ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA

Ingegneria e Scienze informatiche

FANTA ANALYTICS

Elaborato in
SISTEMI DI SUPPORTO ALLE DESCISIONI

Presentato da
GABRIELE GUERRINI,
SALVATORI RICCARDO,
SALVATORI STEFANO

Anno Accademico 2019 – 2020

Table of contents

Chapter 1

Data retrival and cleaning

1.1 Data retrival

Votes for each season since 05/06 are stored on "Gazzetta dello Sport" website (fantapiu3.com). In order to create the dataset, we exploited an existing page crawler realized for a previous project.
Firstly, we made code refactoring on such project so we could start from a well laid out hard core.
Then, we started by improving the application because the existing code would retrieve votes just for one season; we extended the retrival to more seasons, in particular we used 3 seasons but that's not a constraint since the program is parametric on this number. This parameter has been set to 3 since we evaluated that football market is very dynamic and just few players keep playing in "Serie A" for years; some trials have been made and 3 resulted a good compromise.
In the end, votes were saved on external file using CSV standard format; two files were created respectively for votes and fantavotes.
A third dataset storing bonuses is created too from the existing two by a simple difference $fantavote - vote$.

1.2 Data cleaning

Few checks have been made on categorical attributes in order to fix wrong values (e.g. "OKONKWOO" resulted playing in not existing role).
Then, all players that did not play at least one match were dropped since no analysis can be performed with no vote at all.

Chapter 2

Data preprocessing

2.1 Filling missing values

The main issue is to put a vote on non played matches for each player. Two approaches have been implemented to achieve this goal, as follow:

- **Linear interpolation**: missing values are interpolated using the existing ones; by increasing the cardinality of missing values, this method obviously loses effectiveness but, on the other hand, missing values tend to reflect self trend for each player. (Figure 2.1)

- **Constant placeholder**: a fixed value is set for each missing vote
  $placeholder = minVote - 1$
  No assumption on filled vote confidency can be made since the value is indipendent from player itself. (Figure 2.2)

Chosen a method, all three dataset are modified, and no missing vote is left.

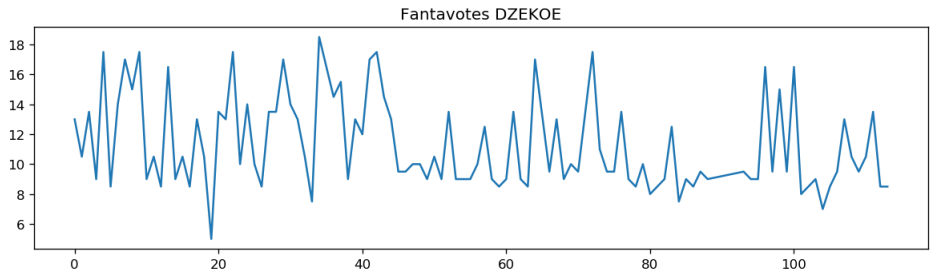

Figure 2.1: *Fantavotes of "Edin Dzeko" with linear interpolation*

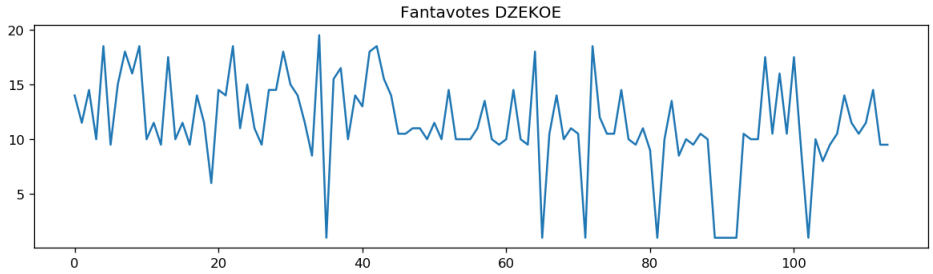

Figure 2.2: *Fantavotes of "Edin Dzeko" with placeholder*

2.2 Translation

Since votes can be negative we decided to coherently translate them up in order to have all positive values; this was necessary, for example, to eventually apply log transform operations.

