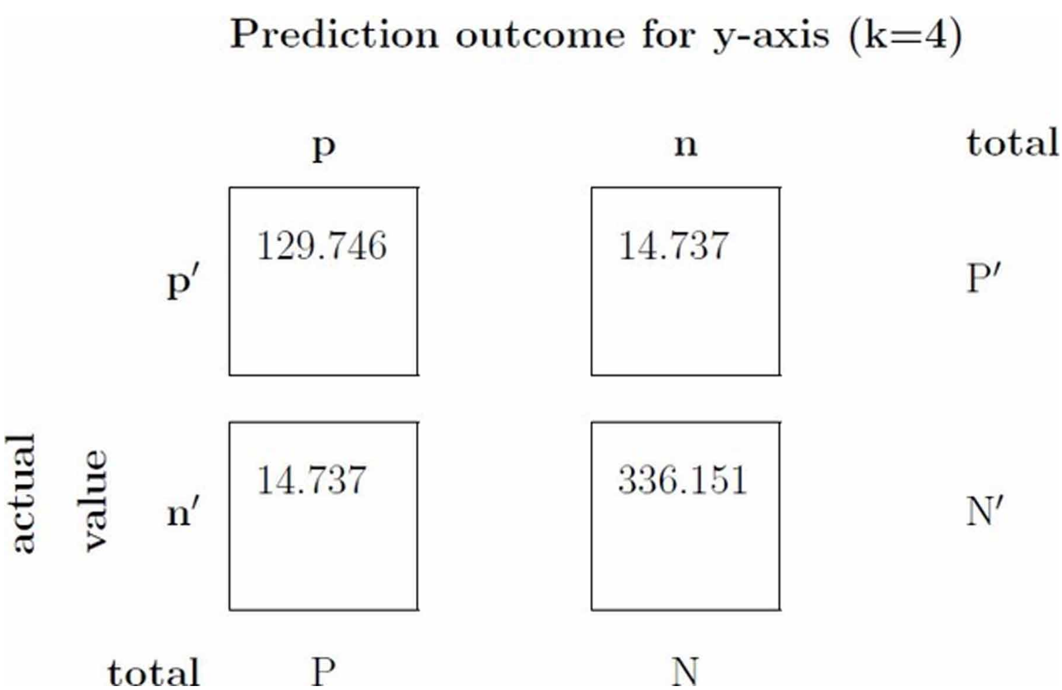


Figure 10. Confusion matrix for Random Tree Algorithm in x-axis position prediction (k=5)

