HW-4: Major-Leagues

Report

Pushkar Singh Negi

Ku ID: 2946319

EECS 731: Introduction to Data Science

Notebook: LabProject#4_Major_Leagues.ipynb

Purpose: Deduced Additional Information, visualization and random forest regression model.

Input raw dataset : nba_elo.csv
Processed dataset : AdditionalInfo#1_Count_Of_Match_Played_dateWise.csv, AdditionalInfo#2_SeasonWise_MatchCount.csv, AdditionalInfo#3_MeanScore_for_AllTeam1.csv, AdditionalInfo#4_MeanScore_for_AllTeam2.csv, AdditionalInfo#6_NumberOfMatchesPlayedYearWise.csv, AdditionalInfo#7_NumberOfMatchesPlayedMonthWise.csv, AdditionalInfo#8_NumberOfMatchesPlayedDateWise.csv
Jupyter Notebook File name : LabProject#4_Major_Leagues.ipynb

1. Imported the nba_elo.csv data file in dataframe using pandas.

2. Checked the total non-null entries to handle missing values.

3. Handled the missing values.

4. Analyzed the dataset with the help of various commands, such as finding the total unique seasons, date, team1 and team2.

5. Next, I found out the **Additional Information #1: To find total number of matches played each day/ or total days on which match was held/played.**

6. Next, I applied various pandas feature to convert the above result dataset into a panda frame.

7. Converted the above new data into a new data frame (df_nba_elo_No_of_matches_each_day).

8. Saved the resulted dataframe in a csv (AdditionalInfo#1_Count_Of_Match_Played_dateWise.csv)

9. **Additional Information #2:** To find total number of matches played in each season. Converted the above new data into a dataframe (df_nba_elo_Match_per_season).

10. Saved the resulted dataframe in a csv (AdditionalInfo#2_SeasonWise_MatchCount.csv).

11. Additional Information #3: To find the mean score for all the team1.

12. Converted the above new data into a dataframe (df_nba_elo_MeanScore_Team1)

13. Saved the resulted dataframe in a csv (AdditionalInfo#3_MeanScore_for_AllTeam1.csv)

14. Additional Information #4: To find the mean score for all the team2.

15. Converted the above new data into a dataframe (df_nba_elo_MeanScore_Team2)

16. Saved the resulted dataframe in a csv (AdditionalInfo#4_MeanScore_for_AllTeam2.csv)

17. Additional Information #5: Created 3 new columns namely year, month and date for each of the match played.

18. Additional Information #6: To find for each year how many matches were played.

19. Converted the above new data into a dataframe (df_nba_elo_newYEAR)

20. Saved the resulted dataframe in a csv (AdditionalInfo#6_NumberOfMatchesPlayedYearWise.csv)

21. Additional Information #7: To find for each month how many matches were played. Converted the above new data into a dataframe (df_nba_elo_newMonth)

22. Saved the resulted dataframe in a csv (AdditionalInfo#7_NumberOfMatchesPlayedMonthWise.csv)

23. Additional Information #8: To find for each month how many matches were played. Converted the above new data into a dataframe (df_nba_elo_newDate)

24. Saved the resulted dataframe in a csv (AdditionalInfo#8_NumberOfMatchesPlayedDateWise.csv)

26. My findings -->

- On dates 2013-04-17, 2016-11-25, 2014-04-16, 2009-01-02, 2011-04-13 maximum number of 15 matches were played.
- For season 2014 and 2016 maximum number of matches were playes i.e. 1319 matches.
- Among all the team1, DNA has the maximum mean score of 125.132653
- Among all the team2, WSA has the maximum mean score of 120.421053
- In year 2012, maximum numbers of 1474 matches were played.
- For all the seasons, the maximum number of matches were played in March i.e. 11877 matches.
- Maximum number of matches were played/held during start or end of the month.

Random Forest: Regression Model

- In order to apply random forest, changed the datatype of team1 and team2 column (object type) to int
- Find the labels and stored them separately i.e. the score we wanted to predict.
- Removed the labels from the features.
- Saved feature names for later use.
- To convert the dataframe to numpy array.
- Used Skicit-learn to split data into training and testing sets.
- Imported the random forest model.
- Instantiated the model with 1000 decision trees
- Trained the model on training data
- Used the forest's predict method on the test data
- Calculated the absolute errors
- Printed the mean absolute error (mae) i.e. 5.16 degree
- Calculated mean absolute percentage error (MAPE)
- Accuracy: 94.71 %.