Chapter 3

Analysis on players data

The previous phases of data cleaning resulted in a dataset of 917 players with 114 votes each. Analysis are performed on different abstraction levels. At the beginning, high-level evaluations were made to discover informations on the number of played matches (mean, min, max etc.). Figure 3.1 is an exapmle of this analysis: it shows how many players played foreach given number of matches.

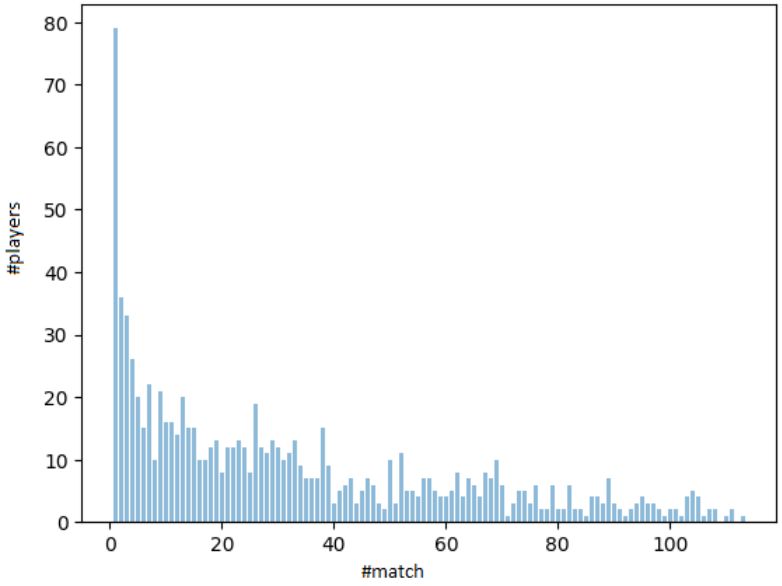

Figure 3.1: *Count number of players that played a certain number of matches.*

Similarly, a fine grained analysis has been made by grouping players per role, as shown in figure 3.2, but no relevant differences have been found.

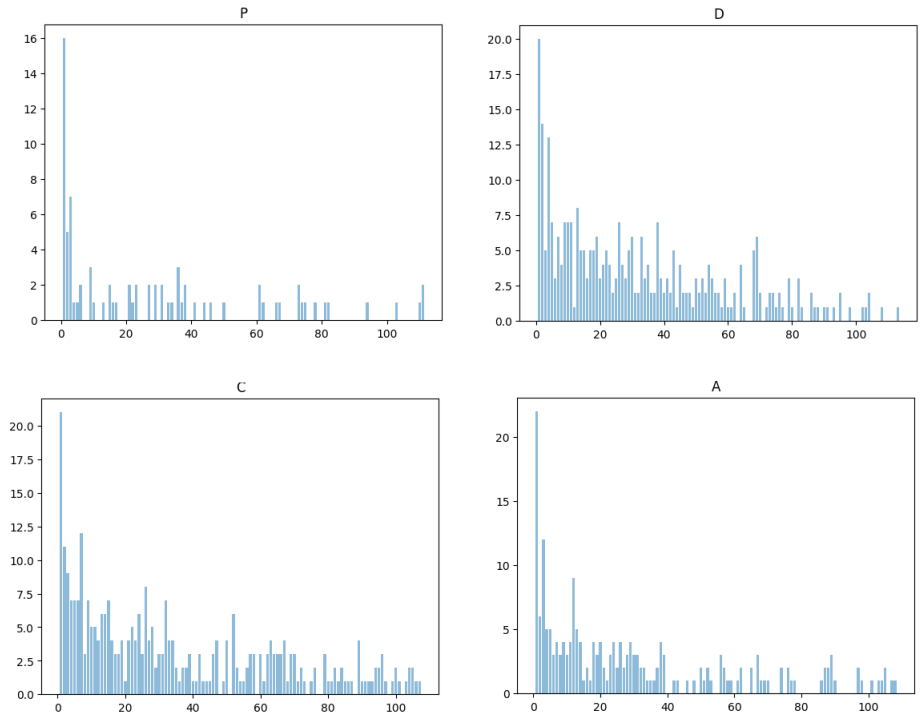

Figure 3.2: *Count number of players of such role that played a certain number of matches. Legend: P = Goalkeeper, D = Defender, C = Midfielder, A = Striker*