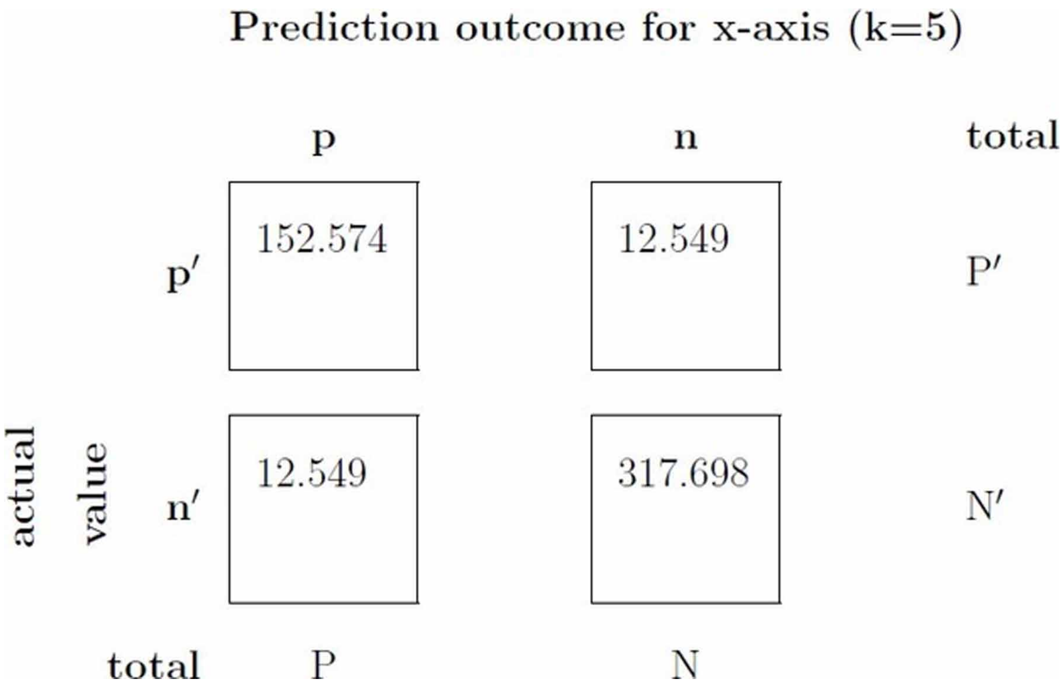
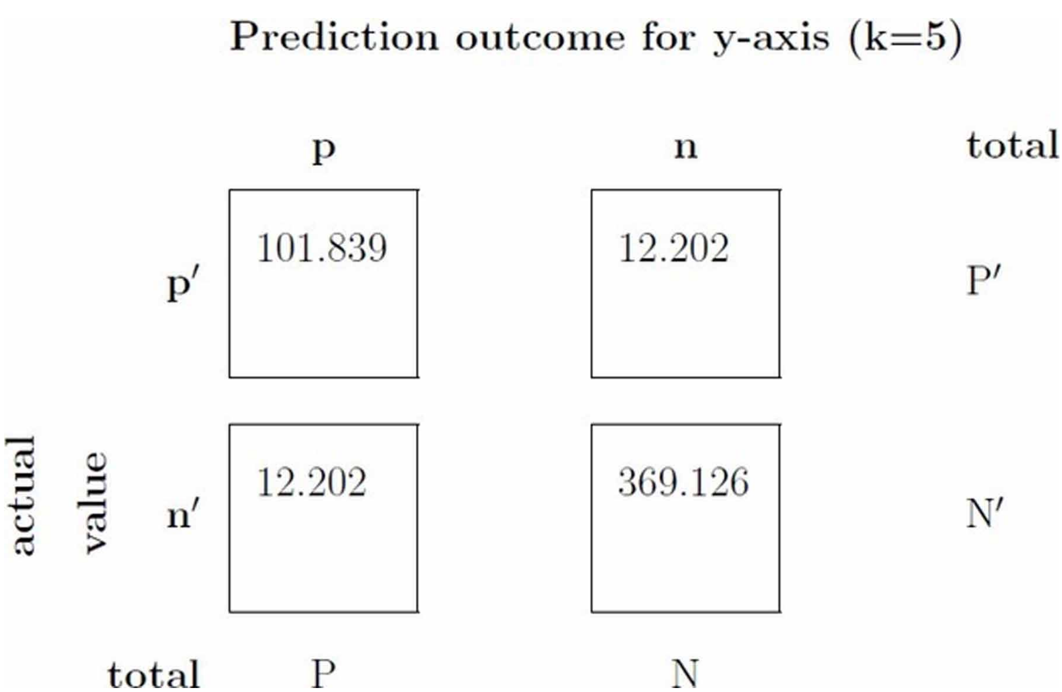


Figure 11. Confusion matrix for Random Tree Algorithm in y-axis position prediction (k=5)



4. CONCLUSION

In this work, it has been proposed a method to predict the position occupied by each player of a team (both horizontally and vertically), with respect to different types of information, such as the ball position, the direction toward the player are moving and looking at, and the speed of the player. The key point of this approach is that it can be used in real-time

during a specific match, in order to verify if a player (or group of players) is playing according to the strict guidelines of the coach. If not, the coach and the players can adjust tactical and positional aspects to deal with the various problems that can be encountered during a match. For example, our approach can be used to check if a player, based on the position of the ball, is occupying the correct zone of the field.

The proposed method exploits machine learning techniques building models with four different algorithms: J48, Rep Tree, Random Tree and Hoeffding Tree. Additionally, it has been considered 3 different types of discretizations of the variables that we want to classify, *x player position* and *y player position*. Each type of discretization represents a specific division of the field, i.e., 3-label discretization means that the pitch has been divided into 3 equal-width zones. For all the three discretizations, we have obtained the best results with Random Tree algorithm. Specifically, for 3-label discretization, we have reached an average precision and recall equal to 0.954 along x-axis, while an average precision and recall equal to 0.920 along y-axis. Clearly, the accuracy is worse as the number of zones considered increases, even if we have obtained a good results with 5-label discretization too, with an average precision and recall equals to 0.924, for both metrics, along x-axis; while, along y-axis, an average precision and recall, both, equals to 0.893. As future work, we will consider the application of formal method techniques, in order to improve the explainability of our results and to construct a system that can support the decision of the coach, during a match, verifying if a specific player observes his guidelines.

