ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA

Ingegneria e architettura
Ingegneria e Scienze informatiche

FANTA ANALYTICS

Elaborato in
SISTEMI DI SUPPORTO ALLE DESCISIONI

Presentata 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 season, in particular we used 3 season but that's not a constraint since the program is parametric on number of seasons. 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 are 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" resulting on playing in not existing role).
Then, all players that did not play at least one match are 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.
- **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.

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

2.2 Translation

Votes are translated coherently in order to yield them in a positive range. This is useful e.g. when applying 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 are made to discover stats on number of played matches (mean, min, max etc.); analysis on number of played matches are reported in figure 3.1.

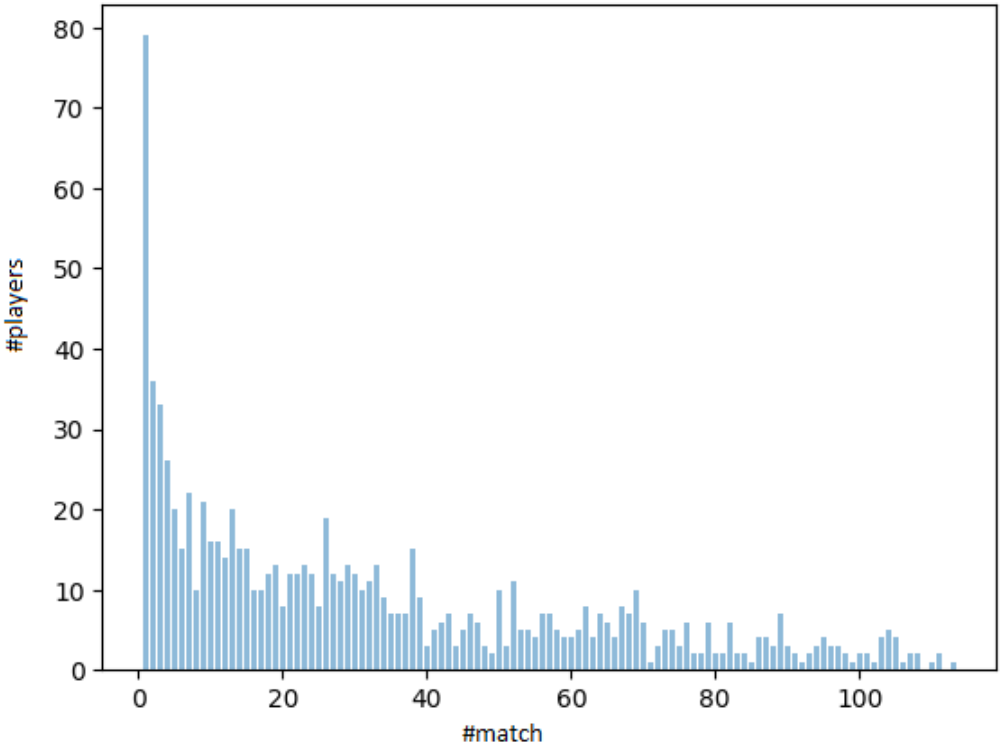

Figure 3.1: *Count number of players that played a certain number of matches.*

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.

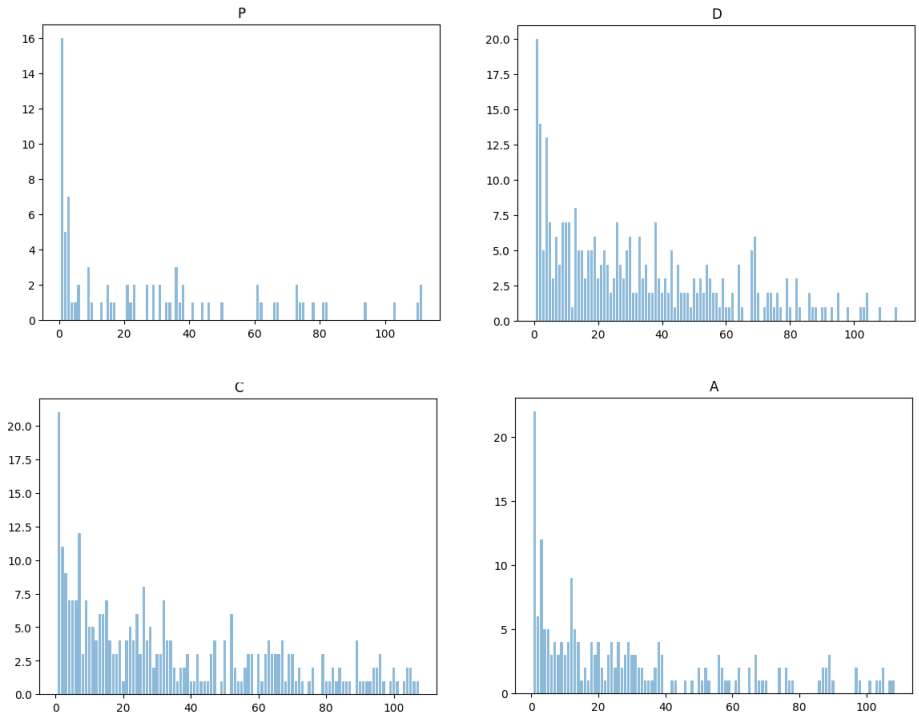

Figure 3.2: *Count number of players of such role that played a certain number of matches. Legend: P = Goalkeeper, D = Defender, C = Midfielder, A = Striker*