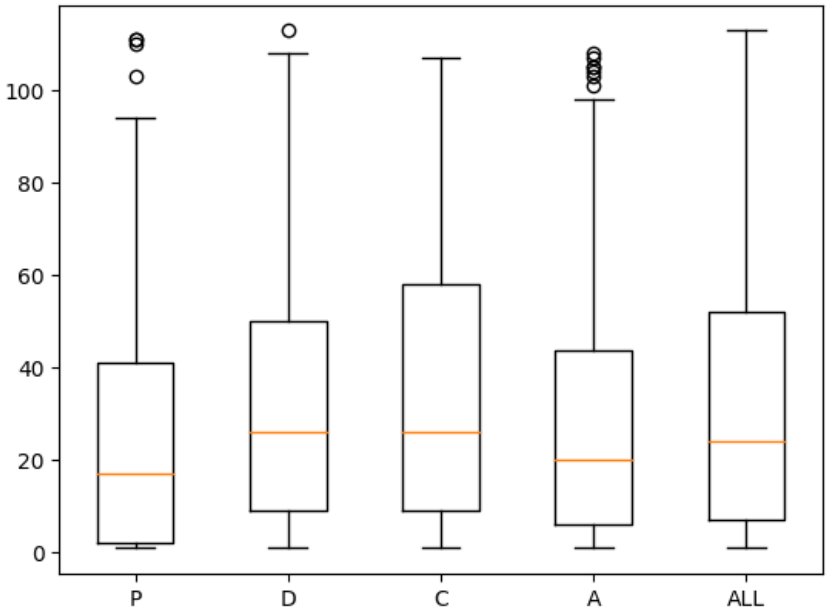

Figure 3.3: *Box plot of number of match played. Legend: P = Goalkeeper, D = Defender, C = Midfielder, A = Striker, ALL = All Players*

From figure 3.1,3.2 and 3.3 we can see that most players in the dataset played very few matches. In fact we computed that, on average, 32 matches are played (of the 114 total) and 75% of the players (almost 687) played less than 52 games. This could lead to some problems because it means that the process of interpolation would have ended up filling a lot values distorting the results of the analysis. In order to solve this issue we decided, for most part of the project, to focus on a subset of players that played at least 100 games .

3.1 Data correlation

The goal is discovering, if present, seasonality on players' votes. In order to achieve such target, *Pearson Correlation Index* has been used. As mentioned, we have been careful not to consider results biased by the process of interpolation: if we look at figure 3.4 we can see that considering players with a lot of interpolated votes (small number of match played) we get best pearsons coefficients that seem very high but this is due to the lack of real values; looking at player that played more matches we can in fact say that no relevant seasonality can be detected.

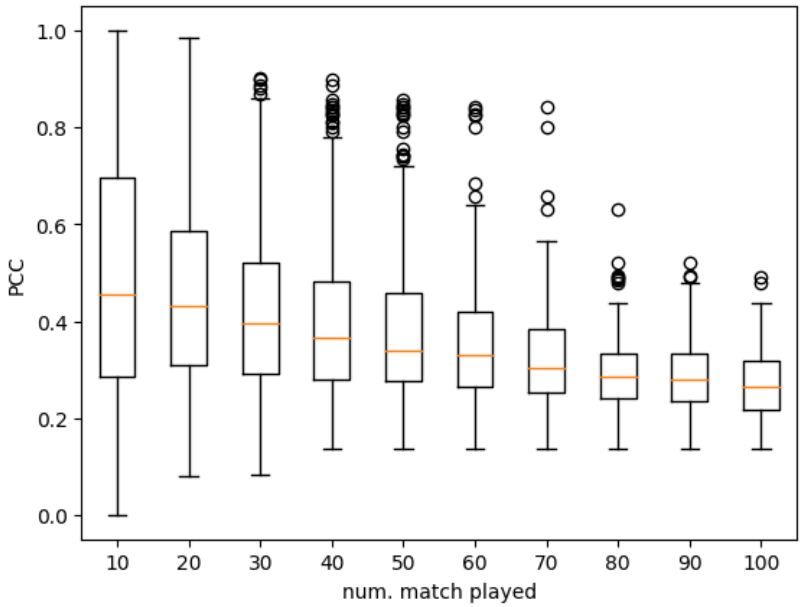

Figure 3.4: *Box plots of the best perason correlation coefficients computed on fantavoti of players that played at least the number of matches in the x axis.*