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

REFERENCES

Bartlett, R. (2001). Performance analysis: Can bringing together biomechanics and notational analysis benefit coaches? *International Journal of Performance Analysis in Sport*, *1*(1), 122–126. doi:10.1080/24748668.2001.11868254

Berrar & Dubitsky. (2019). Incorporating domain knowledge in machine learning for soccer outcome prediction. *Special Issue on Machine Learning for Soccer*.

Capobianco, G., Di Giacomo, U., Mercaldo, F., Nardone, V., & Santone, A. (2019). Can machine learning predict soccer match results? *11th International Conference on Agents and Artificial Intelligence*, 458–465. doi:10.5220/0007307504580465

Dietrich, C., Koop, D., Vo, H. T., & Silva, C. T. (2014). Baseball4d: A tool for baseball game reconstruction & visualization. *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, 23–32.

Fernandez & Cervone. (2019). Decomposing the immeasurable sport: A deep learning ex- pected possession value framework for soccer. *Sport Analytics Conference*.

Hughes & Franks. (2004). Notational analysis of sport. *Journal of Sports Science and Medicine*, *3*, 6.

Jain & Kaur. (2017). *Machine learning approaches to predict basketball game outcome*. doi: .10.1109/ICACCAF.2017.8344688

Joseph, A., Fenton, N. E., & Neil, M. (2006). Predicting football results using bayesian nets and other machine learning techniques. *Knowledge-Based Systems*, *19*(7), 544–553.

Kharrat, T., Pen˜a, J., & Mchale, I. (2017). Plus-minus player ratings for soccer. *European Journal of Operational Research*.

Liti, C., Piccialli, V., & Sciandrone, M. (2017). Predicting soccer match outcome using machine learning algorithms. *Proceedings of MathSport International 2017 Conference*, 229.

Liu & Schulte. (2018). *Deep reinforcement learning in ice hockey for context-aware player evaluation.* doi: 10.24963/ijcai.2018/478

Pettersen, A., Johansen, D., Johansen, H., Berg-Johansen, V., Gaddam, V. R., Mortensen, A., Langseth, R., Griwodz, C., Stensland, H. K., & Halvorsen, P. (2014). Soccer video and player position dataset. In *Proceedings of the 5th ACM Multimedia Systems Conference, MMSys '14*, (pp. 18–23). Association for Computing Machinery. doi:10.1145/2557642.2563677

Razali, M. (2017). *Yatim, and A. Aziz. Predicting football matches results using bayesian networks for English Premier League (EPL)*. International Research and Innovation Summit.

Schultze & Wellbrock. (2017). A weighted plus/minus metric for individual soccer player performance. *Journal of Sports Analytics*, *4*, 1–11. doi: 10.3233/JSA-170225

APPENDIX A: TABLES

Table 2. Classification results for x player position (3-labeled): FP rate, Precision, Recall, F-Measure and RocArea computed with J48, RepTree, RandomTree, Naive Bayes, Hoeffding Tree and Decision Stump classification algorithms

Algorithm	FP rate	Precision	Recall	F-Measure	Roc Area	Result Prediction
J48	0.007	0.901	0.885	0.892	0.966	*Zone 1*
	0.060	0.958	0.961	0.959	0.965	*Zone 2*
	0.028	0.948	0.946	0.947	0.973	*Zone 3*
	0.045	0.950	0.950	0.950	0.968	*Average*
Rep Tree	0.007	0.901	0.885	0.892	0.966	*Zone 1*
	0.060	0.958	0.961	0.959	0.965	*Zone 2*
	0.02	0.948	0.946	0.947	0.973	*Zone 3*
	0.045	0.950	0.950	0.950	0.968	*Average*
Random Tree	0.007	0.907	0.895	0.901	0.944	*Zone 1*
	0.055	0.962	0.963	0.963	0.954	*Zone 2*
	0.026	0.951	0.951	0.951	0.962	*Zone 3*
	0.041	0.954	0.954	**0.954**	0.957	*Average*
Hoeffding Tree	0.016	0.570	0.300	0.393	0.902	*Zone 1*
	0.293	0.806	0.857	0.830	0.864	*Zone 2*
	0.116	0.779	0.766	0.772	0.911	*Zone 3*
	0.213	0.781	0.788	0.781	0.883	*Average*

Table 3. Classification results for y player position (3-labeled): FP rate, Precision, Recall, F-Measure and RocArea computed with J48, RepTree, RandomTree, Naive Bayes, Hoeffding Tree and Decision Stump classification algorithms

