University of Warsaw
Faculty of Economic Sciences

Working Papers
No. 13/2021 (361)

MACHINE LEARNING IN THE PREDICTION OF FLAT HORSE RACING RESULTS IN POLAND

Piotr Borowski
Marcin Chlebus

Warsaw 2021

WORKING PAPERS 13/2021 (361)

Machine learning in the prediction of flat horse racing results in Poland

Piotr Borowski*, Marcin Chlebus

University of Warsaw, Faculty of Economic Sciences
** Corresponding author: piotr_borowski@int.pl*

Abstract: Horse racing was the source of many researchers considerations who studied market efficiency and applied complex mathematic formulas to predict their results. We were the first who compared the selected machine learning methods to create a profitable betting strategy for two common bets, Win and Quinella. The six classification algorithms under the different betting scenarios were used, namely Classification and Regression Tree (CART), Generalized Linear Model (Glmnet), Extreme Gradient Boosting (XGBoost), Random Forest (RF), Neural Network (NN) and Linear Discriminant Analysis (LDA). Additionally, the Variable Importance was applied to determine the leading horse racing factors. The data were collected from the flat racetracks in Poland from 2011-2020 and described 3,782 Arabian and Thoroughbred races in total. We managed to profit under specific circumstances and get a correct bets ratio of 41% for the Win bet and over 36% for the Quinella bet using LDA and Neural Networks. The results demonstrated that it was possible to bet effectively using the chosen methods and indicated a possible market inefficiency.

Keywords: horse racing prediction, racetrack betting, Thoroughbred and Arabian flat racing, machine learning, Variable Importance

JEL codes: C53, C55, C45

Introduction

For many years, horse racing has aroused many emotions among the audience. Fresh air, dignified animals, sportsmanship, competition and the possibility of placing bets attract the public effectively.

The most extensive horse racing tracks in Poland are located in Warsaw, Wroclaw, and Sopot. Before each event taking place on them, we can buy a race program at a low price. It contains information about starting horses, jockeys, and their statistics from previous races. Many bettors read them to help increase their chances of winning a bet. Some of them base their results on thoughts, others on the formulas they have written. The topic also interested the researchers who used different scientific approaches in their predictions. They studied market efficiency (Asch, Malkiel, Quandt, 1984) & (Gabriel, Marsden, 1990), ranking probability models (Lo, Bacon-Shone, Busche, 1995), the application of Neural Networks (Williams & Li, 2008), Support Vector Regression (Schumaker, 2013) and many others (Hausch and Ziemba, 1981), (Henery, 1981), (Pudaruth, Medard and Dookhun, 2013).

In the research, we decided to use machine learning algorithms to predict the race results. We aimed to answer what features of the starting horse influence the race's high place and whether it is the same for all bets. After all, based on the data, we wanted to check how much we could earn and which betting strategy was the best or completely unprofitable. According to our knowledge, there is no similar study analysing horse racing results prediction in Poland. At the same time, it is a widely discussed topic as evidenced by many articles and even whole books such as the famous 'Dr. Z's Beat the Racetrack' by Ziemba & Hausch (1987) or 'The Skeptical Handicapper: Using Data and Brains to Win at the Racetrack" by Barry Meadow (2019).

This work uses publicly available data from seasons 2011 to 2020 for Arabian and Thoroughbred horse races to predict Win and Quinella's two common bets. The Win is when a given horse competing in the race will finish in the first position (top 1). The definition of a Quinella is to select the first two finishers in a horse race in any order (top 2). We took a completely new approach in terms of the selection of variables and the methods used. We applied six classification models, namely Classification and Regression Tree (CART), Generalized Linear Model (Glmnet), Extreme Gradient Boosting (XGBoost), Random Forest (RF), Neural Network (NN) and Linear Discriminant Analysis (LDA). The performance of the models was later compared based on three criteria: the AUC measure, the possible rate of return

from the placed bets and the ratio of correctly placed bets. Explainable AI (XAI) tools were used to ensure the model's reliability and quality of prediction. Variable Importance (VI) was applied to understand the crucial features of the models and how black-box models made their predictions.

The structure of this paper was composed as follows. In the first part, we presented a literature review. The second part contains materials and methodology. In the third part, we described the results. In the last part, we presented a summary and included conclusions.

Literature overview

The approach to research on the prediction of horse racing results has evolved over the years. Schumaker (2013) pointed three main areas on this topic: market efficiency, mathematics, and data mining. While some of the presented studies examples were concerned with harness racing and not flat racing, the conclusions drawn can be applied to both types of racing.

In the 1980s and 1990s, a popular topic in racetracks was market efficiency. When the market is efficient, prices fully reflect all available information (Fama, 1970). The idea was to determine whether tracking the odds (the information about the wagers) could effectively predict the race outcome. In their study, Hausch, Ziemba, and Rubinstein (1981) analysed betting horses to Place (given horse finishes as first or second) or to Show (given horse finishes as first, second, or third) and found that inefficiencies exist. Doubts arose as to whether there exist departures from efficiency sufficiently large to permit profitable betting strategies. Researchers wanted to prove that profit can be made, not only on paper, by analysing the past results of races but also in reality (Asch, Malikel, Quandt, 1984). For this purpose, they asked two questions: "Can one, (..), devise strategies based on observable betting patterns that imply positive rates of return?" and "If one were able to predict winners by using publicly available information, would such a finding be inconsistent with the assumption of rational behaviour in the racetrack market?". They studied 712 races from season 1978 at the Atlantic City with 5,714 horses and included information about the odds determined by the professional handicappers and the players. Next, they applied logit analysis to the obtained data. They concluded that although it was possible to outperform familiar bettors in the Win (top 1) bet, the profits could not be earned finally. It turned out otherwise in the case of Place and Show bets. Net profit in these scenarios could be made, but probably not on a significant scale.

Bird and McCrae (1987) studied market efficiency in Australia. The data came from Melbourne racetracks from the years 1983-1984 and consisted of 1026 events. Their research

evaluated the movement in bookmaker odds during betting on each race and the selections of newspaper tipsters. Based on the results, market efficiency was claimed in those two examples. However, they found out that "those with prior knowledge of movements in odds during the course of betting could use this knowledge to earn significant returns". Firstly, it could suggest that not all information was included in the odds; secondly, that there was a group of people with access to private information whose wagers could earn a positive return. Further, it could prove market inefficiency. This phenomenon has been proven by Gabriel and Marsden (1990), who compared starting price bets placed with bookmakers and totalizator from England. According to their study, there was substantial evidence under which the British racetrack market did not satisfy the conditions of semistrong efficiency. There were also high chances to fail to meet conditions for solid efficiency.

The mathematics in horse racing consists of presenting the observed variables in the form of the model. Harville (1973) proposed using the ranking model created by Luce and Suppes (1965). He compared racing horses based on the values estimated by the win bet fractions. If we assumed that Pi and Pj were the probabilities of winning horse i and j, the formula (example by Lo & Bacon-Shone, 1994) for the probability that horse i won the race and horse j finished the second was as follows:

$$P_{ij} = \frac{P_i * P_j}{1 - P_i}, \qquad (1)$$

Harville assumed that horse j finished before all other horses except horse i is independent of the event that horse i wins. This approach was considered simple and was commonly used in the future. However, McCulloch and Van Zijl (1986) pointed out that the model could be biased. The assumption was later confirmed by Lo and Bacon-Shone (1994). They compared the Harville model with the more sophisticated model created by Henery (1981) and proved the model had a systematic bias in estimating ordering probabilities. In subsequent analyses, Lo and Bacon-Shone proposed a new ranking probability model with Bushe assistance (1995) that coupled their previous studies with the Hausch and Ziemba (1981) HZR system. For the data from Hong Kong and the United States, they improved profits and lower levels of risk using final betting data assuming zero computational costs.

The other approach applied researchers from the University of Mauritius (Pudaruth, Medard and Dookhun, 2013) who decided to use the weighted probabilistic approach in predicting horse race results. In their studies, they collected data from 240 flat races for season 2010. Next, the results were thoroughly examined in order to find the factors that affect race.

They described jockey's and horse's characteristics such as the performance in the previous races, weight, odds and the others. Based on the interviews with experts from the horse business, they proposed nine scaled and weighted variables that created the formula of the model after summing. *Total = Factor 1 + Factor 2 + Factor 3 + Factor 4 + Factor5 + Factor 6 + Factor 7 + Factor 8 + Factor 9.* The horse with the highest Total result was predicted as with the highest chance for winning. The system had an accuracy of 58% and outperformed professional tipsters who could forecast only 44% of the winners.

The use of neural networks, machine learning and other data mining methods seems to be a natural process in predicting sports events due to the amount of data and the level of their complexity. They are accurate and can deal with imbalanced data. Its usage has been proven in many racing disciplines, e.g. Greyhound racing (Chen et al., 1994) or (Schumaker, Johnson, 2008).

Wiliams and Li (2008) studied horse racing in Jamaica racetracks. They applied four Neural Networks algorithms to predict the winner, namely Back-propagation, Quasi-Newton, Levenberg-Marquardt and Conjugate Gradient Descent. The data was composed of 143 flat races from 2007, for distances over 1000m to 3000m. Eight variables describing thoroughbred horse and jockeys characteristics were used as input neurons, and one neural network represented one horse. The research was performed with the usual for this study procedure with the 80%-20% split of training and test dataset. The Investigations showed that the average Accuracy of all four algorithms was 74%, and none of the used algorithms outperformed significantly.

In his research, Schumaker (2013) applied the machine learning algorithm called Support Vector Regression (SVR) to predict harness racing results. He decided to base his methods on the approach used in greyhound racing (Chen et al. 1994). The focus was on three main questions: "Can a Machine Learner predict Harness races better than established prediction methods?", "How important is race history to a machine learner?" and "What wager combinations work best and why?". The created S&C betting system was able to gather data from the browser, put them in the model, make a wager and analyse performance. The list of available bets was extensive and not limited only to the Win. The program could also manage the other bets such e.g. Place, Show, Exacta, Quinella, Trifecta, Trifecta Box. The collected data were obtained for the years 2009-2010 for Northfield Park, Ohio and contained 400 races. The selected eight variables describing horses past performance from the maximum of seven races were used to create the SVR model. As a result, the created betting system outperformed

the crowd and Dr Z System (1994) in all wagers. It was also proven that the history of four races maximises the profit and the Accuracy in a particular approach. Finally, the study and the possibility of making a profit using machine learning methods suggested an informational inequality within harness racing.

The given examples of literature show that the horse racing market is most likely inefficient. Many researchers successfully applied a variety of methods to predict the results of the racetrack. It gives a high chance of succeeding in our research and creating a profitable model to predict flat horse racing results in Poland.

2. Materials & Methods

The study aims to compare machine learning models in the prediction of horse racing. For this purpose, we applied six different and commonly used classification algorithms, namely Classification and Regression Tree (CART), Generalized Linear Model (Glmnet), Extreme Gradient Boosting (XGBoost), Random Forest (RF), Neural Network (NN) and Linear Discriminant Analysis (LDA).

We wanted to check whether we could create a profitable betting system. Additionally, we tried to answer the following questions.

- Which machine learning method performed the best for Arabian and Thoroughbred horses, and whether could we observe differences between those two race categories?
- How should a betting strategy look like? Should we place the bet only when our system indicates exactly one horse for Win and exactly two horses for Quinella, or should we choose those two horses with the highest probability?
- How much could we earn, and what would be the return on investment if we applied our system in reality?
- What features of the starting horse influence the race's high place, and whether it is the same for Win and Quinella bets?

2.1. Betting strategy

The way of betting on the racetrack differs from better known for soccer or basketball, where the bookmaker is the one that set the odds rate. In the case of horses, the payout changes and is the higher, the more people bet on the other horse than us, and the more bettors there are. The total jackpot comprises the amount paid by the players minus the bookmaker fees and tax and is paid in proportion to the bet placed (this is why we can speak of the crowd wisdom

phenomenon and market inefficiency that was discussed in the literature review section). For this reason, we defined four levels of probable win values for which we determined the profits of our models. The payouts were calculated as the multiplication of bet price times 1.5, 2, 3 and 4. These levels allowed to present the subject of the study in the most accessible form and omit the bias effect due to one extremely high bet missed.