Figure 3.5 reports pearson coefficient computed on an example player for voti, fantavoti and bonus.

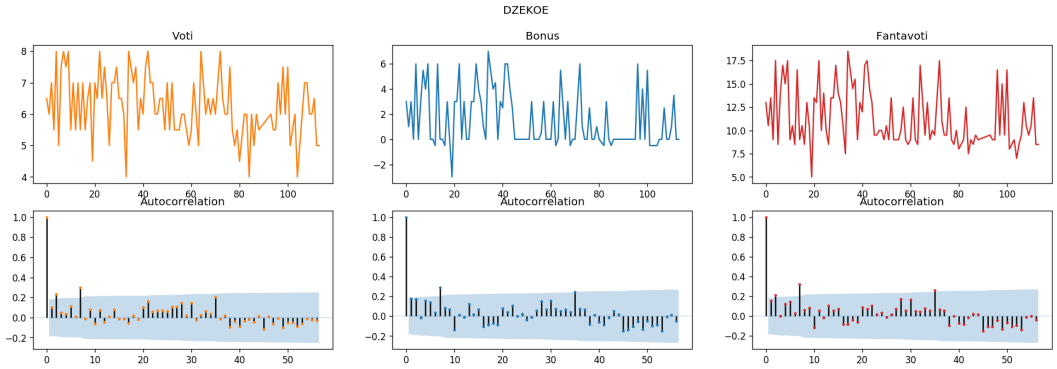

Figure 3.5: *The image shows storic series and respective Pearson correlations for "Edin Dzeko".*

3.2 Normality test

Normalty test has been used on votes and fantavotes both in order to determine if the two datasets could be modeled similarly to a normal distribution. QQ-plot and histograms of votes showed a correlation between votes and normal distribution for almost all players (Figure 3.6 and Figure 3.7). The same can't be asserted about fantavotes (Figure 3.8 and Figure 3.9).

