**Alexandria University**

**Faculty of Computer and Data Science Department : Data Science**

**Course Title: Data Science 2023-2024**

**Exploring chess games from Lichess dataset**

Introduction to Data Science Course Code: 02-24-00104

# Members Names and Role

|  |  |  |
| --- | --- | --- |
| Name | ID | Role |
| 1.Ahmed Alaa | 22010002 | Data cleaning and preparation  &  Supervised learning |
| 2.Mariam Mohamed | 22011676 | Unsupervised learning |
| 3.Marawan mohamed | 22010248 | UI and visualization |
| 4.Marawan gaber | 22010246 | Apriori |
| 5. |  |  |
| 6. |  |  |

1. Introduction:

In this section you should describe the idea of the project and its objective, the inputs and outputs, the used dataset and its parameters.

The main idea of the project is to try to get some insights from chess games data , find similarities between them and try to train a model to predict the winner of the games without looking at the actual game, based on some information about the game, and also try to find some association rules in the different openings of chess.

The dataset can be found here: https://www.kaggle.com/datasets/datasnaek/chess

It contains around 20000 entries including the following features:

* Game ID;
* Rated (T/F);
* Start Time;
* End Time;
* Number of Turns;
* Game Status;
* Winner;
* Time Increment;
* White Player ID;
* White Player Rating;
* Black Player ID;
* Black Player Rating;
* All Moves in Standard Chess Notation;
* Opening Eco (Standardized Code for any given opening, [list here](https://www.365chess.com/eco.php));
* Opening Name;
* Opening Ply (Number of moves in the opening phase)

# Methodologies used:

In this section you should explain your project steps in details, write the name of your project methodologies or techniques used and how and why you use them.

**Step 1: Exploring the dataset:**

We explored all the features in the dataset , to get to understand the data before working on it , and here are some of our findings:

* The start time and the end time are in unix format
* Over 40% of the entries have the same time in both date started and date ended, which means we can’t calculate the some of games time.
* The id of the games and players is useless in our analysis
* Opening names contains the main openings and the variations , which makes them a lot, it would be better to work with the main openings only.
* the data contains a lot of categorical data, and some of the numerical data are highly skewed
* data contains no null values, and no duplicated data

**Step 2: cleaning the data**

* We removed the useless columns game id, white id, black id
* Separated the game moves to de analyzed separately
* Cleared the variations from opening names
* Removed the created at and end time from the data , because of the problem mentioned in step 1
* Separated the increment code into time control and increment

Features in data after cleaning :

# rated -> true or false

# turns -> 1 : 349

# victory status -> draw, mate, outoftime, resign

# winner -> white , black, draw

# time control -> 1 : 180

# increment -> 0 : 180

# white rating, black rating -> 780 : 2700

# main opening -> 277 unique openings

# opening ply -> 1:28

**Step 3: preparing the data**

In this step we had to prepare data for clustering, modeling and association rules.

First preparing the data for clustering and modeling:

In order to preform clustering and modeling we had to convert some important categorical features into numerical values,

And choose the features that will contribute the most to the process, so we did the following:

* Convert rated from TRUE, False to 1, 0
* Convert winner to 0 for white, 1 for black, 0.5 for draw
* Do one hot encoding to victory status , by creating dummy variables
* Do count encoding to main openings, the choice of count encoding here was to avoid high dimensionality from one hot encoding and to classify openings by their relevance.

For clustering decided to :

* Use the mean rating instead of white rating and black rating to reduce the variance
* Use the rating difference of the winner – loser or negative half of the difference if draw

For supervised modeling we decided to :

* Normalize the data by taking the log of the highly skewed numerical values.
* Scaling the numerical values to ensure all the features are in the same scale.

**Step 4: unsupervised learning:**

The main goal here was to find relations between games and split them into groups

* After plotting the data the variance was very high so we applied PCA to reduce the variance
* The data seemed to fit the most with 3 centroids
* We preformed kmeans clustering
* Explored the results to find the relations within each cluster
* The clusters seemed to split based on the winner without too much information so we preformed clustering again on the data but without the winner, which gave us a better insights.

**Step 5: supervised learning**

The main goal here was to train a model that predicts a winner without the game moves

* First step was to split the data into training data and testing data we choose it to be 7:3
* Then we choose 2 models to train them decision trees and naïve bayes.
* After training the models on train data, we tested them on test data
* We then got the results which will be discussed on the outcome chapter of the report

**Step 6: Associations**

The main goal here was to find some association rules in the moves of the openings using apriori algorithm

* First we extract the opening moves from the moves of the each game
* We convert it to transactions
* Then we apply apriori and display the results

**Step 7: UI**

* First I created the “App.R” to be the main file of the shiny app.
* I created the “ui.R” file to handle the ui of the app.
* In it, there’s a sidebar with two dropdowns to choose from the graphs created.
* The “sever.R” handles the main logic of our app.
* It controls the content of the sidebar dropdowns and plots the graphs onto the main area.
* The “getGraphs.R” contains functions to plot the graphs for the different models and visualization.
* getUnsupervised plots the Kmeans model with the data using the “factoextra” library.
* “plot\_results” mutates the data so the white and black ratings are classified into 100 interval levels for easier visualization organizes them and plot the results of the win/loss/draw for white and black accordingly.
* “most\_openings” gets the top 20 played openings and plots them.
* “calculate\_time” categorizes the ratings into 100 intervals again and combines the two columns into a new one with the time of each one and then calculates and plots the average time for each bracket.
* “get\_supervised” plots the naïve bayes and decision tree model results side by side
* The “getGraphs.R” file relies on the work done in the “DSfinal.R

# Challenges in the dataset:

In this section you should write the difficulties and challenges you face while working on your dataset.

1. The data had a lot of categorical features.
2. Most of the numerical features are skewed and had a high variance.
3. The data is pretty complex .
4. Finding the relations in each cluster was challenging.
5. R tends to make work harder , it would be better if we can work with python

# Interpretations of the results

In this section you should write the results, its explanation and show the plotted graphs.

Results of clustering:

A graph of a cluster plot

Description automatically generated

After preforming kmeans clustering to the data while including winners it was apparent that:

* The first cluster represented the draw games, and usually has high time control , players tend to stick to opening moves and they have the highest average rating.
* While the second cluster represented the games won by white, games has less turns and have high rating difference, and Usually games ends by mate or resign.

* And the last cluster represented the games won by the black side, games has less time control , most games ends by mate or resign but most of the games that ends by out of time are in this clusterA graph showing different colored shapes

  Description automatically generated

While preforming kmeans clustering on the data without including the winners resulted in :

The first cluster having games always end up in a mate , lower time control in average , players tend to stick less to opening moves and the average rating of players is low.

The second cluster having games end up with either a draw or out of time, the highest turns, the average player rating is high.

The last cluster having games end with one of the players resigning, the lowest average, turns yet the highest time control , players in this cluster lean to stick to opening moves.

Results of modeling:

A graph of a number of different colored squares

Description automatically generated with medium confidence

The two models gave similar results with naïve bayes giving a slightly better accuracy.

Decision trees balanced accuracy:

White: %61.8 Draw: %98 Black: %60.71

Naïve Bayes balanced accuracy:

White: %63.16 Draw: %98 Black: %61.87

Results from arules:

A graph with red dots and black text

Description automatically generated

Results from exploring the data:

A graph of a graph

Description automatically generated with medium confidence

A graph of a number of bars

Description automatically generated with medium confidence

A graph of a number of bars

Description automatically generated with medium confidence

# Conclusion:

In this section you should highlight the major findings.

According to the games in this dataset:

* The most famous opening is Sicilian defense.
* The higher the rating the less time control they prefer.
* Higher rated players stick the most to the opening moves and usually play famous openings.
* Games that end in a draw usually have high time control.
* It’s hard to accurately guess the winner of chess games , as sometimes higher rated players lose or draw with lower rated players.