The most popular bets available on the market are Win (bet on one horse that wins), Quinella (select the first two finishers in a horse race in any order), and Exacta (select the first two finishers in a horse race in a given order). In Poland, their cost is 5 PLN (approx. 1,12 EUR) for Win and 3PLN (approx. 0,67 EUR) for Exacta or Quinella. We observed that Quinella was not available for all of the races we analysed. Since the given bets did not precisely match our study's subject, prediction of top 1 and top 2 places, we had to adapt them. In our approach, we decided to buy two Exacta bets for the total price of 6 PLN (approx. 1,12 EUR). Thanks to this, we obtained the Quinella bet. (If we buy Exacta for horses A and B in the given order, and next Exacta for B and A, we get Quinella).

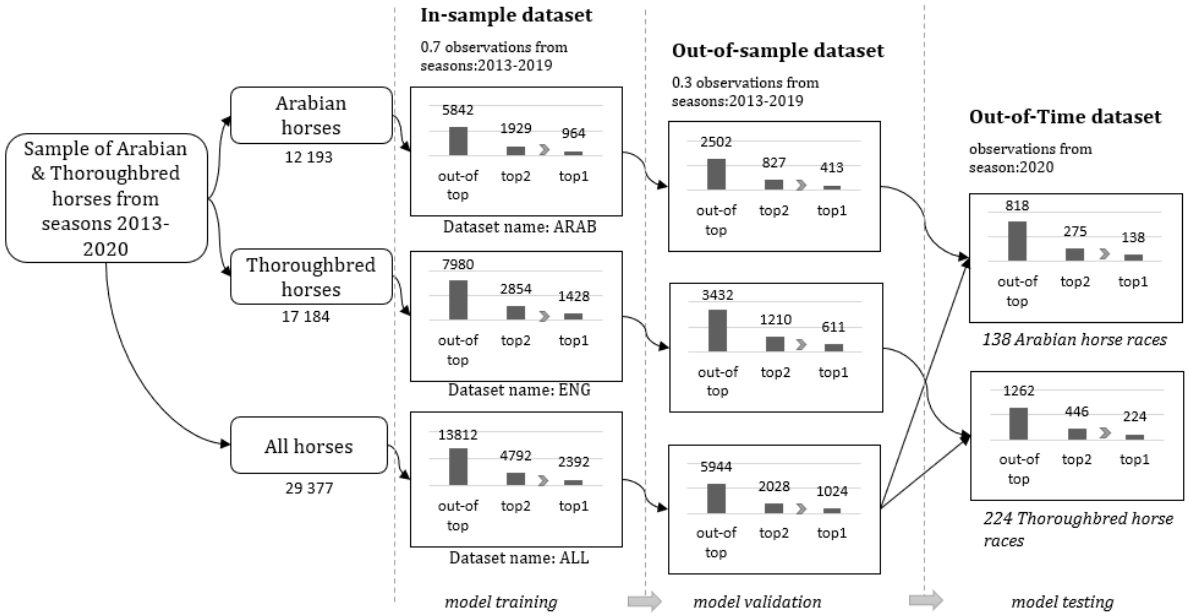
Predicting the score of the single horse differs from predicting the whole race. Let us take the example of a race of seven Arabian horses. When forecasting, e.g., the Win bet, we want only one participant to be selected. However, there could be a situation that our model classified more possible winners (two, three or even seven competitors). For this reason, we analysed two different strategies. In the first, we selected a horse with a higher probability score and placed a bet on him when it was a Win bet, and in the case of Quinella, we would choose two such horses. The second system was to place the bet only if the model indicated one horse equally for the Win and two horses for the Quinella. To keep the transparency and not complicate the article unnecessarily, we presented only the results for a more profitable strategy.

2.2. Data

The data used in the study represent horse racing performance for years 2011-2020 and came from the three largest racetracks in Poland: "Tor Służewiec" - Warsaw, "Tor Partynice" – Wroclaw and "Hipodrom Sopot" – Sopot. They were automatically gathered using web scraping from the website koniewyscigowe.pl. The seasons 2011 and 2012 were used to create the horses' history, while 2013-2020 were used for the model training, validation and testing. Nobody before us has researched such a long history of horse racing results.

The necessary principles limited the data. Firstly, we chose only Arabian and Thoroughbred horses. Importantly, these two breeds are not racing against each other. Next, we limited categories to groups I, II, III, IV, A, B and LA, corresponding to Poland's most common flat horse racing. Achievements outside these groups were not considered. The maximal race distance was set to 3000 metres because long-distance racing was rare and applied mainly to special events. We also omitted debuting horses, so those had not had any racing career before. In the end, we obtained 29,377 observations of the horses from 3,782 races. On the graph below, we presented the data division process in the subsequent stages of the study.

Figure 1. Study process and data distribution



Source: Own preparation.

Collected data created a sample that we divided into two groups corresponding to Arabian and Thoroughbred horses. Because we wanted to check whether models built on the whole dataset resulted in better performance than those built on a particular breed, we kept the dataset composed of All horses. We decided to use the standard approach of statistical model validation. Years 2013-2019 with the split 70% - 30% were used for training and validation purposes as the in-sample (dataset names: ARAB, ENG and ALL for Arabian, Thoroughbred, and all horses respectively) and out-of-sample datasets, while the season 2020 was used for models testing as the out-of-time dataset.

2.3. Variables design

Unlike similar studies (Wiliams and Li, 2008; Shumaker, 2013), we did not just focus on horses' characteristics to predict race result. When composing variables describing the starting horse, we considered horse's, jockey's, and even trainer's racing history. It was the first study to use such many factors.

Table 1 presents the description of variables where the values t0, t3, t5 (accordingly entire career, last three races and last five races) correspond to the period from which we determined the racing history. The dependent variable was top 1 for the Win bet and top 2 for the Quinella.

Table 1. Variables description

Horse characteristics		Jockey characteristics		Trainer characteristics	Variables description
Horse's entire career	**Last 3 horse's races**	**Jockey's entire career**	**Last 5 jockey's races**	**Trainer's entire career**	
Characteristics					
horse_t0_prestige_max	horse_t3_prestige_max	jockey_t0_prestige_max	jockey_t5_prestige_max	-	*maximum prize pool to be won in races*
horse_t0_prestige_median	horse_t3_prestige_median	jockey_t0_prestige_median	jockey_t5_prestige_median	-	*median prize pool to be won in races*
horse_t0_prestige_mean	horse_t3_prestige_mean	jockey_t0_prestige_mean	jockey_t5_prestige_mean	-	*mean prize pool to be won in races*
horse_t0_win_max	horse_t3_win_max	jockey_t0_win_max	jockey_t5_win_max	-	*maximum of prizes won (PLN)*
horse_t0_win_sum	horse_t3_win_sum	jockey_t0_win_sum	jockey_t5_win_sum	trainer_t0_win_sum	*sum of prizes won (PLN)*
horse_t0_win_median	horse_t3_win_median	jockey_t0_win_median	jockey_t5_win_median	-	*median of prizes won (PLN)*
horse_t0_win_mean	horse_t3_win_mean	jockey_t0_win_mean	jockey_t5_win_mean	-	*average of prizes won (PLN)*
horse_t0_speed_max	-	-	-	-	*maximal speed (km/h)*
horse_t0_speed_mean	-	-	-	-	*average speed (km/h)*
horse_t0_position_median	horse_t3_position_median	jockey_t0_position_median	jockey_t5_position_median	trainer_t0_position_median	*median place in the race*
horse_t0_position_mean	horse_t3_position_mean	jockey_t0_position_mean	-	trainer_t0_position_mean	*average place in the race*
horse_t0_number_in_top3	horse_t3_number_in_top3	jockey_t0_percent_in_top3	jockey_t5_percent_in_top3	trainer_t0_percent_in_top3	*number of times in the top3*
horse_t0_number_in_top2	horse_t3_number_in_top2	jockey_t0_percent_in_top2	jockey_t5_percent_in_top2	trainer_t0_percent_in_top2	*number of times in the top2*
horse_t0_number_in_top1	horse_t3_number_in_top1	jockey_t0_percent_in_top1	jockey_t5_percent_in_top1	trainer_t0_percent_in_top1	*number of times in the top1*
horse_t0_percent_in_top3	-	-	-		*% in top3*
horse_t0_percent_in_top2	-	-	-	-	*% in top2*
horse_t0_percent_in_top1	-	-	-	-	*% in top1*
horse_t0_no_of_appearances	-	jockey_t0_no_of_appearances	-	trainer_t0_no_of_appearances	*number of starts*
horse_t0_style_sum	horse_t3_style_sum	-	-	-	*The sum for style, which is awarded as one for each race the horse easily wins*
Binary variables					
is_stallion					*is the horse a stallion?*
is_gelding					*is the horse a gelding?*
top1*					*did the horse finish in the top 1?*
top2*					*did the horse finish in the top 2?*

Source: Own preparation.

Presented variables described the observed horse but did not contain information on how it performed compared to other horses in the same race. Imagine two races. In Sopot, there were starting Arabian horses and won horse A, whose median of prizes won in the last three races (horse_t3_win_median) was 20,000 PLN. In Warsaw, there was a great end of season event for the best horses, and the median of prizes won in the last three races (horse_t3_win_median) for each participant exceeded 100,000 PLN, so even the horse that finished in the last place had it higher than the horse A from Sopot. For this reason, the predicted race result from these values may not have been reliable. To solve this problem, we created new variables by using standardisation with the below formula. All of the variables except those in the section' binary variables' were transformed this way.

$$X_z = \frac{X_i - \bar{X}}{X_{sd}} \quad (2)$$

The new variables had a '_z' appended to the end. In such a case, the value of 0 meant that the given horse's variable did not differ from the average result in the given race. A value greater than zero meant that the given horse had better characteristics than the average participators in the race. Finally, a value lower than zero indicated the horse's characteristic was worse than the average. Due to a large number of variables, we did not put these standardised in table 1. However, they appeared in the results when examining their impact on the construction of the model.

2.4. Models

All models were built in R using the caret Package (Kuhn, 2020), which contains functions to streamline the model training process for complex regression and classification problems. Every model was trained on each of the in-sample datasets called: ARAB, ENG, ALL corresponding to Arabian, Thoroughbred and all horses giving us 42 models total. We applied three times repeated cross-validation with five folds under the specific tuning parameters. A random search procedure was applied to find them, and the TuneLength parameter that denotes the amount of granularity in the tuning parameter grid was set to 100. Moreover, all models were optimised to ROC measure. In table 2, we presented calculated hyperparameters, and in further subsections, we described applied algorithms.

Table 2. Models created within caret Package

Models	built on	R Package	hyperparameters	values
top1				
C5.0	*ALL*	C5.0	trials, model, winnow	99, 2, FALSE
C5.0	*ANG*	C5.0	trials, model, winnow	98, 1, FALSE
C5.0	*ARAB*	C5.0	trials, model, winnow	98, 2, FALSE
GLMnet	*ALL*	glmnet	alpha, lambda	0.8127, 0.0021
GLMnet	*ANG*	glmnet	alpha, lambda	0.5489, 0.0023
GLMnet	*ARAB*	glmnet	alpha, lambda	0.9683, 0.0014
Linear Discriminant Analysis	*ALL*	lda	parameter	1
Linear Discriminant Analysis	*ANG*	lda	parameter	1
Linear Discriminant Analysis	*ARAB*	lda	parameter	1
Neural Network	*ALL*	nnet	size, decay	1, 4
Neural Network	*ANG*	nnet	size, decay	1, 4
Neural Network	*ARAB*	nnet	size, decay	4, 4
Random Forest	*ALL*	rf	mtry	11
Random Forest	*ANG*	rf	mtry	11
Random Forest	*ARAB*	rf	mtry	11
Classification tree	*ALL*	rpart	cp	0.0008
Classification tree	*ANG*	rpart	cp	0.0012
Classification tree	*ARAB*	rpart	cp	0.0014
XGBoost	*ALL*	xgbTree	nrounds, max_depth, eta, gamma, colsample_bytree, min_child_weight, subsample	100, 9, 0.05, 1, 0.36, 94, 0.75
XGBoost	*ANG*	xgbTree	nrounds, max_depth, eta, gamma, colsample_bytree, min_child_weight, subsample	360, 13, 0.01, 1, 0.09375, 55, 0.75
XGBoost	*ARAB*	xgbTree	nrounds, max_depth, eta, gamma, colsample_bytree, min_child_weight, subsample	100, 5, 0.05, 1, 0.16, 39, 0.6
top2				
C5.0	*ALL*	C5.0	trials, model, winnow	32, 2, FALSE
C5.0	*ANG*	C5.0	trials, model, winnow	31, 1, FALSE
C5.0	*ARAB*	C5.0	trials, model, winnow	23, 2, FALSE
GLMnet	*ALL*	glmnet	alpha, lambda	0.3169, 0.0012
GLMnet	*ANG*	glmnet	alpha, lambda	0.6087, 0.0025
GLMnet	*ARAB*	glmnet	alpha, lambda	0.6269, 0.0010
Linear Discriminant Analysis	*ALL*	lda	parameter	1
Linear Discriminant Analysis	*ANG*	lda	parameter	1
Linear Discriminant Analysis	*ARAB*	lda	parameter	1
Neural Network	*ALL*	nnet	size, decay	8, 10
Neural Network	*ANG*	nnet	size, decay	10, 10
Neural Network	*ARAB*	nnet	size, decay	10, 10
Random Forest	*ALL*	rf	mtry	11
Random Forest	*ANG*	rf	mtry	11
Random Forest	*ARAB*	rf	mtry	11
Classification tree	*ALL*	rpart	cp	0.0045
Classification tree	*ANG*	rpart	cp	0.0013
Classification tree	*ARAB*	rpart	cp	0.0020
XGBoost	*ALL*	xgbTree	nrounds, max_depth, eta, gamma, colsample_bytree, min_child_weight, subsample	100, 9, 0.05, 1, 0.76, 94, 0.8
XGBoost	*ANG*	xgbTree	nrounds, max_depth, eta, gamma, colsample_bytree, min_child_weight, subsample	100, 7, 0.05, 1, 0.56, 55, 0.7
XGBoost	*ARAB*	xgbTree	nrounds, max_depth, eta, gamma, colsample_bytree, min_child_weight, subsample	300, 15, 0.01, 1, 0.16, 39, 0.75