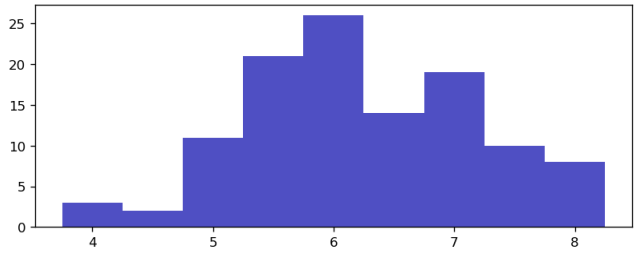

Figure 3.6: *'Votes' histogram for "Edin Dzeko".*

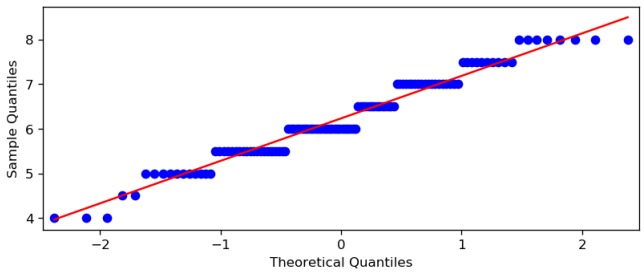

Figure 3.7: *'Votes' qqplot for "Edin Dzeko".*

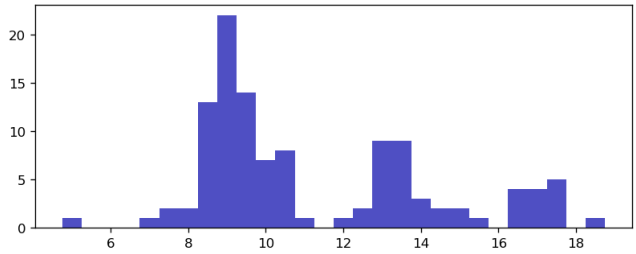

Figure 3.8: *'Fantavotes' histogram for "Edin Dzeko".*

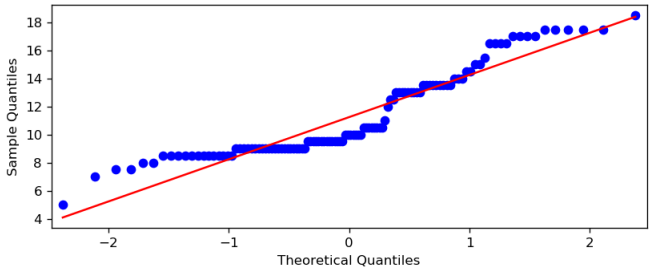

Figure 3.9: *'Fantavotes' qqplot for "Edin Dzeko".*

Chapter 4

Forecast

The forecasting has been implemented in three differents ways: SARIMA , MLP and LSTM. A subset of dataset has been used on forecast.This choice was made to avoid working on too many interpolated data; this also allowed us to reduce the training time for the forecasting models, particularly LSTM and MLP, Indeed, the dataset is filtered on players that played more than 100 matches in the considered range of three years, resulting on 25 selected. Fantavotes for each player have been divided into train and test sets (70% - 30% respectively). Performance are compared in table 4.1 using *Root Mean Squared Error* (RMSE) as common metric.

4.1 SARIMA

Regarding SARIMA, *auto-arima* function has been used to find the best parameters for ARIMA model. An example of application of SARIMA model is visible in Figure 4.1

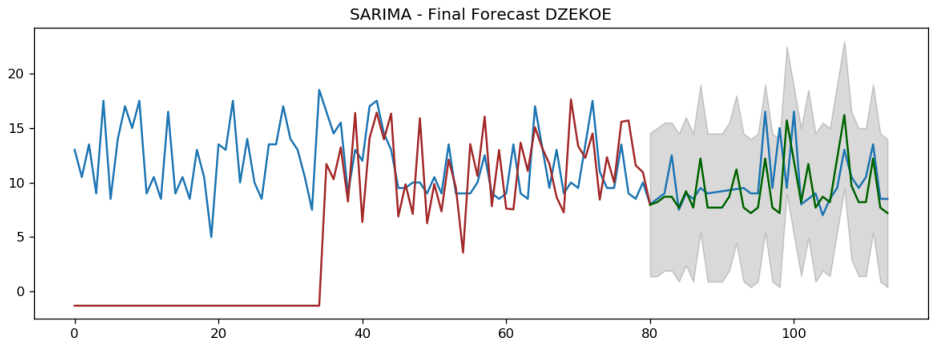

Figure 4.1: *SARIMA forecast on "Edin Dzeko".*

4.2 MLP

The MLP model architecture includes 1 input layer with 12 nodes, 1 hidden layer with 38 nodes and 1 output layers with 1 node. The activation function is *RELU*. The network has been trained with *Adam* optimizer and 150 epochs. An example of application of the MLP model is visible in Figure 4.4

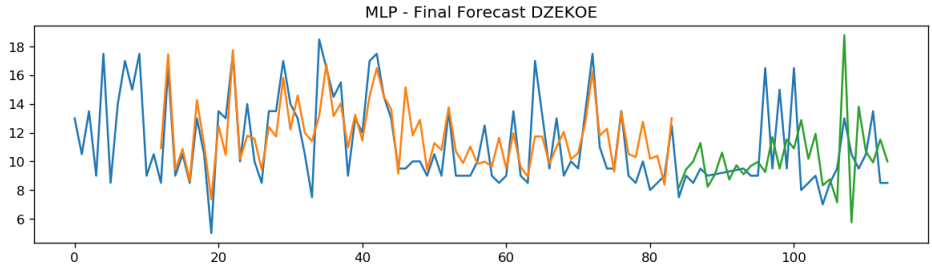

Figure 4.2: *MLP forecast on "Edin Dzeko".*

4.3 LSTM

The LSTM architecture includes 1 input layer with 12 nodes, 1 hidden layer with 32 LSTM cells and 1 output layers with 1 node. The activation function is *relu*. The network has been trained with Adam optimizer and 75 epochs. An example of application of the LSTM model is shown in Figure 4.4

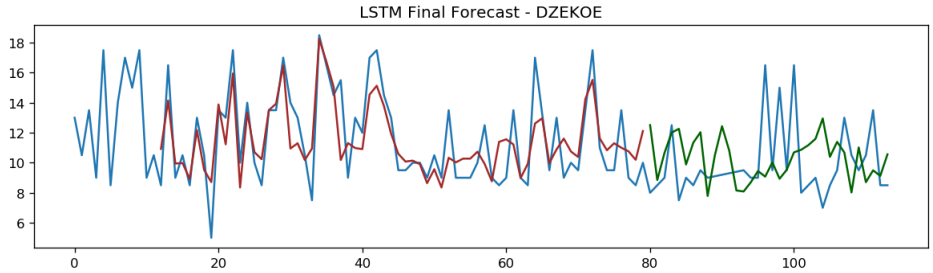

Figure 4.3: *LSTM forecast on "Edin Dzeko".*

4.4 Models combination

In order to improve accuracy on predictions, two type of models: statistic (SARIMA) and neural (LSTM) were ensembled to come up with an hybrid version.

The final model used for forecasting is defined as:

$$\alpha * LSTM + (1 - \alpha) * SARIMA$$

where:

- α is a weight in $[0; 1]$ that minimizes the RMSE
- **LSTM** is the value predicted by LSTM model
- **SARIMA** is the value predicted by SARIMA model

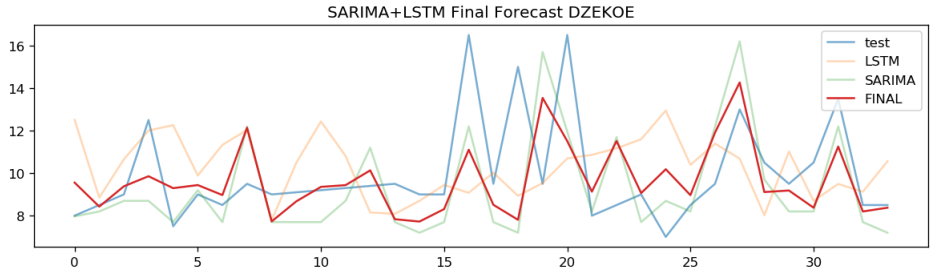

Figure 4.4: *Final forecast on test set of "Edin Dzeko".* $\alpha = 0.35$, *linear interpolation*

Filling method	SARIMA+LSTM	SARIMA	MLP	LSTM
Linear interpolation	1.89	2.77	2.21	2.04
Placeholder	3.48	4.22	4.31	4.25

Table 4.1: *Comparison between different models and filling methods using RMSE.*

Chapter 5

Conclusions

We made an in-depth analysis of football players performances by studying the dataset of votes and fantavotes from the past 3 seasons of 'Serie A'. First we found out that only a subset of players (from a total of almost 1000) could be used for our purpose because some of them didn't played any matches or played so few that they were not statistically relevant. The remaining players also had some missing values that were filled using linear interpolation or with a constant placeholder.

Then we made some high level analaysis on the dataset looking for patterns on data or general informations for the forecast phase. Using Pearson Correlation Coefficient we showed that players don't have a relevant seasonality component but in the other hand, votes are modeled pretty weel with a normal distribution unlike the fantavotes dataset that doesn't seem to have the same behaviour.

In the end we tried to forecast player's fantavotes using three different models: SARIMA,MLP and LSTM. We didn't reach really high accuracies probably due to the complexity of the task and the lack of releveant patterns in the dataset. The best we could get is a 1.89 average RMSE using a linear combination of SARIMA and LSTM models.