Algorithm	FP rate	Precision	Recall	F-Measure	Roc Area	Result Prediction
J48	0.024	0.865	0.852	0.859	0.947	*Zone 1*
	0.078	0.908	0.919	0.914	0.943	*Zone 2*
	0.040	0.936	0.927	0.932	0.962	*Zone 3*
	0.055	0.912	0.912	0.912	0.951	*Average*
Rep Tree	0.034	0.806	0.778	0.792	0.949	*Zone 1*
	0.117	0.865	0.893	0.879	0.940	*Zone 2*
	0.052	0.916	0.893	0.904	0.965	*Zone 3*
	0.079	0.876	0.875	0.875	0.951	*Average*
Random Tree	0.023	0.876	0.872	0.874	0.925	*Zone 1*
	0.069	0.918	0.923	0.921	0.927	*Zone 2*
	0.039	0.938	0.934	0.936	0.947	*Zone 3*
	0.050	0.920	0.920	**0.920**	0.935	*Average*
Hoeffding Tree	0.022	0.589	0.174	0.268	0.806	*Zone 1*
	0.478	0.616	0.913	0.736	0.782	*Zone 2*
	0.047	0.897	0.641	0.747	0.869	*Zone 3*
	0.240	0.721	0.693	0.668	0.820	*Average*

Table 4. Classification results for x player position (4-labeled): FP rate, Precision, Recall, F-Measure and RocArea computed with J48, RepTree, RandomTree, Naive Bayes, Hoeffding Tree and Decision Stump classification algorithms

Algorithm	FP rate	Precision	Recall	F-Measure	Roc Area	Result Prediction
J48	0.005	0.892	0.881	0.887	0.964	*Zone 1*
	0.026	0.896	0.881	0.888	0.959	*Zone 2*
	0.082	0.945	0.953	0.949	0.957	*Zone 3*
	0.015	0.921	0.917	0.919	0.971	*Zone 4*
	0.057	0.929	0.930	0.929	0.960	*Average*
Rep Tree	0.070	0.837	0.814	0.825	0.975	*Zone 1*
	0.036	0.853	0.826	0.839	0.961	*Zone 2*
	0.121	0.920	0.935	0.927	0.957	*Zone 3*
	0.021	0.888	0.876	0.882	0.978	*Zone 4*
	0.083	0.898	0.899	0.898	0.962	*Average*
Random Tree	0.004	0.903	0.895	0.899	0.945	*Zone 1*
	0.024	0.903	0.898	0.900	0.937	*Zone 2*
	0.072	0.951	0.955	0.953	0.941	*Zone 3*
	0.014	0.925	0.922	0.924	0.954	*Zone 4*
	0.051	0.936	0.936	**0.936**	0.943	*Average*
Hoeffding Tree	0.009	0.510	0.220	0.308	0.900	*Zone 1*
	0.070	0.598	0.416	0.491	0.829	*Zone 2*
	0.455	0.736	0.855	0.791	0.806	*Zone 3*
	0.069	0.612	0.566	0.588	0.907	*Zone 4*
	0.297	0.679	0.695	0.678	0.831	*Average*

Table 5. Classification results for y player position (4-labeled): FP rate, Precision, Re- call, F-Measure and RocArea computed with J48, RepTree, RandomTree, Naive Bayes, Hoeffding Tree and Decision Stump classification algorithms

Algorithm	FP rate	Precision	Recall	F-Measure	Roc Area	Result Prediction
J48	0.014	0.855	0.846	0.851	0.942	*Zone 1*
	0.059	0.860	0.867	0.864	0.931	*Zone 2*
	0.062	0.871	0.872	0.872	0.932	*Zone 3*
	0.018	0.955	0.949	0.952	0.977	*Zone 4*
	0.044	0.891	0.891	0.891	0.945	*Average*
Rep Tree	0.020	0.781	0.755	0.768	0.953	*Zone 1*
	0.089	0.793	0.809	0.801	0.927	*Zone 2*
	0.092	0.810	0.818	0.814	0.927	*Zone 3*
	0.024	0.940	0.920	0.930	0.982	*Zone 4*
	0.065	0.840	0.839	0.840	0.945	*Average*
Random Tree	0.013	0.867	0.861	0.864	0.924	*Zone 1*
	0.055	0.870	0.874	0.872	0.910	*Zone 2*
	0.058	0.879	0.880	0.879	0.911	*Zone 3*
	0.018	0.956	0.953	0.955	0.968	*Zone 4*
	0.042	0.898	0.898	**0.898**	0.928	*Average*
Hoeffding Tree	0.016	0.475	0.153	0.232	0.824	*Zone 1*
	0.244	0.456	0.487	0.471	0.726	*Zone 2*
	0.323	0.489	0.640	0.554	0.742	*Zone 3*
	0.017	0.947	0.747	0.835	0.927	*Zone 4*
	0.184	0.611	0.583	0.583	0.798	*Average*

