Comparing ARIMA and XGBoost algorithms on multiple time series golf data

James Barton
Data Science
KTH Royal Institute of Technology
Stockholm, Sweden
jbarton@kth.se

Anouk van Kasteren
Data Science
KTH Royal Institute of Technology
Stockholm, Sweden
anoukvk@kth.se

Jonas Birk
Data Science
KTH Royal Institute of Technology
Stockholm, Sweden
jsbirk@kth.se

***Abstract*—XGBoost is one of the most versatile, accurate and popular algorithms at the moment. ARIMA (AutoRegressive Integrated Moving Average) however, is a less well-known, more specialised algorithm that is designed to work specifically with time-series data. Professional golf events produce a unique format of time-series data in the form of the results of the event. The aim of the project is to compare how the two algorithms perform on the golf data set. Results show the both models perform rather bad on the data in terms of MSE, R^2 and Kendall's Tau.**

Keywords—ARIMA, XGBoost, time-series, golf, machine learning

I. Introduction

Time series analysis is a well studied and applied method in data science. Application domains are for example the forecasting of sales, the weather, and sports results. In time series analysis past data is used to make predictions on future values. This is straightforward for individual time series but can also be applied to multiple time series by combining all data to train one model [1]. Reasons for this could be a lack of data on some of the series which makes training individual models impossible. Also, when it is expected that all series have similar relations to past observations and can be predicted using a single model, fitting one model could save on computational cost and time. Application in our named examples could be several products, or multiple sports players. The latter will be used for the current study. Several algorithms can be applied for this purpose. In this study we will compare two of them. The autoregressive integrated moving average (ARIMA) and the XGBoost algorithm.

A previous work by Kane et al.[2] already carried out a comparison between the two methods on influenza data. The results of the investigation form the basis of the research question and hypotheses to be examined in this paper where we will test how these models perform on sports analytics, specifically golf results forecasting of multiple time series. The following research question will be investigated:

"In analyzing multiple time series how performs ARIMA compared to a random-forest model for the complete dataset in terms of MSE, R^2 and Kendall tau?"

With the following corresponding hypothesis:

H1: The Predictive ability of the Random Forest algorithm will be better than the performance of the ARIMA algorithm in terms of MSE and R^2 and Kendall tau.

II. Method

A. Algorithms

The two compared algorithms (ARIMA and XGBoost) will be explained here.

The ARIMA model is an update of the ARMA model first introduced by Peter Whittle [3] and later popularized by George E. P. Box and Gwilym Jenkin [4]. The model consists of two parts: Autoregression and a moving average. The i stands for integrated and takes care of the non-stationarity data. The model has 3 parameters: p, d, and q. P is the number of past values considered in the model, q is the number of past values considered in the moving average and d is the number of times the data are differenced to overcome non-stationarity. By differencing the data a past value is subtracted from the current value. For the golf data, this is applied once with a lag of 1, meaning that of each value the last past value is subtracted. However, if there are seasonal trends in the data this lag could be adjusted to this. The dependent variable will be the position and the independent variables will the lagged positions, the moving average and the golferID to correct for individual differences. The algorithm is trained for different value combinations for parameters p and q ranged between 5 and 11, the results are compared to find the best fit.

XGBoost stands for extreme gradient boosting and can be described as a scalable implementation of gradient boosting machines with the focus on improving the model performance and execution speed. It was developed by Tianqi Chen [5] and can be used by an open source library, accessible with popular programming languages like python R or Julia. In past machine learning challenges, like the Kaggle competition, for example, it was a widespread tool among all winning solutions. It is a highly flexible and versatile tool that can work through most regression, classification, and ranking problems.

The basic concept is boosting, which is an ensemble method that aims to create a strong classifier based on weak classifiers. It's an iterative process where the weight of each learner is learned by whether it predicts a sample correctly

or not. If a learner is mispredicting a sample, the weight of the learner is reduced. The process is repeated until converge. To be more precise the boosting of the XGBoost is a gradient boosting. XGBoost can be used in combination with different learning algorithms (T. Chen, C. Guestrin, 2016).