[1] ARAB: in-sample dataset composed of Arabian horses
[2] ENG: in-sample dataset composed of Thoroughbred horses
[3] ALL: in-sample dataset composed of Arabian & Thoroughbred horses

Source: Own preparation

2.4.1 Classification tree

Classification and regression trees CART for short is a term introduced by Leo Breiman (1984) and refer to Decision Tree algorithms. The algorithms' presentation is a binary tree where each root node represents a single input variable (x) and a split point on that variable (Brownlee, 2016). The prediction is based on the most frequent value of the target variable in a given terminal node in the dataset. The prediction and interpretation are intuitive with the given model, which is one of the most significant advantages of this method. It is necessary to follow the roots to assign our observation to the appropriate class.

The CART algorithm uses the node purity measure – the Gini coefficient to determine the best split (Rokach L., Maimon O., 2009). It is calculated as $p^2 + q^2$ where p and q are probabilities of success and failure in the node, respectively. Its high value means that most

observations for a given node have the same value of the target variable when the low value means the ratio of observations in each node with different values of the target variable is closer to 50%. Once the tree is built, it may be huge and is likely to overfit the training set. The solution is pruning, which can reduce the level of complexity defined by the number of splits.

2.4.2 Random Forest

The random forest is a widely used tree-based classification model (Mohana, Reddy, Anisha and Murthy, 2021) and is a particular case of bagging. It means it is based on the iterative and repeated process of applying the same algorithm for different data subsamples. Random forest is characterised by strong learning ability, robustness, and feasibility of the hypothesis space (Ao et al., 2019). Its most significant advantage is that it deals well with missing values in data and maintains Accuracy in such a case (Kuhn & Johnson, 2013). The model also does not overfit the data.

On the other hand, although it consists of decision trees, it is a black-box model, and results are somewhat troublesome in interpretation (Donges, 2019). The analyst has low control of what model performs, and they are limited only to change values of two parameters: the number of predictors and the total number of trees.

2.4.3. XGBoost

XGBoost (eXtreme Gradient Boosting) is a scalable and effective tree boosting system. It is quicker and usually better than its ancestor, Gradient Boosting Machine (GBM) (Chen & He, 2020). The number of nodes can be dynamically changed, and nodes with lower weights might be pruned more heavily. Additional randomisation parameter reduces correlations among trees which increase precision. XGBoost offers implemented methods of parallelisation, which can substantially reduce computation time when compared with GBM. Thanks to regularisation, it can reduce the model overfitting effect.

The above reasons lead to the fact that XGBoost is a widely used algorithm by data scientists who provide state-of-the-art results for many problems (Chen & Guestrin, 2016).

2.4.4. GLMnet

John Nelder and Robert Wedderburn (1972) invented the generalised linear model approach to unify various other statistical models. One of the examples is GLMnet which fits a generalised linear model via penalised maximum likelihood. The algorithm is high-speed and efficient. It "uses cyclical coordinate descent, which successively optimises the objective function over

each parameter with others fixed, and cycles repeatedly until convergence" (Hastie, Qian, Tay, 2021).

It solves the following problem:

$$\min_{\beta_0,\beta} \frac{1}{N} \sum_{i=1}^{N} w_i l(y_i, \beta_0 + \beta^T x_i) + \lambda \left[(1-\alpha)||\beta||_2^2/2 + \alpha||\beta||_1 \right] \quad (3)$$

Where λ parameter controls the overall strength of the penalty l(y,η) is the negative log-likelihood contribution for observation, and α controls the elastic-net penalty.

2.4.5. Neural Networks

Neural networks are commonly used within sport prediction studies (Schumaker, 2013). They use forecasting methods that are based on simple mathematical models of the brain. Each neuron of the network is built of an element that aggregates products of weights and input signals and a non-linear transformation element of the neuron called the activation function (Guresen and Kayakutlu, 2011). The weights are determined depending on a particular training algorithm.

Neural networks are efficient and very powerful in dealing with non-linear relationships (Lek et al., 1996). They are also easy to apply parallelisation, which can significantly reduce computational time, and they can be modified with the new data without retraining on the whole data set. All these facts make them versatile and often used by scientists (Gevrey et al., 2003).

However, they meet also with disadvantages. A black-box effect makes the interpretation of relations among variables complex (Olden and Jackson, 2002). Another problem is connected with setting values of network parameters – the learning rate coefficient and the optimal number of hidden layers and neurons can be determined in most cases only by using the trial-and-error method (Heaton, 2017).

2.4.6 LDA

Linear Discriminant Analysis is a technique that is commonly used for supervised classification problems. (Tharwat, Gaber, Ibrahim and Hassanien, 2017). It assumes that each class's observations are drawn from a multivariate Gaussian distribution with a class-specific mean vector and a standard covariance matrix to all k classes. It transforms 'the features into a lower-dimensional space, which maximises the ratio of the between-class variance to the within-class variance, thereby guaranteeing maximum class separability' (Tharwat, Gaber, Ibrahim and Hassanien, 2017).

LDA easily handles the case where the within-class frequencies are unequal (Górecki and Łuczak, 2013). It performs well for facial and speech recognition or bankruptcy prediction. The disadvantages include a lack of stability when the classes are well-separated, or the training set contains a small amount of data (Sarkar, 2019).

2.5. Performance assessment

Having the models calculated, we had to determine their performance. We used the area under the ROC curve (AUC) and the area under the precision-recall curve (AUPRC) measures which were considered as a better measure than Accuracy (Huang & Ling, 2005). The net profit function and the return on investment with the custom measures, namely Expected ROI and Expected Profit, indicated the given model's profitability. In contrast, the Correct Bets Ratio indicated the model's precision. The measures were described in detail below.

2.5.1. AUC & AUPRC

The area under the Receiver Operating Characteristics (ROC) curve measure, for short AUC, is the plot presentation of true positive rate (also known as sensitivity or recall) against false positive rate (calculated as 1-specificity) at various threshold settings. The area under the ROC curve measures degree of discrimination between successes and failures. AUC of 1 denotes the model's perfect performance, while an AUC of 0.5 denotes the model's random classification (Huang & Ling, 2005).

There exists a similar measure called AUPRC, which is for the area under the precision-recall curve. It also indicates the model's performance, but the graph is drawn differently. On the x-axis, there is a true positive rate, and on the y-axis is precision which is the fraction of relevant instances among the retrieved instances (Ozenne, Subtil and Maucort-Boulch, 2015).

Some researchers (Saito, Rehmsmeier, 2015) say precision-recall plots are more informative than ROC plots when evaluating binary classifiers on imbalanced data. In such scenarios, ROC plots may be visually deceptive to conclusions about the classification performance.

2.5.2. Net profit function & optimal cutoff

Net profit determines the expected income made on betting the racetrack after deducting the price of the wager. Because we assumed four different win values proportional to the amount of the bet price, the net profit function had the following formula

$$\pi_i = w_i * number\ of\ correctly\ predicted\ bets - p * number\ of\ bets\ bought \quad (4)$$

where:

p - determines the price of the bet. It takes values depending on the result we want to predict.

$$if\ top\ 1\ (Win) \rightarrow p = 5PLN$$

$$if\ top\ 2\ (Quinella) \rightarrow p = 6PLN$$

i - is one of the four scenarios where the amount of payout for one bet is equal to p*1.5, p*2, p*3 or p*4,

w_i - determines the amount of payout for the given scenario.

We applied the above function to indicate the optimal cutoff point for each bet price level for every model. For this reason, we calculated the result of the function for a grid from 0.01 to 1 on the in-sample dataset. Next, the optimal cutoff point was chosen as the net profit function's point reaches its maximum.

Since the function result is net worth, we could calculate the return on investment (ROI). It is a popular profitability metric used to evaluate how well an investment has performed (Zamfir, Mariana, Manea et al., 2017). The formula was following

$$ROI = \frac{Net\ Profit\ Value}{sum\ of\ costs} * 100\% \quad (5)$$

A negative result for ROI means that the investment is unprofitable. The higher the result, the better.

We adapted the above measures and created Expected Profit and Expected ROI for our research. They helped us present the models' final performance assuming that probability for scenarios (win value = bet price x 1.5, win value = bet price x 2, win value = bet price x 3, win value = bet price x 4) equals 25% each. Expected profit was calculated as:

$$Expected\ \pi = 25\% * \sum_{i=1.5,2,3,4}^{4} \pi_i, \quad (6)$$

while Expected ROI was determined by the sum of the Profits for scenarios divided by the sum of total costs times 100%. We applied two measures because they do not present the profitability of the model in the same way. If we take into account the whole season of the racing, one player would have an unlimited budget, and spending 1000 PLN on bets could earn 400 PLN. His return on investment was 40%. The other player was interested only in the ROI measure, and even though he earned during the same season 300 PLN, his cost of the bets was 600 PLN which gave him 50% ROI. The measures used allowed us to evaluate both cases.

The last measure, Correct Bets Ratio, was calculated as the sum of correctly predicted bets for scenarios divided by the sum of bets bought. It indicated the models' precision.

2.6. Variable Importance

To answer what features of the starting horse influence the race's high place, we used the Variable Importance (VI) measure. It indicates how much the model's fit changed when a given variable was removed from the model and thus allows to find out variables with the highest impact on the dependent variable (Fisher, Rudin, Dominici, 2019).

Thanks to Variable Importance, we can discover the relationship's significance between the model's predictive maintenance and the traits without explaining the entire intra-model representation. It can be helpful in model simplification to exclude variables that do not influence predictions. The comparison of VI in different models may show interrelations between models' features. There is also the possibility of assessing the validity of the model and discovering new mechanisms by identifying the most critical variables (Biecek & Burzykowski, 2020).

In the research, we applied Variable Importance to the algorithms with the highest profitability for each bet. It was Neural Networks and LDA. The calculation was done using the caret Package (Kuhn, 2020), and it differs for those two algorithms. As a result, the values from the range 0-100 were received. They inform about the impact of a given variable on the model construction, and the higher value, the higher the Variable Importance.

In Neural Networks, the method of indicating Variable Importance was based on (Gevrey et al., 2003) and the weights approach. It is a simplification of the Garson (1991) method and "involves partitioning the hidden-output connection weights of each hidden layer".

For LDA, the 'filter' approach was made according to the instruction: "For classification, ROC curve analysis is conducted on each predictor. For two-class problems, a series of cutoffs is applied to the predictor data to predict the class. The sensitivity and specificity are computed for each cutoff, and the ROC curve is computed. The trapezoidal rule is used to compute the area under the ROC curve. This area is used as the measure of Variable Importance. (Kuhn, 2020).

3. Results

In this section, we presented the results of our study. We divided them into two parts to keep clarity, top 1 (Win) and top 2 (Quinella). We should remember that these are two separate studies or even four when considering different breed.

Firstly we analysed models performance based on the AUC and AUPRC measures. The summary table presents results for the model's training, validation, and testing process. We decided to highlight the result of out-of-sample datasets as they would be an excellent indicator for choosing the best algorithm for season 2021 predictions.