Table 6. Classification results for x player position (5-labeled): FP rate, Precision, Recall, F-Measure and RocArea computed with J48, RepTree, RandomTree, Naive Bayes, Hoeffding Tree and Decision Stump classification algorithms

Algorithm	FP rate	Precision	Recall	F-Measure	Roc Area	Result Prediction
J48	0.003	0.889	0.879	0.884	0.962	*Zone 1*
	0.010	0.880	0.873	0.876	0.960	*Zone 2*
	0.044	0.909	0.908	0.909	0.955	*Zone 3*
	0.044	0.935	0.939	0.937	0.961	*Zone 4*
	0.011	0.913	0.906	0.910	0.966	*Zone 5*
	0.041	0.918	0.918	0.918	0.960	*Average*
Rep Tree	0.005	0.825	0.796	0.810	0.975	*Zone 1*
	0.015	0.825	0.800	0.812	0.970	*Zone 2*
	0.065	0.866	0.873	0.869	0.957	*Zone 3*
	0.077	0.908	0.913	0.910	0.965	*Zone 4*
	0.016	0.873	0.858	0.866	0.977	*Zone 5*
	0.059	0.881	0.881	0.881	0.964	*Average*
Random Tree	0.003	0.894	0.881	0.887	0.939	*Zone 1*
	0.010	0.886	0.886	0.886	0.938	*Zone 2*
	0.040	0.917	0.916	0.916	0.938	*Zone 3*
	0.050	0.940	0.944	0.942	0.947	*Zone 4*
	0.011	0.915	0.909	0.912	0.949	*Zone 5*
	0.038	0.924	0.924	**0.924**	0.943	*Average*
Hoeffding Tree	0.007	0.414	0.163	0.234	0.913	*Zone 1*
	0.026	0.491	0.290	0.365	0.872	*Zone 2*
	0.240	0.577	0.681	0.625	0.805	*Zone 3*
	0.245	0.717	0.731	0.721	-0.842	*Zone 4*
	0.048	0.540	0.441	0.485	0.909	*Zone 5*
	0.197	0.622	0.630	0.621	0.842	*Average*

Table 7. Classification results for y player position (5-labeled): FP rate, Precision, Recall, F-Measure and RocArea computed with J48, RepTree, RandomTree, Naive Bayes, Hoeffding Tree and Decision Stump classification algorithms

Algorithm	FP rate	Precision	Recall	F-Measure	Roc Area	Result Prediction
J48	0.010	0.845	0.840	0.843	0.941	*Zone 1*
	0.034	0.835	0.842	0.838	0.935	*Zone 2*
	0.060	0.861	0.867	0.864	0.929	*Zone 3*
	0.040	0.853	0.845	0.849	0.931	*Zone 4*
	0.010	0.972	0.968	0.970	0.986	*Zone 5*
	0.035	0.882	0.882	0.847	0.946	*Average*
Rep Tree	0.013	0.783	0.751	0.767	0.955	*Zone 1*
	0.050	0.762	0.769	0.765	0.939	*Zone 2*
	0.088	0.799	0.819	0.809	0.930	*Zone 3*
	0.057	0.789	0.775	0.782	0.936	*Zone 4*
	0.012	0.963	0.952	0.957	0.991	*Zone 5*
	0.051	0.831	0.831	0.831	0.950	*Average*
Random Tree	0.008	0.865	0.859	0.862	0.925	*Zone 1*
	0.031	0.853	0.858	0.856	0.914	*Zone 2*
	0.053	0.876	0.878	0.877	0.912	*Zone 3*
	0.038	0.862	0.859	0.860	0.910	*Zone 4*
	0.009	0.973	0.972	0.972	0.981	*Zone 5*
	0.032	0.893	0.893	**0.893**	0.930	*Average*
Hoeffding Tree	0.012	0.421	0.137	0.207	0.835	*Zone 1*
	0.064	0.374	0.186	0.248	0.737	*Zone 2*
	0.377	0.445	0.705	0.546	0.740	*Zone 3*
	0.147	0.422	0.390	0.405	0.755	*Zone 4*
	0.011	0.963	0.839	0.897	0.957	*Zone 5*
	0.159	0.558	0.549	0.534	0.803	*Average*