In this work a Random Forest Algorithm with XGBoost is used and implemented in R with the R-Package xgboost. In addition to the standard parameter settings of the so-called general and booster Parameters, certain learning task parameters settings were tested in the model building phase.

B. *The Data*

The basis of the investigation is time series golf data. The dataset consists of around 30.000 records of over 2000 golfers and was scraped from several golf sport websites that report on the weekly results. Each record contains the following independent variables: GolferID, Position, Week. These variables are used to predict the position of a golfer in a specific tournament as target variable. For the ARIMA algorithm the data was transformed in a way that dummy variables consisting of p past values for that specific player/entity and a moving average over q past values. Figure 1 is a screenshot of a table with golfers results taken from owgr.com, this is where the majority of the scraped data comes from.

The histogram in figure 2 shows the number of competitions per golfer, as you can see the majority of golfers in our database have only played a few tournaments. We think this is to be expected as only the best golfers tend to play frequently in the high level tournaments. We suspect that a lot of the golfers in the first bar will be ranked below 500th in the world. Thus the golfers most likely to win are also the ones we will probably have the most data for so this shouldn't be such a problem. To avoid many missing data points the data is filtered such that only golf players with data in more than 50% of the weeks will remain in the dataset.

The position is the target variable with a range 1 to 240 in the data. Due to the nature of the time series data position is also an explanatory variable, as the previous positions will be used to predict future positions. Ranking points is also a explanatory variable - players earn ranking points based on where they finish in events and how strong the field is that competed in the event. With a maximum of 100 points for a 1st place finish in a Major and a minimum of 24 points for a first place finish on any European or PGA Tour event.

T67	Sam Robertshawe	73	73	75	78	299	0
T67	Tomas Silva	74	72	76	77	299	0
69	Chris Selfridge	73	73	75	79	300	0
MC	Bjorn Hellgren	75	72	-	-	147	0
MC	Jack Doherty	75	72	-	-	147	0
MC	Matt Ford	72	75	-	-	147	0
MC	Max Schmitt	75	72	-	-	147	0
MC	Marco Penge	74	73	-	-	147	0

Fig. 1. Example of a table we would scrape, found on owgr.com.

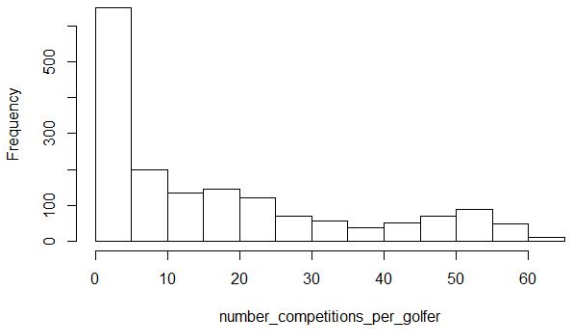


Fig. 2. Histogram of number of competitions per golfer.

C. *Imputation*

Golfers that are in the bottom half of the event after two rounds are said to have 'missed the cut' (MC) and therefore do not complete the event. As such their position is shown as 'MC', instead of the actual position. The scraping code therefore takes the golfer's index in the table as their position as all the tables are sorted on position by default.data comes from.omes from.

Further missing values are caused by golfers not competing every week. To reduce the number of missing values golfers that haven't competed in at least half of the events are removed from the data set. Secondly, remaining missing values are replaced with the average of the nearest available values, taking their distance into account. i.e. if week 2 and week 3 are missing for a golfer then week 2's imputed value will be:

$$\frac{2}{3}\,week\ 1\ position\ +\ \frac{1}{3}\,week\ 4\ position$$

whilst week 3's value will be:

$$\frac{1}{3}\,week\ 1\ position\ +\ \frac{2}{3}\,week\ 4\ position$$

D. *Data preparation*

To make the data usable for our implementation of both some additional data preparation needed to be done. First, the variables 'Rankingpoints' and 'Odds' were dropped as they would not be used in the model due to insufficient data points. Second, the 'Position' variable had to be transformed into a numeric type variable. To make it possible to plot individual golfer's positions the data was transformed into a pivot table with each column representing a golfer, the week number as the index, and the positions as the row values.