Next, we focused on models' profitability, which we showed in the corresponding table and chart. The Expected Profit value sorted algorithms in descending order. For each model, we calculated values for four possible win-value scenarios. Cutoff point divided the horses into the top and out horses. We observed that this value decreased with the increase of the possible gain. In the table, we had also posted information about how the return on investment value changed when the model missed 10% and 20% of true positives. It allowed showing the potential risk connected with the strategy in a given situation. Finally, we showed the impact of the features on the models by using Variable Importance.

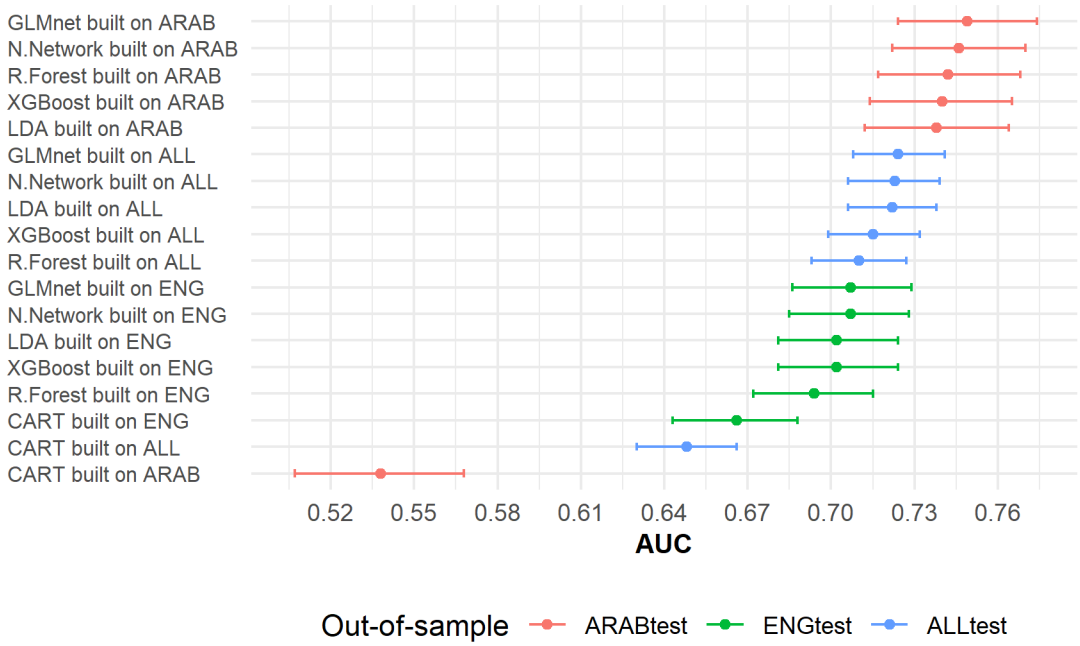
3.1 Top 1 – Win bet

In this study, we tried to predict whether a horse finished the race in the first position. It is one of the most popular bets, and it costs 5 PLN. The best results we obtain for the "highest probability" approach. If the model indicated more than one race winner, we chose the horse with the highest probability value and bet on him.

3.1.1 AUC comparison

Figure 2 presents the mean AUC with a 95% confidence interval for the validation dataset.

Figure 2. Top 1 model's performance on out-of-sample datasets



Note:
ARAB: In-sample dataset composed of Arabian horses
ENG: In-sample dataset composed of Thoroughbred horses
ALL: In-sample dataset composed of Arabian & Thoroughbred horses
ARABtest: Out-of-sample dataset composed of Arabian horses
ENGtest: Out-of-sample dataset composed of Thoroughbred horses
ALLtest: Out-of-sample dataset composed of Arabian & Thoroughbred horses

Source: Own preparation

If we were to bet on a horse in Arabian horse racing, we would consider models built on Arabian and all horses. In red, we marked models trained on Arabian horses. All of them, except the CART algorithm, got a score between 0.712 and 0.774. The highest value got GLMnet, Neural Network, Random Forest, XGBoost and LDA. We observed a higher variance for these models than for the models built on all horses. It indicated that the AUC value was more spread out from the mean, models behave less consistent. However, the differences between models built on Arabian and all horses seem insignificant as all of them (except CART) had their AUC mean value over 0.700, and their confidence intervals overlap.

The highest value of AUC for models built on all horses got GLMnet and then following Neural Networks, LDA, XGBoost, Random Forests. Their AUC value was between 0.672 and 0.729. CART algorithm got the worst result with AUC below 0.688.

In the case of Thoroughbred horse racing, we would consider models marked in green and blue. Models built on all training datasets performed slightly better, but the differences are not significant. The order for models built on Thoroughbred horses sorted by the highest score

was GLMnet, Neural Network, LDA, XGBoost, Random Forests and CART. When omitting the CART algorithm, AUC values for Thoroughbred horses were between 0.693 and 0.741.

Based on the above results, we saw a tendency for higher AUC for models based on neural network and GLMnet algorithms. CART, in all cases, performed the worst. Table 3 presents the results of the models' performance from the training, validation and testing process.

Table 3. Top1 Model's performance comparison, ordered by out-of-sample AUC

		In-sample					Out-of-sample					Out-of-time Thoroughbred horses dataset					Out-of-time Arabian horses dataset				
Model	BuiltOn	AUC	AUPRC	KStest	AUC_L	AUC_U	AUC	AUPRC	KStest	AUC_L	AUC_U	AUC	AUPRC	KStest	AUC_L	AUC_U	AUC	AUPRC	KStest	AUC_L	AUC_U
models trained on Arabian horses																					
GLMnet	ARAB	**0.742**	**0.305**	0.355	0.725	0.758	**0.749**	**0.294**	0.381	0.724	0.774	-	-	-	-	-	**0.711**	**0.229**	0.337	0.668	0.754
N.Network	ARAB	**0.774**	**0.361**	0.412	0.759	0.790	**0.746**	**0.301**	0.388	0.722	0.770	-	-	-	-	-	**0.704**	**0.221**	0.338	0.661	0.747
R.Forest	ARAB	**1.000**	**0.999**	1.000	1.000	1.000	**0.742**	**0.304**	0.365	0.717	0.768	-	-	-	-	-	**0.688**	**0.245**	0.294	0.640	0.736
XGBoost	ARAB	**0.924**	**0.734**	0.691	0.915	0.932	**0.740**	**0.301**	0.366	0.714	0.765	-	-	-	-	-	**0.708**	**0.237**	0.330	0.664	0.753
LDA	ARAB	**0.753**	**0.327**	0.368	0.737	0.769	**0.738**	**0.313**	0.358	0.712	0.764	-	-	-	-	-	**0.714**	**0.250**	0.321	0.672	0.757
CART	ARAB	**0.671**	**0.464**	0.312	0.654	0.689	**0.538**	**0.231**	0.186	0.507	0.568	-	-	-	-	-	**0.479**	**0.163**	0.109	0.433	0.525
models trained on Thoroughbred horses																					
GLMnet	ENG	**0.736**	**0.301**	0.336	0.722	0.749	**0.707**	**0.268**	0.309	0.686	0.729	**0.719**	**0.274**	0.365	0.684	0.755	-	-	-	-	-
N.Network	ENG	**0.741**	**0.300**	0.354	0.728	0.754	**0.707**	**0.267**	0.314	0.685	0.728	**0.723**	**0.282**	0.359	0.687	0.759	-	-	-	-	-
XGBoost	ENG	**0.885**	**0.636**	0.594	0.876	0.894	**0.702**	**0.267**	0.297	0.681	0.724	**0.709**	**0.268**	0.327	0.673	0.744	-	-	-	-	-
LDA	ENG	**0.739**	**0.308**	0.347	0.726	0.753	**0.702**	**0.257**	0.310	0.681	0.724	**0.714**	**0.260**	0.335	0.678	0.749	-	-	-	-	-
R.Forest	ENG	**1.000**	**0.999**	1.000	1.000	1.000	**0.694**	**0.245**	0.290	0.672	0.715	**0.714**	**0.309**	0.325	0.678	0.751	-	-	-	-	-
CART	ENG	**0.762**	**0.478**	0.372	0.747	0.776	**0.666**	**0.221**	0.247	0.643	0.688	**0.650**	**0.213**	0.240	0.612	0.688	-	-	-	-	-
models trained on all horses																					
GLMnet	ALL	**0.736**	**0.296**	0.345	0.725	0.746	**0.724**	**0.283**	0.346	0.708	0.741	**0.723**	**0.289**	0.369	0.688	0.759	**0.732**	**0.264**	0.367	0.690	0.774
N.Network	ALL	**0.740**	**0.300**	0.357	0.730	0.751	**0.723**	**0.278**	0.335	0.706	0.739	**0.722**	**0.289**	0.369	0.687	0.758	**0.733**	**0.266**	0.364	0.691	0.776
LDA	ALL	**0.737**	**0.299**	0.344	0.727	0.747	**0.722**	**0.275**	0.333	0.706	0.738	**0.721**	**0.283**	0.366	0.686	0.756	**0.729**	**0.288**	0.337	0.685	0.772
XGBoost	ALL	**0.845**	**0.536**	0.519	0.837	0.853	**0.715**	**0.267**	0.328	0.699	0.732	**0.713**	**0.271**	0.321	0.678	0.748	**0.712**	**0.244**	0.333	0.670	0.755
R.Forest	ALL	**1.000**	**1.000**	1.000	1.000	1.000	**0.710**	**0.273**	0.306	0.693	0.727	**0.702**	**0.292**	0.303	0.665	0.739	**0.678**	**0.230**	0.288	0.630	0.727
CART	ALL	**0.814**	**0.560**	0.481	0.803	0.824	**0.648**	**0.210**	0.245	0.630	0.666	**0.642**	**0.215**	0.235	0.603	0.681	**0.665**	**0.202**	0.286	0.617	0.714

[1] ARAB: in-sample dataset composed of Arabian horses
[2] ENG: in-sample dataset composed of Thoroughbred horses
[3] ALL: in-sample dataset composed of Arabian & Thoroughbred horses
[a] AUC: Area under ROC curve
[b] AUPRC: Area under Precision-Recall curve
[c] KStest: Kolmogorov-Smirnov test value
[d] AUC_L: Lower bound for the ROC values on test dataset
[e] AUC_U: Upper bound for the ROC values on test dataset

Source: Own preparation

It can be observed that models behave similarly on out-of-time datasets and out-of-sample in predicting the Thoroughbred and Arabian horse racing. The AUPRC measure generally coincided with AUC when indicating the order of the models.

3.1.2 The profitability of the models

In Table 5, we presented the profitability of the models. The bet's cost was set to 5 PLN, so the win values were 7.5 PLN, 10 PLN, 15 PLN and 20 PLN. The total number of top 1 places in the out-of-time dataset was 138 for Arabian horses and 224 for Thoroughbred.