3.1 Data correlation

The task goal is discovering, if present, seasonality on players' votes.
In order to achieve such a target, Pearson Correlation Index has been used, for single player and for role both.
Empirical tests show that no relevant seasonality can be detected.
Figure 3.3 reports data on an example player.

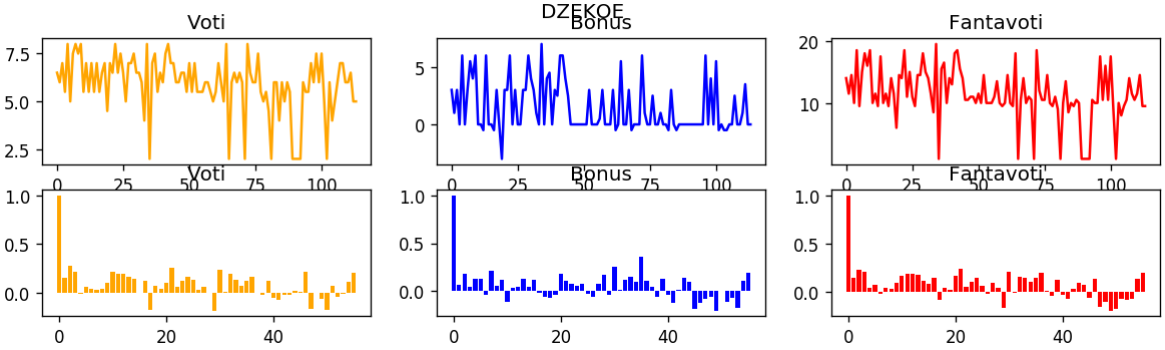

Figure 3.3: *The image shows storic series and respective Pearson correlations for "Edin Dzeko".*

3.2 Normality test

Normalty test were made 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 3.4 and Figure 3.5). The same can't be asserted about fantavotes (Figure 3.6 and Figure 3.7).

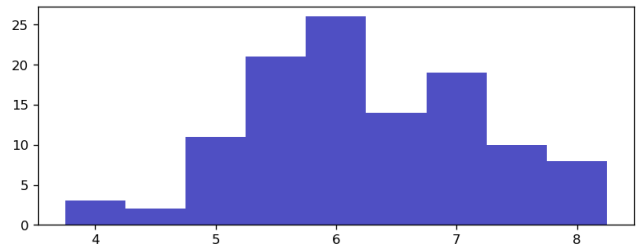

Figure 3.4: *'Votes' histogram for "Edin Dzeko".*

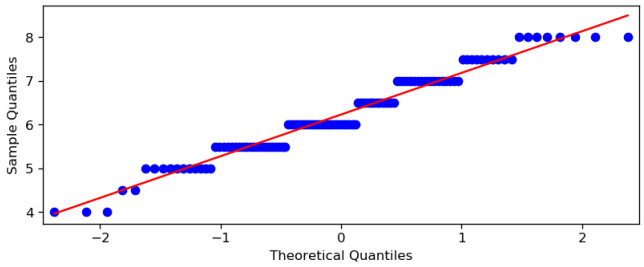

Figure 3.5: *'Votes' qqplot for "Edin Dzeko".*

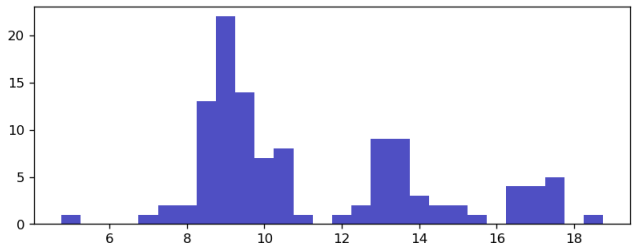

Figure 3.6: *'Fantavotes' histogram for "Edin Dzeko".*

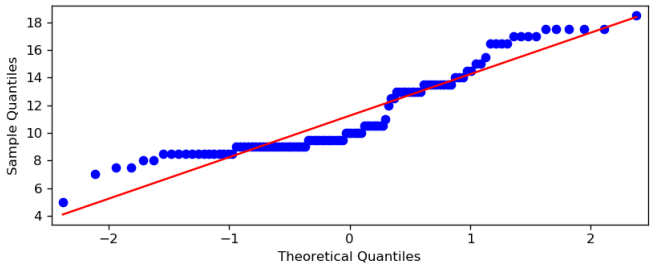

Figure 3.7: *'Fantavotes' qqplot for "Edin Dzeko".*

Chapter 4

Forecast

The forecasting has been implemented in two differents ways: SARIMA and MLP. A subset of dataset has been used on forecast; this choise is due to great time required to create and train the models. 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

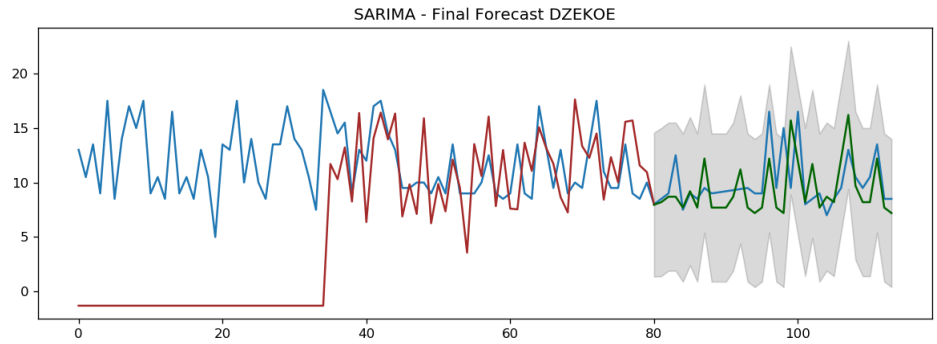

Figure 4.1: *SARIMA forecast on "Edin Dzeko".*

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

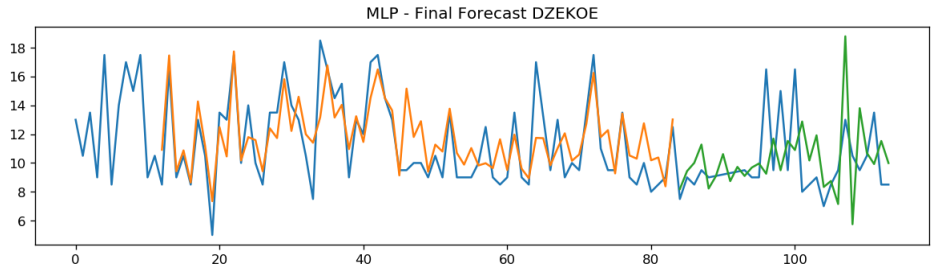

Figure 4.2: *MLP forecast on "Edin Dzeko".*

4.3 Models combination

The two different models are linearly composed in order to improve accuracy and reduce errors (*RMSE*). The final model used for forecasting is defined as:

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

where:

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

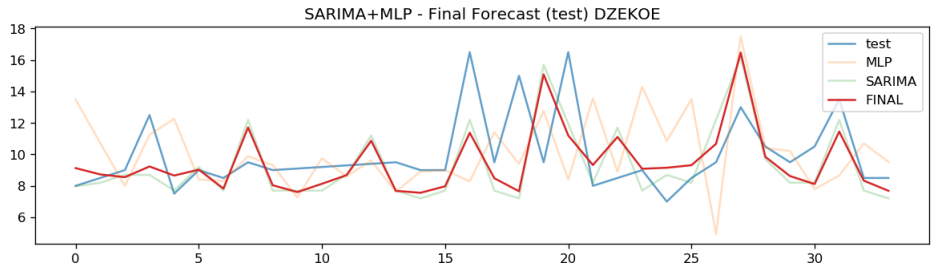

Figure 4.3: *Final forecast on test set of "Edin Dzeko".* $\alpha = 0.21$*, linear interpolation*

Filling method	SARIMA+MLP	SARIMA	MLP
Linear interpolation	1.94	1.97	2.21
Placeholder	3.48	4.22	4.31

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

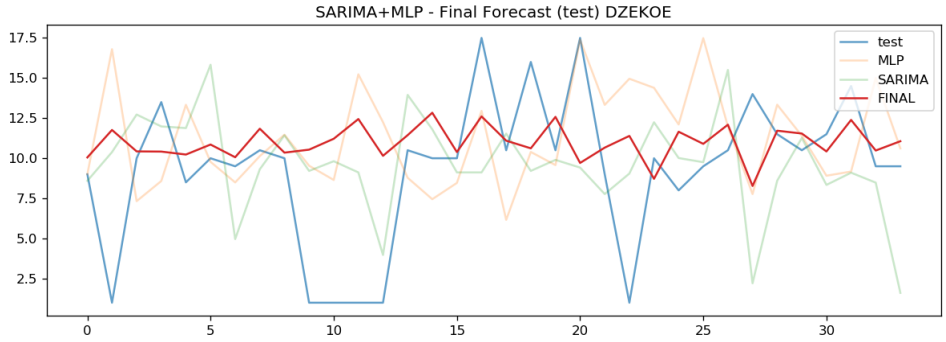

Figure 4.4: *Final forecast on test set of "Edin Dzeko".* $\alpha = 0.21$, *placeholder*