E. *Data preparation* Arima*t*

For the ARIMA Model additional data preparation was necessary. Depending on parameter p, dummy variables are created for p past positions. Also, depending on parameter q, a dummy variable for the moving average over q past values is created. Last, to make the data more stationary, the data is differenced. Depending on parameter d, this will create missing values for the d first observations, these rows will be dropped from the dataset.

F. *Performance metrics*

Results of both models will be compared according to the following performance metrics: mean squared error (MSE), R^2, and Kendall correlation. The MSE will give an indication of the accuracy of the predictions and will allow us to compare the models regarding this. The R^2 tells us

what proportion of the variance in the data is explained by the model, this metric will allow us to compare models regarding how well the model fits the data. Another performance metrics is the Kendall correlation, which is a proper correlation measure for ordinally scaled values. With the correlation between the predicted and the actual position of a golfer, we can evaluate the model in an additional way. The Kendall correlation between two variables is high when observations have a similar rank and low when observations have a dissimilar rank. The highest possible value is 1 for identical ranks and the lowest is -1 for completely different ranks. [6] A high positive correlation value is sought for the objective of this work.

III. Results

This sections will describe the results of different runs of both the ARIMA and XGBoost model. Figure 3 shows the position of randomly selected golf players over time. This figure gives an indication how the data behaves.

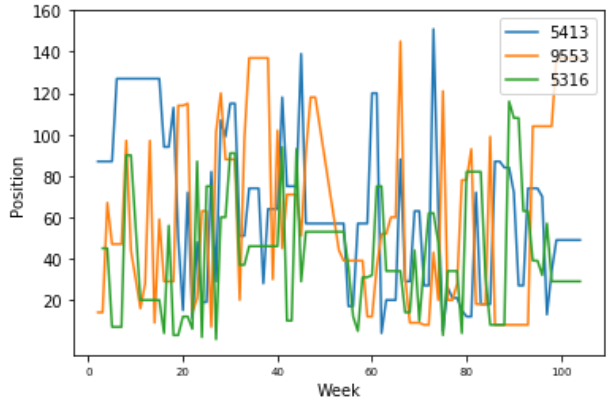


Fig. 3. the position of three randomly selected golf players(GolferID = 5413, 9553, and 5316) showing the behavior over time.

A. ARIMA

The model was fitted for two datasets, one without imputation and one with mean imputation for missing weeks. The ARIMA(p,d,q) model was fitted and evaluated for different parameter values for p and q ranging from 5 to 11. The data was differenced once (d = 1) with a lag of 1 week. Figure 4 show the performance of the model on both datasets measured in MSE, R2, and Kendall's Tau (For larger plots see appendix A). For both models patterns can be seen in the performance of the model with different parameter values. The general performance improves as parameter values increase.

The final model on the not-imputed data was chosen to be ARiMA(9,1,9). For the imputed data the chosen final model was ARiMA(11,1,9). Table I shows the performance metrics on the chosen models on both datasets. The table shows that the MSE is lower for the imputed data and the R^2 and Kendall's Tau are higher for the imputed data.