Table 4. Comparison of models profitability for the TOP 1

TOP1 Models' profitability comparison on out-of-time datasets for bet price = 5 PLN

					Win Value = betPrice*1.5							Win Value = betPrice*2							Win Value = betPrice*3							Win Value = betPrice*4						
							no misses in number of correct bets				ROI when x% of correct bets missed				no misses in number of correct bets				ROI when x% of correct bets missed				no misses in number of correct bets				ROI when x% of correct bets missed		
Model	Built on	Correct Bets Ratio	Exp. Profit	Exp. ROI	Cutoff	% of correct bets	n. of bets	Profit	ROI	10%	20%	Cutoff	% of correct bets	n. of bets	Profit	ROI	10%	20%	Cutoff	% of correct bets	n. of bets	Profit	ROI	10%	20%	Cutoff	% of correct bets	n. of bets	Profit	ROI	10%	20%
Out-of-time Arabian horses dataset																															
LDA	ALL	41	85	50	0.96	0	0	0	**0**	-	-	0.65	100	1	5	**100**	-100	-100	0.39	44	41	65	**32**	17	2	0.28	39	94	270	**57**	40	23
N.Network	ALL	37	60	35	0.49	0	0	0	**0**	-	-	0.43	33	3	-5	**-33**	-100	-100	0.33	37	52	25	**10**	-2	-13	0.29	38	84	220	**52**	33	19
LDA	ARAB	34	41	22	0.72	33	3	-7.5	**-50**	-100	-100	0.66	40	5	-5	**-20**	-60	-60	0.39	39	44	35	**16**	2	-11	0.22	32	96	140	**29**	12	0
R.Forest	ARAB	33	34	15	0.56	0	0	0	**0**	-	-	0.55	0	0	0	**0**	-	-	0.31	34	86	5	**1**	-9	-20	0.3	32	90	130	**29**	16	2
XGBoost	ALL	33	29	19	0.6	0	0	0	**0**	-	-	0.44	55	11	5	**9**	-9	-27	0.44	55	11	35	**64**	36	9	0.25	29	101	75	**15**	3	-9
GLMnet	ALL	31	26	13	0.94	0	0	0	**0**	-	-	0.45	20	5	-15	**-60**	-100	-100	0.3	29	41	-25	**-12**	-27	-34	0.21	32	111	145	**26**	12	1
R.Forest	ALL	33	25	18	0.66	0	0	0	**0**	-	-	0.56	0	0	0	**0**	-	-	0.36	34	44	5	**2**	-11	-18	0.33	32	69	95	**28**	10	-1
N.Network	ARAB	26	-10	-7	0.59	0	0	0	**0**	-	-	0.42	38	8	-10	**-25**	-50	-50	0.38	26	23	-25	**-22**	-35	-48	0.26	25	85	-5	**-1**	-15	-25
CART	ALL	22	-16	-12	1	0	0	0	**0**	-	-	0.94	0	1	-5	**-100**	-100	-100	0.92	20	10	-20	**-40**	-70	-70	0.14	23	96	-40	**-8**	-21	-29
GLMnet	ARAB	25	-22	-11	0.94	0	0	0	**0**	-	-	0.4	11	9	-35	**-78**	-100	-100	0.28	27	49	-50	**-20**	-33	-39	0.2	25	109	-5	**-1**	-12	-23
CART	ARAB	23	-22	-15	0.96	0	0	0	**0**	-	-	0.96	0	0	0	**0**	-	-	0.72	29	28	-20	**-14**	-25	-36	0.1	21	86	-70	**-16**	-26	-35
XGBoost	ARAB	27	-30	-11	0.51	50	4	-5	**-25**	-62	-62	0.47	44	9	-5	**-11**	-33	-33	0.25	25	103	-125	**-24**	-33	-42	0.23	26	109	15	**3**	-8	-19
Out-of-time Thoroughbred horses dataset																															
N.Network	ENG	41	122	52	0.44	0	0	0	**0**	-	-	0.41	100	1	5	**100**	-100	-100	0.33	40	53	50	**19**	2	-9	0.27	41	133	435	**65**	47	32
XGBoost	ENG	39	116	39	0.47	100	3	7.5	**50**	0	0	0.47	100	3	15	**100**	33	33	0.32	40	73	70	**19**	7	-5	0.25	36	162	370	**46**	31	16
GLMnet	ENG	44	111	54	0.51	50	2	-2.5	**-25**	-100	-100	0.51	50	2	0	**0**	-100	-100	0.32	44	68	110	**32**	19	6	0.29	43	93	335	**72**	55	38
N.Network	ALL	40	100	39	0.49	0	0	0	**0**	-	-	0.43	57	7	5	**14**	-14	-14	0.33	39	79	70	**18**	3	-9	0.29	39	119	325	**55**	38	21
GLMnet	ALL	36	99	31	0.94	0	0	0	**0**	-	-	0.45	36	11	-15	**-27**	-45	-45	0.3	38	82	55	**13**	-1	-12	0.21	36	165	355	**43**	28	14
XGBoost	ALL	38	95	37	0.6	0	0	0	**0**	-	-	0.44	52	23	5	**4**	-13	-22	0.44	52	23	65	**56**	30	17	0.25	35	162	310	**38**	23	9
R.Forest	ENG	35	92	37	0.59	0	0	0	**0**	-	-	0.54	0	0	0	**0**	-	-	0.47	58	12	45	**75**	50	25	0.27	34	187	325	**35**	20	7
R.Forest	ALL	38	90	31	0.66	0	0	0	**0**	-	-	0.56	100	1	5	**100**	-100	-100	0.36	41	99	120	**24**	9	-3	0.33	34	129	235	**36**	21	9
LDA	ALL	38	80	33	0.96	0	0	0	**0**	-	-	0.65	57	7	5	**14**	-14	-14	0.39	39	61	55	**18**	3	-7	0.28	35	124	260	**42**	26	13
LDA	ENG	37	71	31	0.93	0	2	-10	**-100**	-100	-100	0.93	0	2	-10	**-100**	-100	-100	0.33	38	84	60	**14**	0	-11	0.32	38	95	245	**52**	35	18
CART	ALL	33	59	27	1	0	0	0	**0**	-	-	0.94	33	3	-5	**-33**	-100	-100	0.92	44	9	15	**33**	0	0	0.14	32	163	225	**28**	13	1
CART	ENG	33	55	33	1	0	0	0	**0**	-	-	1	0	0	0	**0**	-	-	1	0	0	0	**0**	-	-	0.17	33	132	220	**33**	18	6

[1] ARAB: in-sample dataset composed of Arabian horses

[2] ENG: in-sample dataset composed of Thoroughbred horses

[3] ALL: in-sample dataset composed of Arabian & Thoroughbred horses

[a] Cutoff: Optimal cutpoint determined by profit function on out-of-sample dataset for the given model

[b] % of correct bets: %; the number of correct bets divided by the number of bets times 100%

[c] n. of bets: number of all bets made for the given win value

[d] Correct Bets Ratio: %; the sum of correctly predicted bets for four scenarios divided by the sum of bets bought times 100%

[e] Exp. Profit: PLN; Expected Profit - the sum of the Profits for all scenarios divided by 4

[f] Exp. ROI: %; Expected return on investment - the sum of the Profits for all scenarios divided by the sum of total costs times 100%

[g] ROI when x% of correct bets missed: %; Return on investment calculated when x% of correct bets are for real False

Source: Own preparation.

The obtained values of the Correct Bets Ratio measure were between 27% and 41% Arabian and between 33% and 41% for Thoroughbred horses.

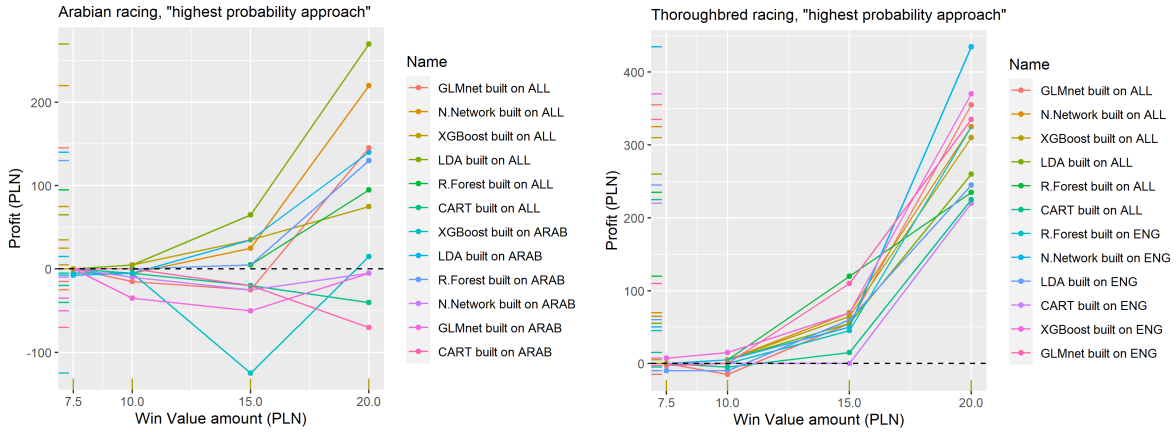
When predicting Arabian horse racing, the highest result got the LDA algorithm built on all horses with an expected profit of 85 PLN and Expected return on investment equal 50%. The next was also model based on all horses – Neural Networks with an expected profit of 60 PLN. Models built on an in-sample dataset composed only of Arabian horses performed quite worse, starting from the LDA with the Expected Profit equals 41 PLN and Expected ROI 22%.

We observed a small number of bets made for bet price 7.5 PLN and 10 PLN. More interesting result started from 15 PLN, for which by buying 41 out of 138 possible coupons, we could earn 65 PLN net revenue maximum. The ROI measure for LDA indicated 32%, and even after 20% of misses stayed on a positive level. For the remaining models, it took a negative value. For the level of 20 PLN and 20% of misses, only two models (all built on ALL) obtained ROI value above 2% - LDA and Neural Networks, both built on All data set.

When considering Thoroughbred horses, there was a similar situation in terms of the buying strategy. However, there were 224 possible winners, so models could not be compared straightforward. Algorithms behaved weak in terms of possible win value equal to 10 PLN or less. When sorted by Expected Profit, the highest result was retrieved for Neural Networks built on Thoroughbred horses with a score of 122 PLN. Next was XGBoost– 116 PLN, and GLMnet 111 PLN. The Expected ROI for them was over 38%. Models used to predict Thoroughbred horse racing behaved better than Arabian when considering win value 20 PLN and 20% miss rate. All of the algorithms got a similar score of ROI between 1% and 38%. We did not expect such high results for the prediction of the top 1 horse.

Figure 3 presents the profit value results from the table on the chart, so it was easier to see how the models behaved. Charts were created based on the calculated profit on an out-of-time dataset under the condition that the win value was constant and equal to one of four values 7.5 PLN, 10PLN, 15 PLN, 20 PLN.

Figure 3 Season 2021 profit under specific win value for Win bet



Source: Own preparation

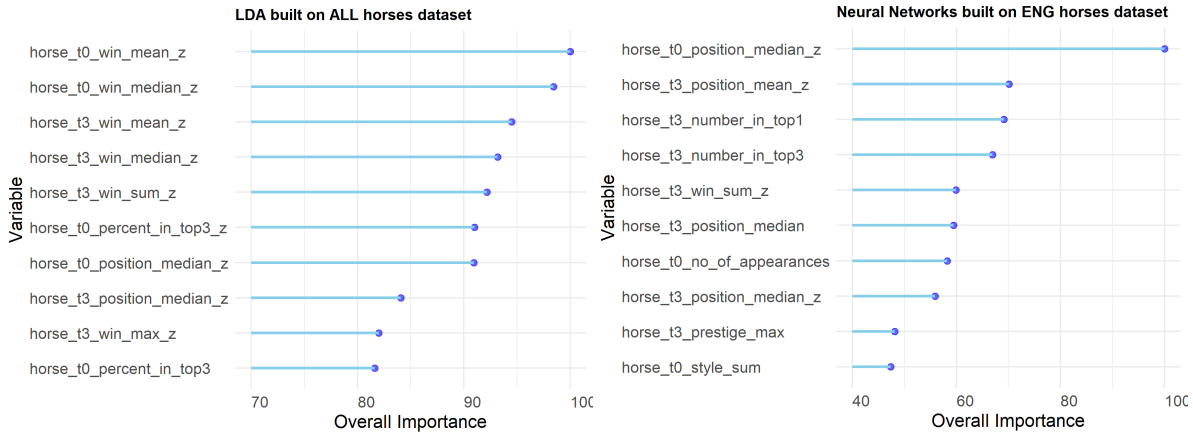
Based on the results, we would accept two models for the winner's predictions for Arabian horse racing; however, only under particular conditions. The win value should be above 20 PLN, so four times bigger than the bet price. It would be Neural Networks and LDA; both built on ALL dataset. They were least exposed to the risk of possible misses and gave the highest return on investment.

For Thoroughbred horse racing, we would apply the exact requirements. The best behaved GLMnet and Neural Network models; both built on the ENG dataset. Also, XGBoost built on ENG had a good result too. In both scenarios – Thoroughbred and Arabian horse racing the CART algorithm behaved the worst.

3.1.3 Variable Importance

We chose two models with the best performance (LDA built on a dataset composed of all horses and Neural Networks built on Thoroughbred horses dataset) to apply Variable Importance. Figure 4 presents ten variables for each of the model with the highest impact on their construction.

Figure 4. Variable Importance of top 1 models



Source: Own preparation

For the chosen models, the highest impact on its construction had standardised variables (_z append) indicating how much the given horse characteristic differs from the mean of this characteristic of all horses in a given race.

In the LDA model, the best score got variables specifying the value of the prizes won by the horse. The first two were the mean and median value from the whole horse career - next, the mean and median and sum from the last three races. A quite worse score got variables, indicating the horse's race position history. Successively variable informing about the percentage of how many times the horse finished in the top 3 places in his career, the median of the position in the whole racing carrier, and in the last three races. The last two variables were the maximum prize won in the last three races and not standardised characteristic informing about the ratio of the horse being in the top 3.

For Neural Networks, a variable indicating the median position of the horse in his whole career got a significantly higher result compared to other variables. Next were the variables specifying the last three horse races. Successively the mean position, the number of being in top 1 and top 3 places, the sum of won prizes and the median position. The number of appearances indicated how many times the horse took part in the race in his entire career, and prestige value indicated the maximum win prize from the last three races that could be won. The last variable specified how many times a given horse won easily (several lengths of the horse's body) in his racetrack history.

The interesting could be the fact that there were no variables describing jockey's or trainer's characteristics. We can assume they were not very important in creating the models predicting the top 1 horse. We observed that the variables describing in various ways the amount

of money won in the race were a good indicator of the horse's performance. The place in the past races, top 1 and top 3, seemed to be essential either.

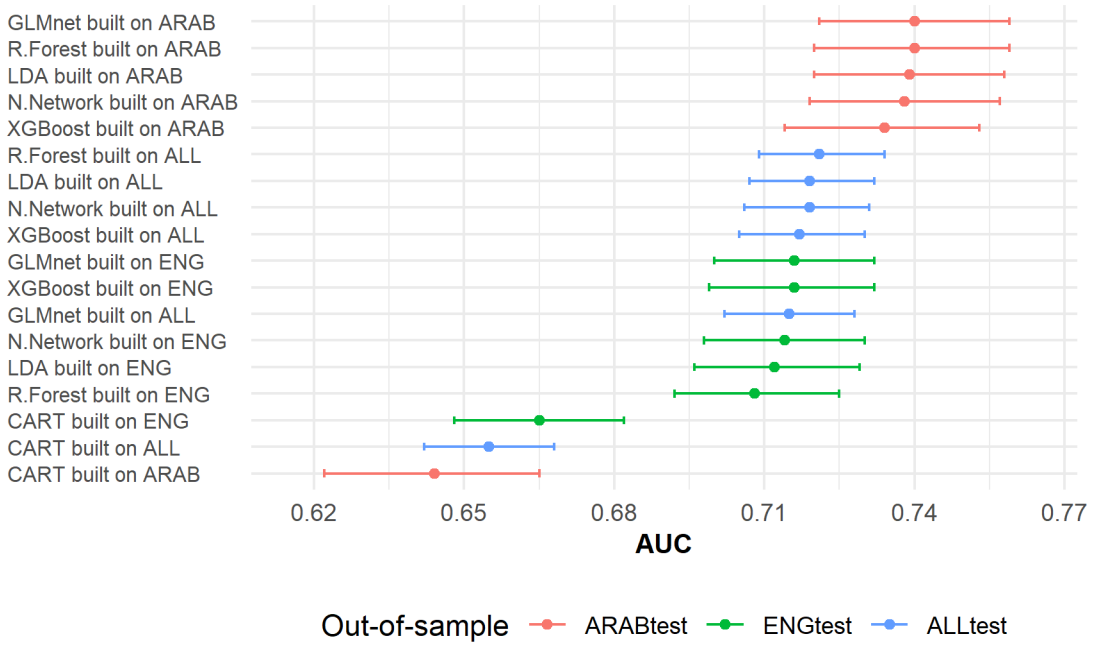
3.2 Top 2 – Quinella bet

The second object of our study was the Quinella bet, which was to predict the top 2 horses in any order. We assumed it cost 6 PLN. Unlike the Win bet, the best results were obtained for the exact number of horses approach. We would play only if the algorithm selected exactly two horses.

3.2.1 AUC comparison

Figure 5 presents the mean AUC with a 95% confidence interval for the validation dataset.

Figure 5. Top 2 model's performance on out-of-sample datasets



Note:
ARAB: In-sample dataset composed of Arabian horses
ENG: In-sample dataset composed of Thoroughbred horses
ALL: In-sample dataset composed of Arabian & Thoroughbred horses
ARABtest: Out-of-sample dataset composed of Arabian horses
ENGtest: Out-of-sample dataset composed of Thoroughbred horses
ALLtest: Out-of-sample dataset composed of Arabian & Thoroughbred horses

Source: Own preparation

Considering Arabian horses, models built on All horses got lower AUC values than those built on Arabian. However, the differences were not significant. The highest result got Random Forest and GLMnet, both with the mean AUC equal to 0.740. Next was LDA, Neural Networks and XGBoost. The values for models built on all horses, except CART, were between 0.702 and 0.734.

If we wanted to select the best algorithm for Thoroughbred horses, there was no significant difference between models built on ALL or ENG dataset. GLMnet, the model with the highest result from models built on the ENG dataset, had a mean AUC of 0.716. XGboost, Neural Networks, LDA, and Random Forests obtained AUC measure exceeding 0.692. Similar to Win bet, CART models performed the worst. In Table 5, we summarised the results of the models' performance for the Quinella bet.

Table 5. Top 2 Model's performance comparison, ordered by out-of-sample AUC

		In-sample					Out-of-sample					Out-of-time Thoroughbred horses dataset					Out-of-time Arabian horses dataset				
Model	BuiltOn	AUC	AUPRC	KStest	AUC_L	AUC_U	AUC	AUPRC	KStest	AUC_L	AUC_U	AUC	AUPRC	KStest	AUC_L	AUC_U	AUC	AUPRC	KStest	AUC_L	AUC_U
models trained on Arabian horses																					
R.Forest	ARAB	**1.000**	**0.999**	1.000	1.000	1.000	**0.740**	**0.482**	0.366	0.720	0.759	-	-	-	-	-	**0.676**	**0.416**	0.250	0.640	0.712
GLMnet	ARAB	**0.742**	**0.486**	0.359	0.730	0.755	**0.740**	**0.480**	0.360	0.721	0.759	-	-	-	-	-	**0.699**	**0.415**	0.302	0.664	0.734
LDA	ARAB	**0.749**	**0.499**	0.370	0.737	0.761	**0.739**	**0.484**	0.348	0.720	0.758	-	-	-	-	-	**0.701**	**0.430**	0.300	0.666	0.737
N.Network	ARAB	**0.750**	**0.498**	0.366	0.737	0.762	**0.738**	**0.480**	0.353	0.719	0.757	-	-	-	-	-	**0.686**	**0.412**	0.283	0.650	0.722
XGBoost	ARAB	**0.888**	**0.757**	0.606	0.879	0.896	**0.734**	**0.472**	0.348	0.714	0.753	-	-	-	-	-	**0.694**	**0.421**	0.292	0.659	0.728
CART	ARAB	**0.748**	**0.588**	0.393	0.734	0.762	**0.644**	**0.365**	0.221	0.622	0.665	-	-	-	-	-	**0.626**	**0.355**	0.221	0.588	0.665
models trained on Thoroughbred horses																					
GLMnet	ENG	**0.726**	**0.482**	0.331	0.716	0.737	**0.716**	**0.459**	0.334	0.700	0.732	**0.708**	**0.462**	0.309	0.681	0.735	-	-	-	-	-
XGBoost	ENG	**0.844**	**0.689**	0.521	0.835	0.852	**0.716**	**0.455**	0.328	0.699	0.732	**0.705**	**0.460**	0.288	0.677	0.732	-	-	-	-	-
N.Network	ENG	**0.739**	**0.504**	0.347	0.728	0.749	**0.714**	**0.455**	0.327	0.698	0.730	**0.700**	**0.443**	0.307	0.673	0.728	-	-	-	-	-
LDA	ENG	**0.729**	**0.486**	0.336	0.718	0.739	**0.712**	**0.452**	0.326	0.696	0.729	**0.707**	**0.443**	0.317	0.679	0.734	-	-	-	-	-
R.Forest	ENG	**1.000**	**1.000**	1.000	1.000	1.000	**0.708**	**0.449**	0.299	0.692	0.725	**0.703**	**0.437**	0.293	0.676	0.731	-	-	-	-	-
CART	ENG	**0.663**	**0.382**	0.263	0.652	0.674	**0.665**	**0.374**	0.287	0.648	0.682	**0.627**	**0.343**	0.202	0.599	0.656	-	-	-	-	-
models trained on all horses																					
R.Forest	ALL	**1.000**	**1.000**	1.000	1.000	1.000	**0.721**	**0.463**	0.327	0.709	0.734	**0.699**	**0.446**	0.280	0.671	0.726	**0.681**	**0.415**	0.263	0.645	0.717
N.Network	ALL	**0.740**	**0.498**	0.351	0.732	0.748	**0.719**	**0.457**	0.317	0.706	0.731	**0.709**	**0.456**	0.316	0.682	0.737	**0.700**	**0.401**	0.330	0.665	0.736
LDA	ALL	**0.730**	**0.478**	0.339	0.722	0.738	**0.719**	**0.453**	0.325	0.707	0.732	**0.707**	**0.446**	0.305	0.680	0.734	**0.709**	**0.446**	0.326	0.674	0.744
XGBoost	ALL	**0.811**	**0.627**	0.459	0.804	0.818	**0.717**	**0.458**	0.323	0.705	0.730	**0.722**	**0.473**	0.347	0.696	0.749	**0.698**	**0.427**	0.305	0.663	0.733
GLMnet	ALL	**0.722**	**0.463**	0.329	0.714	0.730	**0.715**	**0.451**	0.318	0.702	0.728	**0.703**	**0.448**	0.300	0.676	0.731	**0.703**	**0.424**	0.315	0.668	0.738
CART	ALL	**0.657**	**0.428**	0.275	0.648	0.665	**0.655**	**0.409**	0.274	0.642	0.668	**0.627**	**0.380**	0.226	0.599	0.655	**0.599**	**0.349**	0.167	0.563	0.634

[1] ARAB: in-sample dataset composed of Arabian horses
[2] ENG: in-sample dataset composed of Thoroughbred horses
[3] ALL: in-sample dataset composed of Arabian & Thoroughbred horses
[a] AUC: Area under ROC curve
[b] AUPRC: Area under Precision-Recall curve
[c] KStest: Kolmogorov-Smirnov test value
[d] AUC_L: Lower bound for the ROC values on test dataset
[e] AUC_U: Upper bound for the ROC values on test dataset

Source: Own preparation

We did not notice irregularities between tests performed on the out-of-sample and out-of-time dataset. The AUPRC measure got relatively higher results than in the case of the Win bet.

3.2.2 The profitability of the models

The profitability of the models for the Quinella bet was presented in table 6. There were 275 top 2 places overall out of 138 Arabian races and 446 top 2 places out of 224 Thoroughbred races. We assumed win values as 9 PLN, 12 PLN, 18 PLN and 24 PLN.

Table 6. Comparison of models profitability for the TOP 2

TOP2 Models' profitability comparison on out-of-time datasets for bet price = 6 PLN

					Win Value = betPrice*1.5							Win Value = betPrice*2							Win Value = betPrice*3							Win Value = betPrice*4						
						no misses in number of correct bets				ROI when x% of correct bets missed			no misses in number of correct bets				ROI when x% of correct bets missed			no misses in number of correct bets				ROI when x% of correct bets missed			no misses in number of correct bets				ROI when x% of correct bets missed	
Model	Built on	Correct Bets Ratio	Exp. Profit	Exp. ROI	Cutoff	% of correct bets	n. of bets	Profit	ROI	10%	20%	Cutoff	% of correct bets	n. of bets	Profit	ROI	10%	20%	Cutoff	% of correct bets	n. of bets	Profit	ROI	10%	20%	Cutoff	% of correct bets	n. of bets	Profit	ROI	10%	20%
Out-of-time Arabian horses dataset																															
LDA	ALL	36	30	22	0.99	0	0	0	**0**	-	-	0.55	40	5	-6	**-20**	-60	-60	0.35	31	52	-24	**-8**	-19	-31	0.24	43	35	150	**71**	49	37
N.Network	ALL	32	22	16	0.61	0	0	0	**0**	-	-	0.5	0	5	-30	**-100**	-100	-100	0.36	18	56	-156	**-46**	-52	-57	0.22	63	30	276	**153**	127	100
R.Forest	ARAB	31	8	5	0.63	0	0	0	**0**	-	-	0.49	22	9	-30	**-56**	-78	-78	0.35	25	52	-78	**-25**	-37	-42	0.27	41	37	138	**62**	41	30
GLMnet	ALL	27	-4	-2	1	0	0	0	**0**	-	-	0.51	17	6	-24	**-67**	-100	-100	0.34	20	51	-126	**-41**	-47	-53	0.28	36	50	132	**44**	28	12
N.Network	ARAB	27	-9	-7	0.58	0	0	0	**0**	-	-	0.51	30	10	-24	**-40**	-60	-60	0.39	18	44	-120	**-46**	-52	-59	0.27	38	34	108	**53**	29	18
CART	ARAB	25	-9	-7	0.91	0	0	0	**0**	-	-	0.91	0	0	0	**0**	-	-	0.24	13	45	-162	**-60**	-67	-73	0.18	38	39	126	**54**	33	23
LDA	ARAB	29	-14	-8	0.67	33	3	-9	**-50**	-100	-100	0.5	30	20	-48	**-40**	-50	-60	0.34	26	47	-66	**-23**	-36	-43	0.22	31	45	66	**24**	7	-2
XGBoost	ALL	28	-14	-9	0.7	0	0	0	**0**	-	-	0.49	25	16	-48	**-50**	-62	-62	0.34	25	52	-78	**-25**	-37	-42	0.26	32	40	72	**30**	10	0
R.Forest	ALL	23	-22	-15	0.75	0	0	0	**0**	-	-	0.55	0	3	-18	**-100**	-100	-100	0.36	18	51	-144	**-47**	-53	-59	0.29	32	44	72	**27**	9	0
GLMnet	ARAB	25	-24	-16	0.62	0	1	-6	**-100**	-100	-100	0.5	38	8	-12	**-25**	-50	-50	0.35	19	47	-120	**-43**	-49	-55	0.26	29	45	42	**16**	-2	-11
XGBoost	ARAB	22	-51	-26	0.57	0	0	0	**0**	-	-	0.47	15	13	-54	**-69**	-85	-85	0.3	22	58	-114	**-33**	-43	-48	0.3	22	58	-36	**-10**	-24	-31
CART	ALL	15	-54	-42	0.57	0	0	0	**0**	-	-	0.43	8	12	-60	**-83**	-100	-100	0.33	3	33	-180	**-91**	-100	-100	0.18	28	40	24	**10**	-10	-20
Out-of-time Thoroughbred horses dataset																															
N.Network	ALL	37	86	32	0.61	0	0	0	**0**	-	-	0.5	12	17	-78	**-76**	-88	-88	0.36	28	94	-96	**-17**	-27	-36	0.22	58	66	516	**130**	106	82
R.Forest	ALL	34	40	18	0.75	0	0	0	**0**	-	-	0.55	17	6	-24	**-67**	-100	-100	0.36	31	88	-42	**-8**	-18	-28	0.29	41	58	228	**66**	45	31
XGBoost	ENG	33	33	13	0.71	0	0	0	**0**	-	-	0.49	22	23	-78	**-56**	-65	-65	0.33	26	90	-126	**-23**	-33	-40	0.24	48	60	336	**93**	73	53
LDA	ENG	33	32	14	0.86	0	1	-6	**-100**	-100	-100	0.56	36	11	-18	**-27**	-45	-45	0.31	23	86	-156	**-30**	-37	-44	0.22	47	57	306	**90**	68	47
CART	ENG	34	28	14	1	0	0	0	**0**	-	-	0.73	18	22	-84	**-64**	-73	-73	0.17	37	57	36	**10**	-5	-16	0.17	37	57	162	**47**	26	12
LDA	ALL	32	18	8	0.99	0	1	-6	**-100**	-100	-100	0.55	41	17	-18	**-18**	-29	-41	0.35	25	72	-108	**-25**	-33	-42	0.24	37	70	204	**49**	31	14
GLMnet	ENG	32	16	8	0.7	0	0	0	**0**	-	-	0.53	100	2	12	**100**	0	0	0.27	28	75	-72	**-16**	-28	-36	0.25	35	55	126	**38**	24	9
XGBoost	ALL	29	-8	-3	0.7	0	0	0	**0**	-	-	0.49	13	23	-102	**-74**	-83	-83	0.34	28	82	-78	**-16**	-27	-34	0.26	35	63	150	**40**	21	8
GLMnet	ALL	28	-18	-7	1	0	0	0	**0**	-	-	0.51	40	15	-18	**-20**	-33	-47	0.34	29	82	-60	**-12**	-23	-30	0.28	25	75	6	**1**	-9	-20
R.Forest	ENG	25	-24	-11	0.8	0	0	0	**0**	-	-	0.54	0	8	-48	**-100**	-100	-100	0.35	20	81	-198	**-41**	-48	-56	0.3	35	63	150	**40**	21	8
N.Network	ENG	24	-57	-20	0.69	0	0	0	**0**	-	-	0.47	17	24	-96	**-67**	-75	-75	0.31	22	87	-180	**-34**	-41	-48	0.25	28	76	48	**10**	-5	-16
CART	ALL	15	-96	-44	0.57	0	0	0	**0**	-	-	0.43	9	22	-108	**-82**	-91	-91	0.33	8	49	-222	**-76**	-82	-82	0.18	22	73	-54	**-12**	-23	-34

[1] ARAB: in-sample dataset composed of Arabian horses

[2] ENG: in-sample dataset composed of Thoroughbred horses

[3] ALL: in-sample dataset composed of Arabian & Thoroughbred horses

[a] Cutoff: Optimal cutpoint determined by profit function on out-of-sample dataset for the given model

[b] % of correct bets: %; the number of correct bets divided by the number of bets times 100%

[c] n. of bets: number of all bets made for the given win value

[d] Correct Bets Ratio: %; the sum of correctly predicted bets for four scenarios divided by the sum of bets bought times 100%

[e] Exp. Profit: PLN; Expected Profit - the sum of the Profits for all scenarios divided by 4

[f] Exp. ROI: %; Expected return on investment - the sum of the Profits for all scenarios divided by the sum of total costs times 100%

[g] ROI when x% of correct bets missed: %; Return on investment calculated when x% of correct bets are for real False

Source: Own preparation.

The result of expected Profit and Expected ROI was significantly worse if compared to the Win bet. The Corrects Bets Ratio was between 15% and 37% for Arabian horses and between 15% and 36% for Thoroughbred horses.

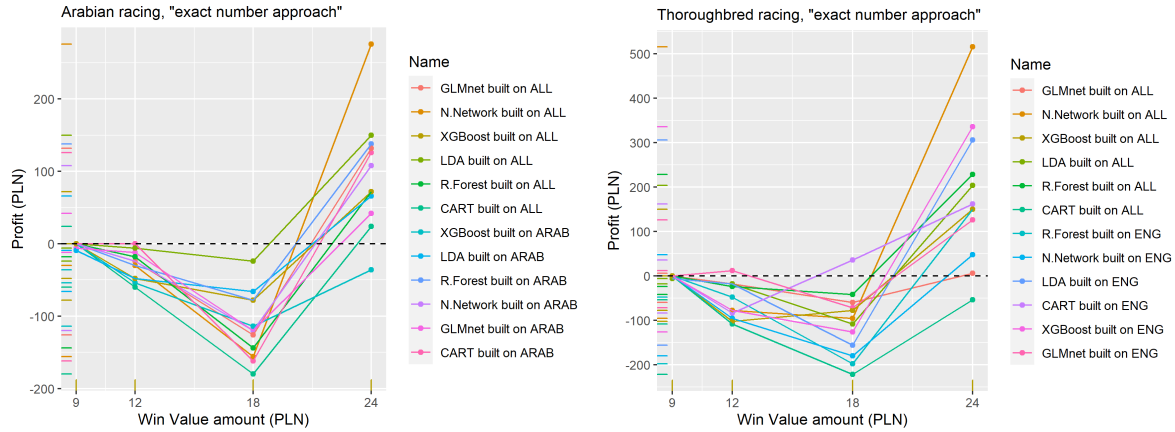
Only three algorithms gain positive values when predicting Arabian horse racing: LDA, Neural Networks, and Random Forest. The best results were obtained for the LDA model built on ALL dataset. Its Expected Profit was 30 PLN and expected ROI 22%. Next was the Neural Networks model, also built on ALL, with an Expected Profit of 22 PLN and ROI 16%. The Random Forest built on the ARAB dataset had only an Expected ROI of 5% and an Expected Profit of 8 PLN.

We observed that all of the models under the condition of win value lower than 18 PLN were unprofitable. It changes for the high win values equal to 24 PLN. In this case, only XGBoost built on the ARAB dataset did not get a positive ROI result. The best result was obtained for the Neural Networks built on ALL, which outperformed other models over two times with the ROI equals 153%, and even after 20% of bet missed, ROI equals 100%.

In the case of Thoroughbred horses, the maximum Expected ROI was 32% for Neural Networks built on ALL dataset. This model also obtained an Expected Profit equals 86 PLN. The results showed it was unprofitable to bet when the win value for the bet was 18 PLN or lower. However, for the level of 24 PLN, only the CART model built on ALL dataset resulted in negative profit. The best behaved the Neural Networks model built on ALL dataset profit 516 PLN and ROI 130% (and 82% after 20% bets missed).

Figure 6 presents the graphical representation of the models' profitability comparison. It was created in the same way as for the Win value but with different levels - 9 PLN, 12 PLN, 18 PLN and 24 PLN.

Figure 6. Season 2021 profit under specific win value for Quinella bet



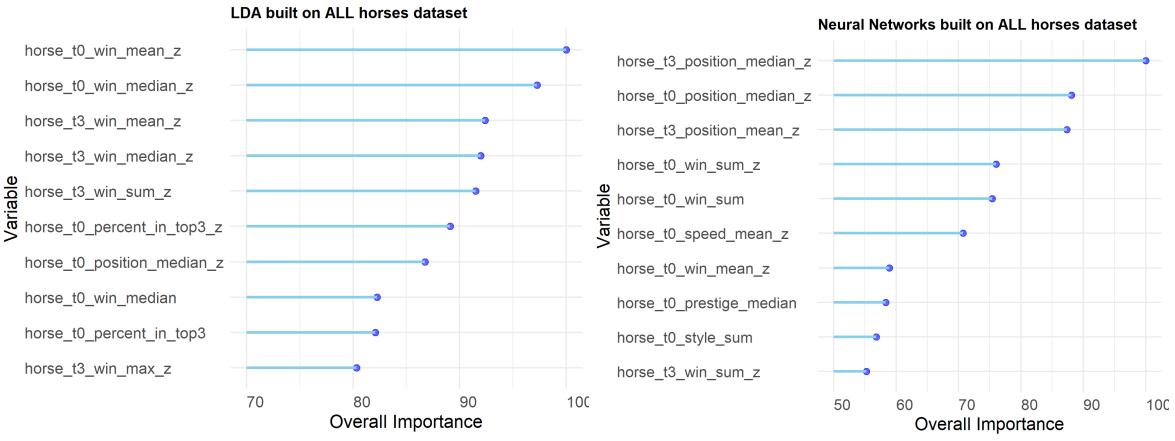
Source: Own preparation

We saw that it was unprofitable to place a bet when the win value was 18 PLN or lower. For the level of 24 PLN, so four times bigger than the bet price, many models obtained favourable results for both Arabian and Thoroughbred horse racing even if 20% of bets missed. Neural Networks built on all horses outperformed the rest of the models for Arabian and Thoroughbred horses in such a case.

3.2.3 Variable Importance

Two models were chosen to calculate Variable Importance. It was LDA and Neural Networks, and both built on all horses dataset. In figure 7, we presented the results for ten variables with the highest score.

Figure 7. Variable Importance of top 2 models



Source: Own preparation

In the case of LDA, the nine variables were identical as for the Win bet. Instead of the variable informing about the median position in the last three races, the variable indicating the median

of the horse's won prizes in his entire career was selected. The first seven variables were in the same order. We suppose that two reasons caused such results. Firstly, predicting whether a given horse will win the race or be in the second position was characterised by very similar determinants. Secondly, because the dataset used (ALL) was the same for both models.

The chosen Neural Networks model was built on all horses dataset for Quinella bet, while for the Win on Thoroughbred. The highest score was obtained for the variables specifying the position of the horse in the last races. Following were variables informing about the sum of the winning prizes. Compared to the Win bet, one kind of new variables appeared. The mean speed indicated the average speed of the horse in all track performances compared to the other horse starting. Again, no variables for characteristics other than the horse's were observed.

4. Summary

Hausch, Ziemba, and Rubinstein (1981) analysed the horse betting market and found that inefficiencies exist. Asch et al. (1984) and Bird and McCrae (1987) made similar conclusions. However, they claimed the profit could be made, but probably not on a significant scale. Gabriel and Marsden (1990) finally proved the phenomenon by comparing starting price bets with bookmakers and totalizator from England. Some of the researchers applied mathematic methods to bet effectively. Harville (1973) proposed the ranking model that was simple and commonly used, but because of the bias, it was later improved by Lo, Bacon-Shone and Bushe (1995). Researchers from the University of Mauritius (Pudaruth, Medard and Dookhun, 2013) decided to use the weighted probabilistic approach. Finally, scientists applied data mining methods, such as Neural Networks (Wiliams, Li, 2008; Schumaker, Johnson, 2008) or Support Vector Regression (Schumaker, 2013).

In this study, flat racing data was explored with six different machine learning algorithms to create a profitable betting system. Analysis was conducted using Polish racetrack data from the three largest hippodromes located in Warsaw, Wroclaw and Sopot. The racing history covers the years 2011-2020 with Arabian and Thoroughbred 3,782 races. For the first time, the dataset consisted of 128 variables describing horse's, jockey's, and even trainer's characteristics. The research aimed to check two betting strategies and two bets: the Win (bet were considered one horse that wins), the Quinella (select the first two finishers in a horse race in any order), and whether it was possible to profit from them. Additionally, we wanted to answer the following questions. Which machine learning method performed the best for Arabian and Thoroughbred horses? How should a betting strategy look like, and how much

could we earn by using them? Finally, we applied Variable Importance to check which features of the starting horse influence the race's high place. For these reasons, predictions were made using classification algorithms, namely Classification and Regression Tree (CART), Generalized Linear Model (Glmnet), Extreme Gradient Boosting (XGBoost), Random Forest (RF), Neural Network (NN) and Linear Discriminant Analysis (LDA). AUC and AUPRC with the comparison of adapted Return on Investment, Net Profit Function and Correct bets Ratio measures were used to specify their performance. We decided to use three steps approach of statistical model validation with the split 70% - 30% on in-sample (training) and out-of-sample (validation) datasets and the season 2020 as the out-of-time (testing) dataset.

For the Win bet, the best performed 'the highest probability' strategy, in which in case the model selected more horses than one for the winner, we would choose one with the highest probability. For Arabian horses, the best was the LDA model built on a dataset composed of all horses with the AUC equals 0.729 and an average Return on Investment of 50%. Its Correct bets Ratio, which represents the precision of the model, was 41%. Neural Networks models built on only Thoroughbred horses were the most profitable for the same bet and Thoroughbred horses. It obtained a score of 0.723 for the AUC measure and 52% on average of Return on Investment. Its Correct Bets Ratio was 41%. The best results were obtained in Quinella bet for the 'exact number of selected horses approach. However, they were quite worse when compared to the Win bet. Best performed models built on all horses – the LDA for Arabian and Neural Networks for the Thoroughbred horses. The first got AUC equals 0.709 on the test dataset, and Neural Networks got 0.700. The average Return on Investment was 22% and 32% successively, and the Correct Bets Ratio 36% and 37%.

Our observations showed that bets became profitable above a given win value level. For the Win bet, it was set as the three times multiplication of the bet price. It was then possible to obtain a 32% return on investment even with 20% of bets missed. In the Quinella bet, the level was set as the fourth time multiplication of the bet price. Using the same criteria as for the Win bet we could gain 100% of invested capital. We should remember that the real win value for the Quinella often exceeds the values given here, and a noticeable profit was already at the level of PLN 24. In addition, in many races, it was possible to buy a Quinella bet costing 3PLN, and the worst-case scenario we have adopted required us to buy two Exacta plants for a total of PLN 6. It seems to us that the reason for such a good prediction of models results from the amount of test data we had to train. We suppose that the results between the models would differ more

with fewer of them. At the same time, the results of these models suggest that the horse racing market in Poland is ineffective.

The Variable Importance showed that the main factor in constructing the best models was information about the amount of previous prizes won of a given horse. Surprisingly, it turned out that the jockey's and trainer's characteristics did not have such a significant impact and did not appear in the results at all.

The study confirmed that it was possible to create a profitable betting system on horse racing in Poland. We believe that it is an excellent basis for further considerations on this topic. Future work on this matter should include an analysis of the comparison between machine algorithms and the odds made by the bettors. Also, the number of variables should be limited only to those with the highest impact on the final view of the model. The quoted literature and this study show that there is still a lot to be discovered in this scientific field.

References

Ali, M., 1977. Probability and Utility Estimates for Racetrack Bettors. Journal of Political Economy, 85(4), pp.803-815.

Ao, Y., Li, H., Zhu, L., Ali, S. and Yang, Z., 2019. The linear random forest algorithm and its advantages in machine learning assisted logging regression modeling. Journal of Petroleum Science and Engineering, 174, pp.776-789.

Asch, P., Malkiel, B. and Quandt, R., 1984. Market Efficiency in Racetrack Betting. The Journal of Business, 57(2), p.165.

Balakrishnama, A. Ganapathiraju; Institute for Signal and Information Processing Department of Electrical and Computer Engineering Mississippi State University, [Online] Available at: http://www.music.mcgill.ca/~ich/classes/mumt611_07/classifiers/lda_theory.pdf [Accessed 20 5 2021]

Biecek, P. & Burzykowski, T., 2020. Explenatory Model Analysis. [Online] Available at: https://pbiecek.github.io/ema/preface.html [Accessed 20 5 2021]

Bird, R. and McCrae, M., 1987. Tests of the Efficiency of Racetrack Betting Using Bookmaker Odds. Management Science, 33(12), pp.1552-1562.

Breiman, L., 1984. Classification and regression trees.

Breiman, L., 2001. Random Forest. Machine Learning, 45(1), pp. 5-32.

Brownlee, 2016, [Online] Available at: https://machinelearningmastery.com/classification-and-regression-trees-for-machine-learning/ [Accessed 27 5 2021]

Chen, T. & Guestrin, C., 2016. XGBoost: A Scalable Tree Boosting System. San Francisco, ACM.

Chen, T. & He, T., 2020. xgboost: eXtreme Gradient Boosting. [Online] Available at: http://cran.fhcrc.org/web/packages/xgboost/vignettes/xgboost.pdf [Accessed 22 04 2020].

D. Harville, 1973, Assigning probabilities to the outcomes of multi-entry competitions, Journal of the American Statistical Association 68 (342) 312–316.

Donges, 2019, Random Forest: [Online] Available at: https://builtin.com/data-science/random-forest-algorithm, [Accessed 20 05 2021]

Elreedy, Dina & Atiya, Amir & Shaheen, Samir. 2019. A Novel Active Learning Regression Framework for Balancing the Exploration-Exploitation Trade-Off. Entropy. 21. 651. 10.3390/e21070651.

Eugene F. Fama, 1970 Efficient Capital Markets: A Review of Theory and Empirical Work

Fama, E., 1970. Efficient Capital Markets: A Review of Theory and Empirical Work. The Journal of Finance, 25(2), p.383.

Fisher, Aaron, Cynthia Rudin, and Francesca Dominici. 2019. "All Models Are Wrong, but Many Are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously." Journal of Machine Learning Research 20 (177): 1–81. http://jmlr.org/papers/v20/18-760.html

Gabriel, P. and Marsden, J., 1990. An Examination of Market Efficiency in British Racetrack Betting. Journal of Political Economy, 98(4), pp.874-885.

Garson, G.D., 1991. Interpreting neural network connection weights. Artificial Intelligence Expert 6, 47-51

Gevrey, M., Dimopoulos, I. and Lek, S., 2003. Review and comparison of methods to study the contribution of variables in artificial neural network models. Ecological Modelling, 160(3), pp.249-264.

Górecki, T. and Łuczak, M., 2013. Linear discriminant analysis with a generalisation of the Moore–Penrose pseudoinverse. International Journal of Applied Mathematics and Computer Science, 23(2), pp.463-471.

Guresen, E. and Kayakutlu, G., 2011. Definition of artificial neural networks with comparison to other networks. Procedia Computer Science, 3, pp.426-433.

H. Chen, et al., 1994. Expert prediction, symbolic learning, and neural networks: an experiment on greyhound racing, IEEE Expert 9 (6) (1994) 21–27.

Harville, D., 1973. Assigning Probabilities to the Outcomes of Multi-Entry Competitions. Journal of the American Statistical Association, 68(342), pp.312-316.

Hastie, Qian, Tay, 2021, [Online] Available at: https://glmnet.stanford.edu/articles/glmnet.html [Accessed 27 5 2021]

Hausch, D., Ziemba, W. and Rubinstein, M., 1981. Efficiency of the Market for Racetrack Betting. Management Science, 27(12), pp.1435-1452.

Heaton, 2017, [Online] Available at: https://www.heatonresearch.com/2017/06/01/hidden-layers.html [Accessed 27 5 2021]

Henery, R., 1981. Place probabilities in normal order statistics models for horse races. Journal of Applied Probability, 18(4), pp.839-852.

J. Ritter, 1994. Racetrack betting—an example of a market with efficient arbitrage, in: D. Hausch, V. Lo, W. Ziemba (Eds.), Efficiency of Racetrack Betting Markets, Academic Press, San Diego.

Kuhn, M. & Johnson, K., 2013. Applied Predictive Modeling. 5 ed. Ney York City: Springer.

Lek, S., Delacoste, M., Baran, P., Dimopoulos, I., Lauga, J. and Aulagnier, S., 1996. Application of neural networks to modelling non-linear relationships in ecology. Ecological Modelling, 90(1), pp.39-52.

Lo, V. and Bacon-Shone, J., 1994. A Comparison Between Two Models for Predicting Ordering Probabilities in Multiple-Entry Competitions. The Statistician, 43(2), p.317.

Lo, V., Bacon-Shone, J. and Busche, K., 1995. The Application of Ranking Probability Models to Racetrack Betting. Management Science, 41(6), pp.1048-1059.

Luce, R. D. and Suppes, 1965. Preference, utility and subjective probability. In Handbook of Mathematical Psychology (eds R. D. Luce, R. R. Bush and E. Galanter), vol. III, ch. 19, pp. 249-410. New York: Wiley.

Mcculloch, B. and Zijl, T., 1986. Direct test of Harville's multi-entry competitions model on racetrack betting data. Journal of Applied Statistics, 13(2), pp.213-220.

Mohana, R., Reddy, C., Anisha, P. and Murthy, B., 2021. Random forest algorithms for the classification of tree-based ensemble. Materials Today: Proceedings,.

Nelder, J. and Wedderburn, R., 1972. Generalised Linear Models. Journal of the Royal Statistical Society. Series A (General), 135(3), p.370.

Olden, J. and Jackson, D., 2002. Illuminating the "black box": a randomisation approach for understanding variable contributions in artificial neural networks. Ecological Modelling, 154(1-2), pp.135-150.

Ozenne, B., Subtil, F. and Maucort-Boulch, D., 2015. The precision–recall curve overcame the optimism of the receiver operating characteristic curve in rare diseases. Journal of Clinical Epidemiology, 68(8), pp.855-859.

Pudaruth, S., Medard, N. and Bibi Dookhun, Z., 2013. Horse Racing Prediction at the Champ De Mars using a Weighted Probabilistic Approach. International Journal of Computer Applications, 72(5), pp.37-42.

R.P. Schumaker, J.W. Johnson, 2008. Using SVM regression to predict greyhound races, International Information Management Association (IIMA) Conference: San Diego, CA.

Rokach L., Maimon O. (2009) Classification Trees. In: Maimon O., Rokach L. (eds) Data Mining and Knowledge Discovery Handbook. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-09823-4_9

Saito, Takaya; Rehmsmeier, Marc (2015-03-04) [Online] Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4349800/ precision-recall curve [Accessed 27 5 2021]

Sarkar, 2019, LDA, [Online] Available at: https://www.knowledgehut.com/blog/data-science/linear-discriminant-analysis-for-machine-learnin [Accessed 27 5 2021]

Schumaker, R., 2013. Machine learning the harness track: Crowdsourcing and varying race history. Decision Support Systems, 54(3), pp.1370-1379.

Tharwat, A., Gaber, T., Ibrahim, A. and Hassanien, A., 2017. Linear discriminant analysis: A detailed tutorial. AI Communications, 30(2), pp.169-190.

Trevor HastieJunyang QianKenneth Tay, 2021, [Online] Available at: https://cran.r-project.org/web/packages/glmnet/vignettes/glmnet.pdf [Accessed 18 05 2021]

Williams, J. and Li, Y., 2008. "A Case Study Using Neural Network Algorithms: Horse Racing Predictions in Jamaica," in International Conference on Artificial Intelligence, Las Vegas, NV.

Zamfir M., Manea M. and Ionescu L. (2017) Return on Investment – Indicator for Measuring the Profitability of Invested Capital. Valahian Journal of Economic Studies, Vol.7 (Issue 2), pp. 79-86. https://doi.org/10.1515/vjes-2016-0010

University of Warsaw
Faculty of Economic Sciences
44/50 Długa St.
00-241 Warsaw
www.wne.uw.edu.pl