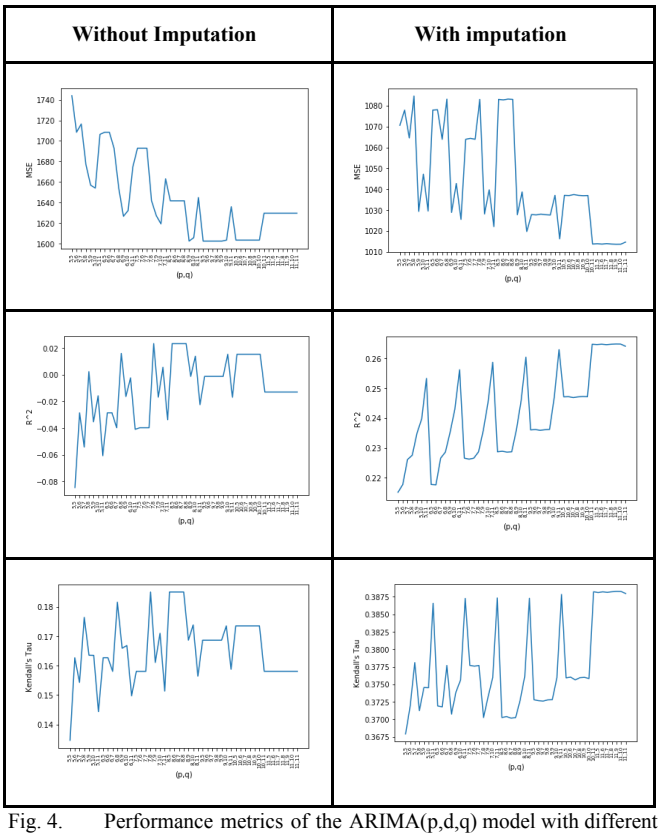


Fig. 4. Performance metrics of the ARIMA(p,d,q) model with different parameter values of p and q. top to bottom: MSE, R^2, and Kendall's Tau. Fitted to both the imputed and not-imputed data.

TABLE. I Performance Metrics On The ARIMA Model

Metrics	**ARiMA(9,1,9) Without Imputation**	**ARiMA(11,1,9) With Imputation**
MSE	1602.43	1013.62
R^2	-0.001	0.265
Kendall cor.	0.1687	0.388

B. Random Forest XGBoost

As the ARIMA Model the Random Forest XGBoost Model was also fitted for the two datasets with and without imputation for missing weeks. Comparing all chosen metrics (see Table II) for both models with the same model parameters one can clearly see that the imputation of the input data improves the performance of the model.

After deciding for choosing the dataset with imputation the second step was to optimize the performance of the model. The strongest influence on the performance in the case of this work is the maximum tree depth (max depth). Therefore the model was trained with the following five different max depth settings (5, 7, 10, 12 or 15). Results can be seen in Table III.

TABLE II:Comparison of the performance with and without imputation

Metrics	Without Imputation	With Imputation
MSE	1781.87	1309.51
R^2	0.054	0.158
Kendall cor.	0.158	0.279

TABLE III: Comparison of the performance with different max depth settings

Performance Metrics	Max depth: 5	Max depth: 7	Max depth: 10	Max depth: 12	Max depth: 15
MSE	1266.15	1278.17	1309.51	1315.11	1293.49
r2	0.179	0.174	0.158	0.155	0.166
Kendall cor.	0.297	0.294	0.279	0.272	0.282

As one can see in the table above the model performs best with a max depth of five in all of the three performance metrics. For a deeper evaluation of the predictive ability of the model with the best parameter setting the following plot is helpful.

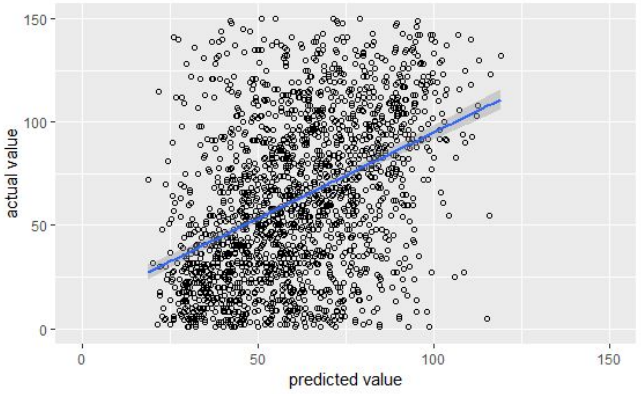


Fig. 5. comparison of predicted and actual values with the regression line:

The Plot in Fig. 5 shows a scatterplot with a comparison of the predicted and actual values which can be used to uncover the strength and weaknesses of the model. Even if there are many clear false predictions there is a recognizable pattern. The regression line can be seen as a visualization of the correlation of both variables. When looking at this line, a trend can be seen that shows that the model is not completely wrong. One obvious weakness of the model is the ability to predict low rankings. It only predicts two players to finish in the Top 20 with a lowest predicted ranking of 18.

With the results above the hypothesis, that the predictive ability of the Random Forest algorithm will be better than the performance of the ARiMA algorithm in terms of MSE and R^2 and Kendall tau can not be confirmed. The reason for the opposite result compared to the work of Kane et al.[2] may lie in the different areas of application and the kind of data that is used in both works.

IV. Discussion

Both the ARIMA as the XGBoost model resulted in predictions with a large error and a weak fit. This indicates a lack of a linear relation between past positions and future positions. Imputation improved the models slightly but performance would still not be considered acceptable. Contrary to our hypothesis, the ARIMA model performs better on both datasets compared to the XGBoost algorithm on all metric except the R^2 on the not-imputed data. Which is surprising. It would be expected that the boosted algorithm, which also allows for non-linear patterns would outperform the ARIMA which is based on a linear regression.

We compared the results of our data to that of a dataset known to be successful with time series analysis. This dataset contains data of the sales of multiple products over several weeks. The results of the ARIMA are: MSE = 13.1; R^2 = 0.91; and Kendall's Tau = 0.80. For the XGBoost the results are: MSE = 3.5; R^2 = 0.93; and Kendall's Tau = 0.68. On this data the model performs reasonably better, which simply indicates that there is a stronger relationship between past and future trends in the comparison data set, which is to be expected. As such, measuring the performance of the model more accurately would require comparing with a related system. Betting odds serve as an appropriate metric for such a comparison and going forwards, the model will be deployed in a live environment and measured in accuracy against these odds.

A. *Limitations*

One of the limitations of the data being presented as a time series i.e. week by week is the number of missing values this produces. With there being at most two events per week on the two high profile tours (PGA and European) with an average of 150 players per event and over 2000 Golfers in our data set.

Another limitation was the fact that the available ARiMA statsmodel library cannot be applied to multiple time series data. We therefore had to work around to implement our own version.

B. *Effect of the Imputation*

By imputing the missing weeks with the mean of the position the week before and the week after a pattern more linear than the reality might have been created. This may have caused the model to be overfit to unrealistic idealised data and thus could be the reason the model performance increased after imputation. However, the accuracy of the model on new, unimputed test instances also improved, thus any overfitting is justified.

References

[1] N. Wagner, Z. Michalewicz, S. Schellenberg, C. Chiriac, and A. Mohais, “Intelligent techniques for forecasting multiple time series in real-world systems,” International Journal of Intelligent Computing and Cybernetics, vol. 4, no. 3, pp. 284–310, 2011.

[2] M. J. Kane, N. Price, M. Scotch, and P. Rabinowitz, “Comparison of ARIMA and Random Forest time series models for prediction of avian influenza H5N1 outbreaks,” BMC Bioinformatics, vol. 15, no. 1, p. 276, 2014.

[3] P. Whitle, Hypothesis testing in time series analysis. 1951.

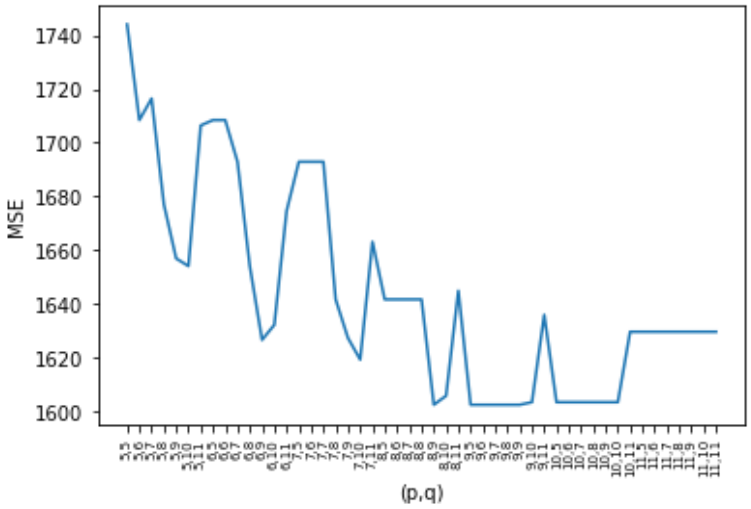
[4] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, Time series analysis: forecasting and control. 1970.

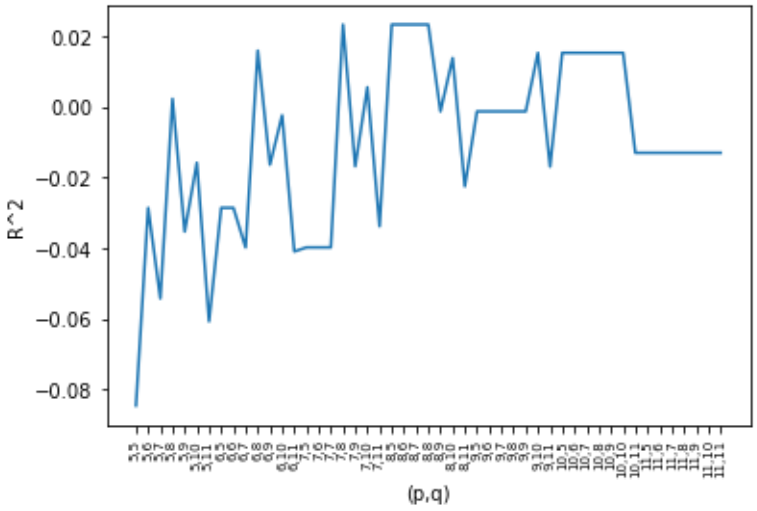
[5] Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794). ACM.

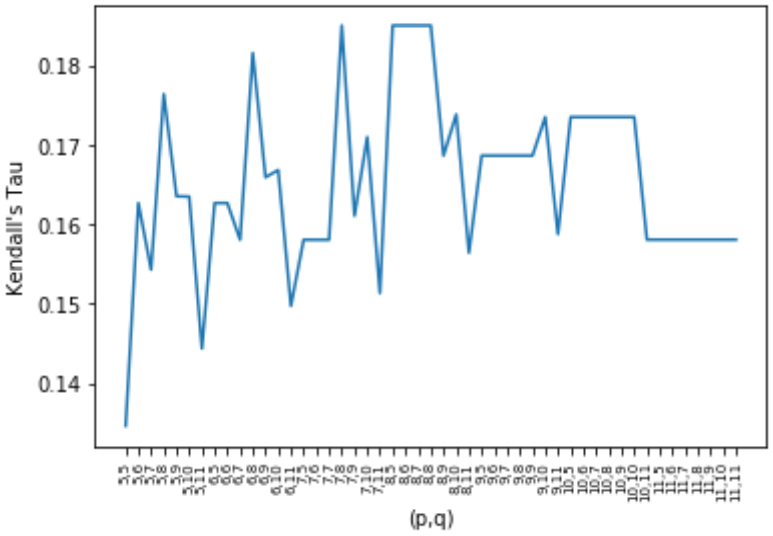
[6] Hahs-Vaughn, D. L., & Lomax, R. G. (2013). *An introduction to statistical concepts*. Routledge.

APPENDIX A: ARIMA METRICS PLOTS

Without imputation:







With imputation: