Logistic Regression in Chess Matches

Game Prediction Theory

Jessica Erasmus
October 2023

Table of Contents

Introduction 2
Context 2
Requirements 2
Logistic Regression 2
- Binary Classification 3
- Log-odds Transformation 3
- Model Parameters 3
- Maximum Likelihood Estimation 3
- Model Interpretability 3
- Decision Boundary 3
- Evaluation 3

Dataset 4
- Sample Data 4
- Dataset Links 4
  - Sample Dataset 4
  - Actual Datasets 4
- Dataset Preview 4

Data Wrangling 6
Exploratory Data Analysis 6
Data Preparation 11
- Target variable 11
- Feature Variables 11
  - String Indexer 11
  - One-Hot Encoder 11
  - Vector Indexer 11

Logistic Regression Model 12
References 14

Introduction

The prediction of the outcomes of games, matches, or events has become an intriguing and vital endeavour in the ever-evolving context of the gaming industry. Game prediction has sparked the interest of both players and stakeholders in the gaming environment, whether it's predicting the winner of a competitive esports' tournament, anticipating the success of a new video game release, or estimating the likelihood of a specific in-game event occurring. To address these challenges and provide valuable insights, data-driven techniques have grown increasingly popular, and logistic regression is included as one of these many effective instruments from the data analysis tool collection.

Logistic regression is a statistical modelling technique used to make binary and multi-class predictions in a variety of fields, such as the gaming industry. It is especially well-suited for predicting game outcomes because it enables users to understand the relationship between one or more independent variables and the likelihood of a specific event occurring. In the context of gaming, these events can range from the likelihood of a team winning a match to the likelihood of a player reaching a specific in-game milestone.

The findings from this analysis explores into the mutually beneficial connection between game prediction and logistic regression, bringing insight into the value of this statistical approach in deciphering complex gaming statistics. This technique can help gamers, gamblers, analysts, and game sponsors acquire deeper insights and make informed decisions by investigating the principles of logistic regression and its applications within the gaming domain. The fusion of game prediction and logistic regression offers an exciting approach to unravelling the mysteries of the gaming world, whether for competitive players looking for an edge on their future opponents or investors and spectators looking to understand player behaviour.

Context

An interesting application of logistic regression is predicting the outcome of a chess game, such as whether player white will win or not. For this analysis, a chess prediction model is built using logistic regression. The model predicts the likelihood of the player using white chess pieces' chances of winning based on both players' Elo rating and player white's difference in ranking from player black. Hence, the assumption made here is that logistic regression is a powerful statistical technique which may be used for analysing the various factors that influence the outcome of a chess match.

Requirements

The task involves developing and improving a logistic regression model as well as evaluating the model's accuracy. This model will be implemented on a dataset that is suitable for performing logistic regression as well as contains more than 500000 entries. The analysis must make use of Spark and be perform in Jupyter Notebook. The dataset must be an open data set and the main language is python.

Logistic Regression

Logistic regression is a statistical modelling technique used for binary classification, which means it predicts one of two outcomes. When the dependent variable is binary (has only two possible outcomes), this type of regression analysis is used. The goal of logistic regression is to find the best-fitting model that can predict whether the dependent variable will fall into one of two categories based on the independent variables. A logistic function is used in the model to convert the output of a linear regression model into a probability value between 0 and 1.

The logistic function is an S-shaped curve with a starting value of 0 and an ending value of 1, and it is used to model the probability of the dependent variable falling into one of two categories. This method is especially useful when determining the likelihood of a categorical outcome based on one or more independent variables. Logistic Regression considers the following instances.

Binary Classification

In logistic regression, the dependent variable (the one you're trying to predict) is categorical and binary. For instance, it can represent outcomes like "yes" or "no," "0" or "1," "fraudulent" or "non-fraudulent," "win" or "lose,". The goal is to understand the relationship between this binary outcome and one or more independent variables.

Log-odds Transformation

Logistic regression uses a logistic function, also known as the sigmoid function, to transform the linear combination of the independent variables into a value between 0 and 1. The logistic function, denoted as "σ," is defined as:

$$\sigma(z) = 1 / (1 + e^{(-z)})$$

"z" represents the linear combination of independent variables. The transformed value "σ(z)" is interpreted as the probability of the event occurring (e.g., the probability of winning a game or the probability of an email being spam).

Model Parameters

Logistic regression involves estimating model parameters that determine the shape of the sigmoid curve. These parameters include coefficients (β) associated with each independent variable. By adjusting these coefficients, the model tries to fit the data and make accurate predictions.

Maximum Likelihood Estimation

The logistic regression model is trained using a technique called Maximum Likelihood Estimation (MLE). MLE finds the values of the model parameters that maximize the likelihood of the observed data given the model. In simpler terms, it tries to find the model that makes the observed outcomes the most probable.

Model Interpretability

One of the advantages of logistic regression is its interpretability. You can assess the impact of each independent variable on the probability of the binary outcome by examining the estimated coefficients. A positive coefficient indicates that as the variable increases, the probability of the event also increases, while a negative coefficient suggests the opposite.

Decision Boundary

The logistic regression model creates a decision boundary that separates the two classes in the feature space. This boundary is defined by the equation σ(z) = 0.5, and it determines which side of the boundary an observation falls on, thereby making the binary prediction.

Evaluation

Once the model is trained, it needs to be evaluated using appropriate metrics, such as accuracy, precision, recall, F1-score, and the receiver operating characteristic (ROC) curve. These metrics assess the model's performance in making accurate binary predictions.

As a result, logistic regression is widely used in a variety of fields, including healthcare (predicting disease risk), finance (credit scoring), marketing (predicting customer churn), and as previously mentioned, gaming (predicting game outcomes). Because of its simplicity, interpretability, and well-established mathematical foundations, it is a fundamental tool for binary classification problems. Although it is limited to binary outcomes, extensions such as multinomial logistic regression can also be used to solve problems with more than two categories.

Dataset

The dataset chosen for this task is based on real chess games which are collected from the Lichess Elite Database from August and September 2023. It contains over 500000 games which contains match data and statistics.

This dataset is suitable for logistic regression as it contains data which has player and match statistics and match outcomes which help to predict future game outcomes. It is then predicted if white or black will win.

Sample Data

For this analysis display, a sample dataset is used for the analysis, which uses a section of a dataset which contains chess matches taken from September 2020. This is to test the initial output of the code before using the newer dataset which require conversion from PGN to csv. The code is provided for the conversion however and the games file will be used from the two output files of the code.

Dataset Links

Sample Dataset

https://www.kaggle.com/datasets/sahit2509/chess-dataset-100000-games-lichess

Actual Datasets

https://database.nikonoel.fr/

Dataset Preview

The dataset initially looks like this when input into spark. There is nothing wrong with this however if we consider the number of columns and differing lengths. Hence the dataset is fine.

```
+---+--------------------+----------+-----+----------------+--------------------+-------+--------+---+--------------------+---
---------+-----------+--------+----------------+
|_c0|               Event|      Date|Round|           White|               Black| Result|BlackElo|ECO|             Opening| Te
rmination|TimeControl|WhiteElo|WhiteRatingDiff|
+---+--------------------+----------+-----+----------------+--------------------+-------+--------+---+--------------------+---
---------+-----------+--------+----------------+
|  0|Rated Blitz tourn...|2020.09.01|    -|   AttackSparrow|            danicuva|    1-0|    2218|C00|French Defense: S...|Tim
e forfeit|      180+0|    2460|               2|
|  1|    Rated Blitz game|2020.09.01|    -|      onthewaygm|         starkspieler|    1-0|    2424|E90|King's Indian Def...|
Normal|      180+0|    2428|               6|
|  2|Rated Rapid tourn...|2020.09.01|    -|        OjaiJoao|     FitzwilliamDarcy|    1-0|    2300|B06|Modern Defense: S...|
Normal|      600+5|    2441|               5|
|  3|Rated Blitz tourn...|2020.09.01|    -| WenceslaoRodrigo|            zonrobla|    0-1|    2667|E71|King's Indian Def...|
Normal|      180+1|    2280|              -2|
|  4|    Rated Blitz game|2020.09.01|    -|        HoldenHc|           gg-gm-gmg|    1-0|    2682|A41|        Queen's Pawn|
Normal|      180+0|    2557|               8|
|  5|    Rated Blitz game|2020.09.01|    -|        Evgen_88|               JLeon|    0-1|    2552|E35|Nimzo-Indian Defe...|
Normal|      180+0|    2554|              -5|
|  6|    Rated Blitz game|2020.09.01|    -|          Napo18|        Nice_Ice_Eyes|    1-0|    2377|B01|Scandinavian Defe...|
Normal|      180+0|    2558|               3|
|  7|Rated Blitz tourn...|2020.09.01|    -|        spidernv|   Quepaseelquesigue|    1-0|    2362|A10|     English Opening|
Normal|      180+2|    2412|               5|
|  8|Rated Blitz tourn...|2020.09.01|    -|        vangulio|         GlennTipton|    0-1|    2445|C91|Ruy Lopez: Closed...|
Normal|      180+0|    2437|              -6|
|  9|Rated Blitz tourn...|2020.09.01|    -|Club-Jaque-al-Rey|         emmacristal|    1-0|    2331|B40|Sicilian Defense:...|
Normal|      180+0|    2539|               3|
| 10|    Rated Blitz game|2020.09.01|    -|      Elhlwagy11|     Mondesespoir2700|1/2-1/2|    2425|C13|French Defense: C...|
Normal|      180+0|    2309|               2|
| 11|    Rated Blitz game|2020.09.01|    -|         LouiVos|zpxocivubyntmraeswdq|    1-0|    2685|E16|Queen's Indian De...|
Normal|      180+0|    2743|               5|
| 12|    Rated Blitz game|2020.09.01|    -|     Dimitriy1975|              mixo23|    0-1|    2418|A43|Benoni Defense: B...|
Normal|      180+0|    2386|              -5|
| 13|Rated Blitz tourn...|2020.09.01|    -|          Fletov|         espinozasgod|    0-1|    2443|A45|Trompowsky Attack...|Tim
e forfeit|      180+0|    2235|              -2|
| 14|Rated Blitz tourn...|2020.09.01|    -|  venadorecargado|           Derrotado|    0-1|    2409|B00|        Ware Defense|Tim
e forfeit|      180+0|    2281|              -3|
| 15|Rated Blitz tourn...|2020.09.01|    -| juancruzariasTDF|             Tenessy|    1-0|    2367|B23|Sicilian Defense:...|
Normal|      180+0|    2431|               5|
| 16|    Rated Blitz game|2020.09.01|    -|   alirezafirulais|          Phoenix-20|    1-0|    2493|D19|Queen's Gambit De...|
Normal|      180+0|    2347|               8|
| 17|    Rated Blitz game|2020.09.01|    -|            Glig|            Al_Shima|    0-1|    2492|B02|Alekhine Defense:...|
Normal|      180+0|    2463|              -5|
| 18|    Rated Blitz game|2020.09.01|    -|     self_service|             Atomrod|    1-0|    2499|A35|English Opening: ...|Tim
e forfeit|      180+0|    2425|               7|
```

Hence the initial data frame schema is shown below as well as the data frame transposed in pandas.

```
root
 |-- _c0: integer (nullable = true)
 |-- Event: string (nullable = true)
 |-- Date: string (nullable = true)
 |-- Round: string (nullable = true)
 |-- White: string (nullable = true)
 |-- Black: string (nullable = true)
 |-- Result: string (nullable = true)
 |-- BlackElo: integer (nullable = true)
 |-- ECO: string (nullable = true)
 |-- Opening: string (nullable = true)
 |-- Termination: string (nullable = true)
 |-- TimeControl: string (nullable = true)
 |-- WhiteElo: integer (nullable = true)
 |-- WhiteRatingDiff: integer (nullable = true)
```

	0	1	2	3	4
_c0	0	1	2	3	4
Event	Rated Blitz tournament https://lichess.org/tou...	Rated Blitz game	Rated Rapid tournament https://lichess.org/tou...	Rated Blitz tournament https://lichess.org/tou...	Rated Blitz game
Date	2020.09.01	2020.09.01	2020.09.01	2020.09.01	2020.09.01
Round	-	-	-	-	-
White	AttackSparrow	onthewaygm	OjaiJoao	WenceslaoRodrigo	HoldenHc
Black	danicuva	starkspieler	FitzwilliamDarcy	zonrobla	gg-gm-gmg
Result	1-0	1-0	1-0	0-1	1-0
BlackElo	2218	2424	2300	2667	2682
ECO	C00	E90	B06	E71	A41
Opening	French Defense: Schlechter Variation	King's Indian Defense: Normal Variation, Rare ...	Modern Defense: Standard Defense	King's Indian Defense: Makogonov Variation	Queen's Pawn
Termination	Time forfeit	Normal	Normal	Normal	Normal
TimeControl	180+0	180+0	600+5	180+1	180+0
WhiteElo	2460	2428	2441	2280	2557
WhiteRatingDiff	2	6	5	-2	8

Data Wrangling

There was not much data to clean, as no values were null in the dataset however, a few columns were removed to make the data look neater.

```
+---+-----+----+-----+-----+-----+------+--------+---+-------+-----------+-----------+--------+---------------+
|_c0|Event|Date|Round|White|Black|Result|BlackElo|ECO|Opening|Termination|TimeControl|WhiteElo|WhiteRatingDiff|
+---+-----+----+-----+-----+-----+------+--------+---+-------+-----------+-----------+--------+---------------+
|  0|    0|   0|    0|    0|    0|     0|       0|  0|      0|          0|          0|       0|              0|
+---+-----+----+-----+-----+-----+------+--------+---+-------+-----------+-----------+--------+---------------+
```

A new column was added to clarify who won as well.

```
# The unnamed and Round columns are repetitive and do not provide much information, hence the columns are dropped
df = df.drop('_c0', 'Round')
```

```
df = df.withColumn('Winner', when(df['Result'] == '1-0', 'White').when(df['Result'] == '0-1', 'Black').otherwise('Draw'))
```

```
# Print shape of dataframe
print((df.count(), len(df.columns)))
```

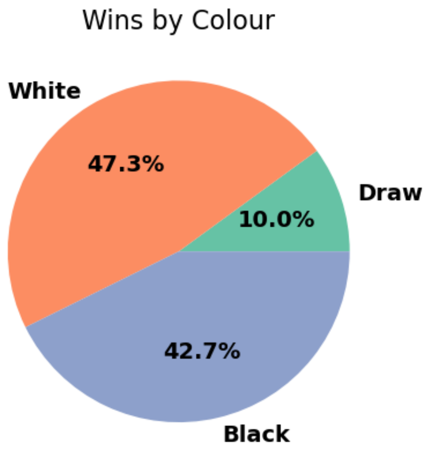
```
(99913, 13)
```

```
df.printSchema()
```

```
root
 |-- Event: string (nullable = true)
 |-- Date: string (nullable = true)
 |-- White: string (nullable = true)
 |-- Black: string (nullable = true)
 |-- Result: string (nullable = true)
 |-- BlackElo: integer (nullable = true)
 |-- ECO: string (nullable = true)
 |-- Opening: string (nullable = true)
 |-- Termination: string (nullable = true)
 |-- TimeControl: string (nullable = true)
 |-- WhiteElo: integer (nullable = true)
 |-- WhiteRatingDiff: integer (nullable = true)
 |-- Winner: string (nullable = false)
```

Exploratory Data Analysis

The basic statistics are look at first. White has most win while draws are the least. The Elo ratings also have similar statistics results as well.



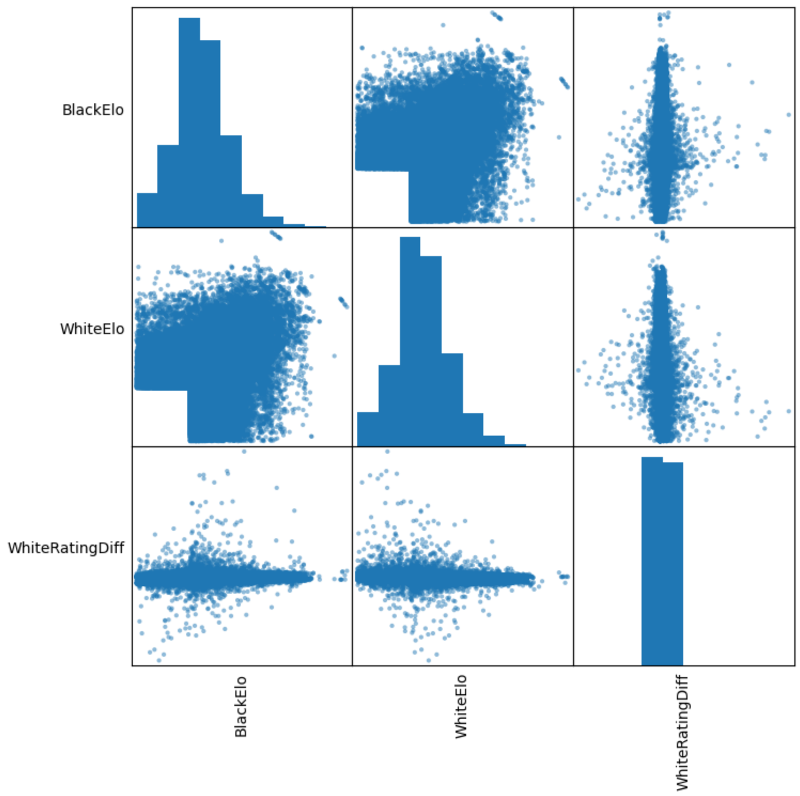

```
df.toPandas().groupby(['Winner']).size()
```

```
Winner
Black     42645
Draw      10023
White     47245
dtype: int64
```

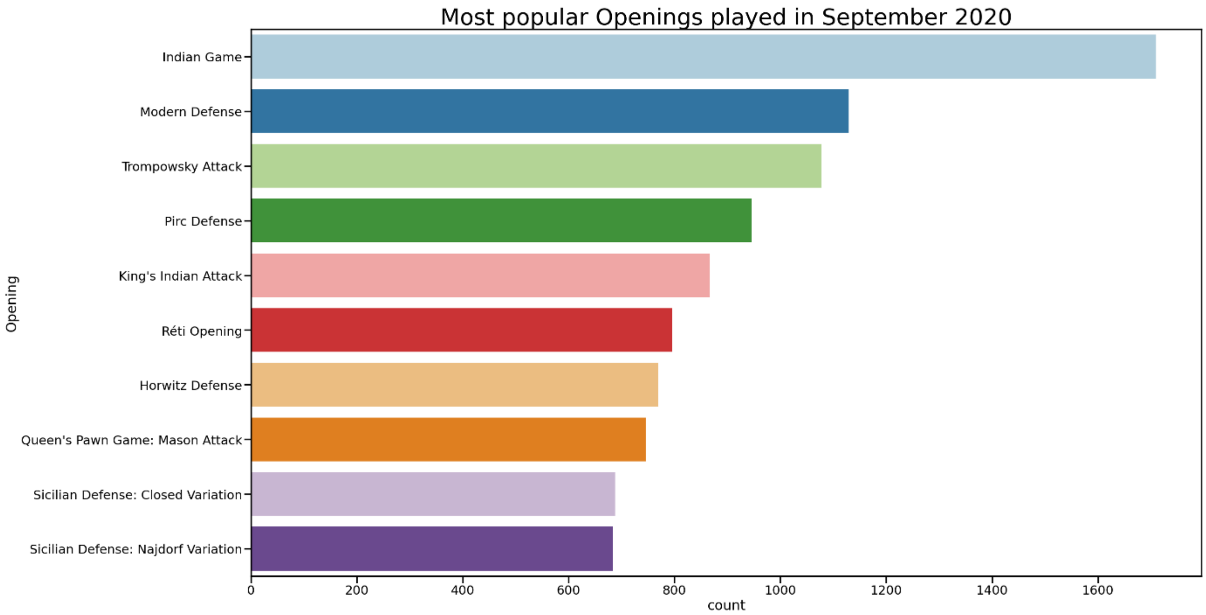
```
# Describe the Spark DataFrame
numeric_features = [t[0] for t in df.dtypes if t[1] == 'int']
df.select(numeric_features).describe().toPandas().transpose()
```

	0	1	2	3	4
summary	count	mean	stddev	min	max
BlackElo	99913	2443.3011620109496	99.56966156291874	2200	2988
WhiteElo	99913	2443.180897380721	99.4675319877633	2200	2986
WhiteRatingDiff	99913	0.3830832824557365	6.667514224211187	-152	233

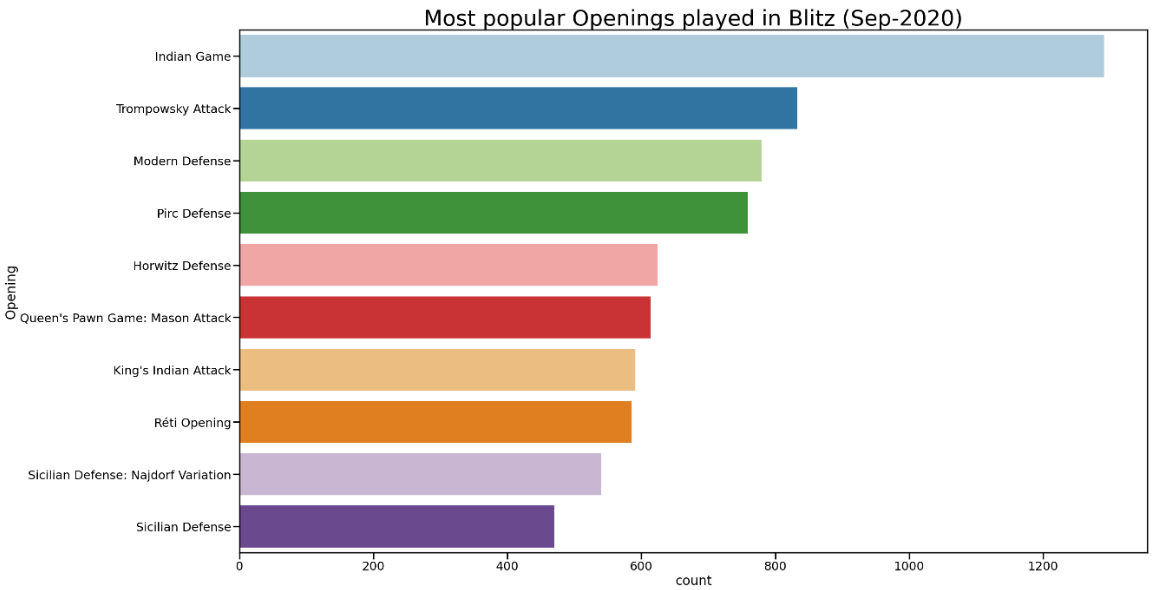
In the scatter matrix, black Elo ratings and White Elo ratings have similar correlations with the White Rating difference.

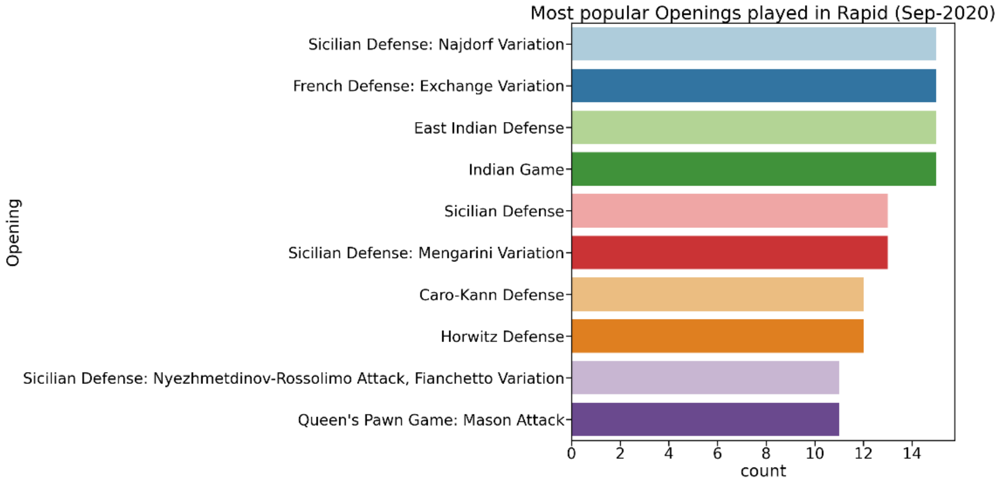


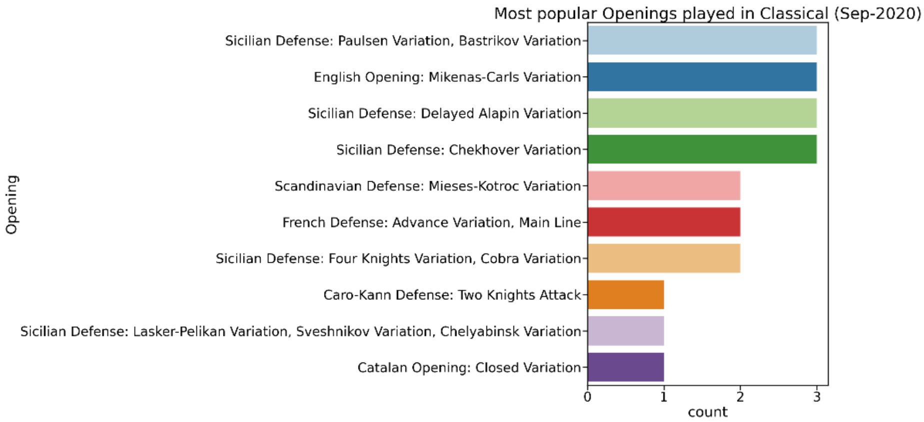
Next, we look at the counts of Openings played at different chess events and get out top 10 popular openings.



Indian Game is the most popular chess opening, next is looking at the openings per event.





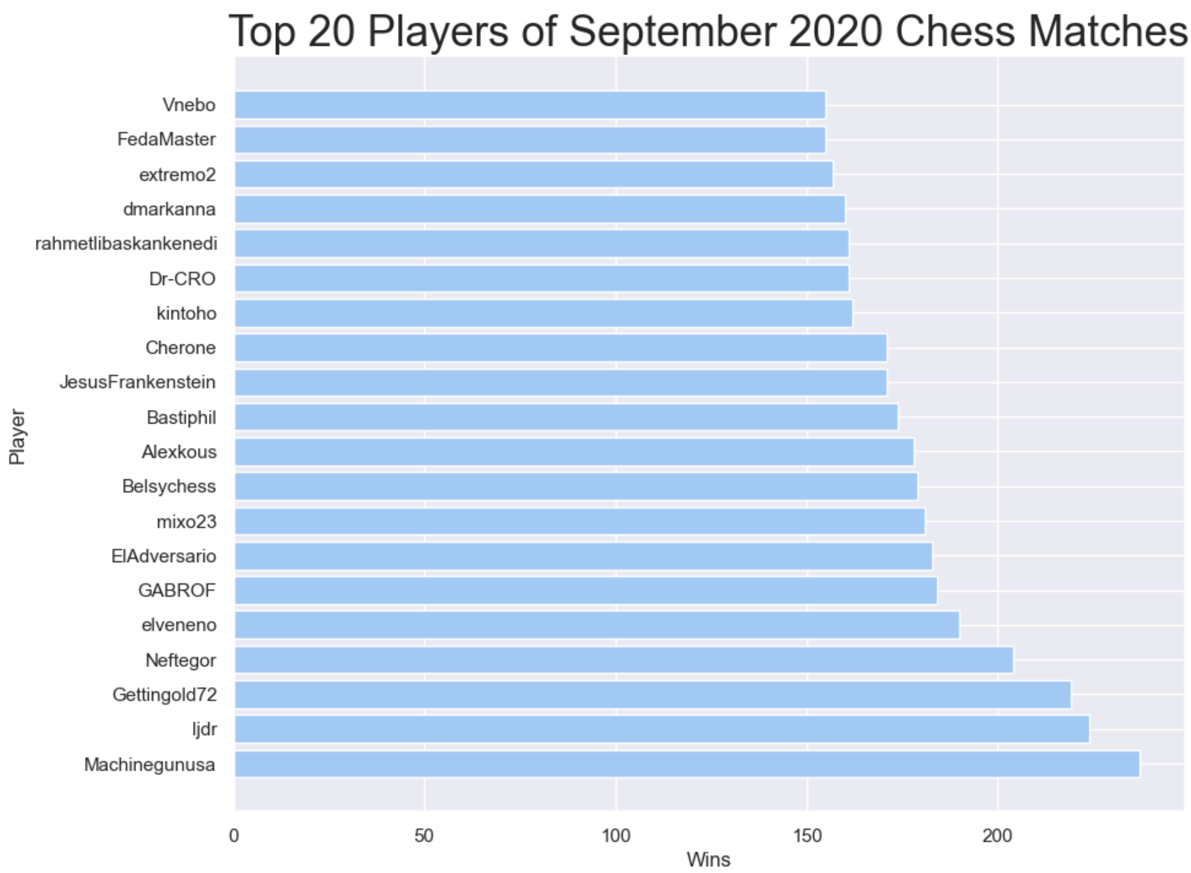


Indian Game is popular in the first three events' top 10. The Sicilian Defense strategies are also popular in all three events.

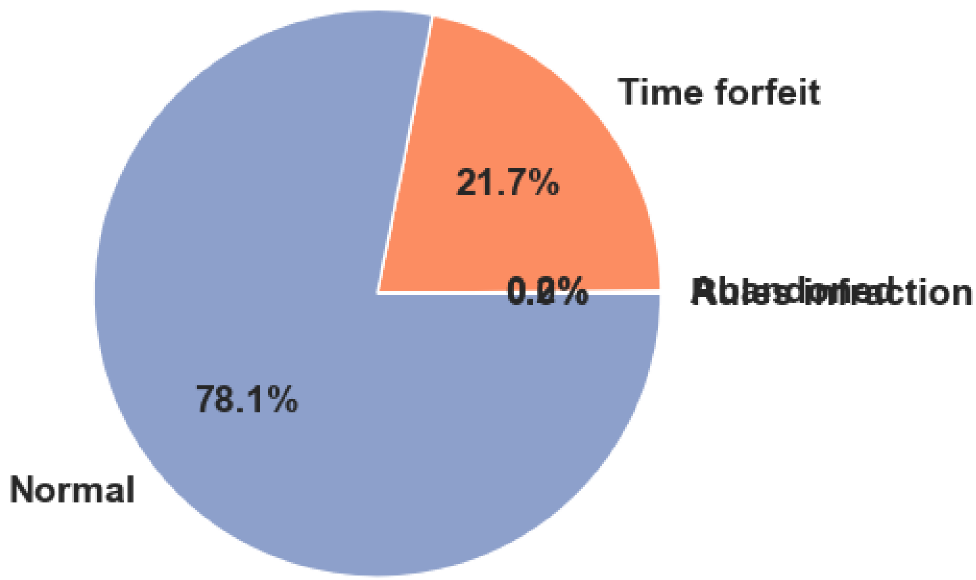
Next, the popular players are explored. Machinegunusa seems to be in number one spot.

Player	White Wins	Black Wins	Total Wins
Machinegunusa	124	113	237
ljdr	116	108	224
Gettingold72	113	106	219
Neftegor	110	94	204
elveneno	94	96	190
GABROF	92	92	184
ElAdversario	97	86	183
mixo23	94	87	181
Belsychess	90	89	179
Alexkous	93	85	178
Bastiphil	91	83	174
JesusFrankenstein	76	95	171
Cherone	84	87	171
kintoho	74	88	162
Dr-CRO	82	79	161
rahmetlibaskankenedi	88	73	161
dmarkanna	81	79	160
extremo2	84	73	157
Vnebo	87	68	155
FedaMaster	85	70	155

only showing top 20 rows



We look at the different ways the game ends.



Normal is the most common way the game ends, followed by time forfeit and other is less than one percent.

Data Preparation

For this section, a new data frame is made based of the old one. It holds the following values for analyses.

```
root
 |-- BlackElo: integer (nullable = true)
 |-- ECO: string (nullable = true)
 |-- Opening: string (nullable = true)
 |-- TimeControl: string (nullable = true)
 |-- WhiteElo: integer (nullable = true)
 |-- WhiteRatingDiff: integer (nullable = true)
 |-- Winner: integer (nullable = false)
```

Target variable

Winner will be used as the target variable as this is the outcome that needs to be predicted.

Feature Variables

In logistic regression, the independent variables are called feature variables. These variables are used to predict the probability of the dependent variable being in one of the two categories based on the independent variables. The model uses a logistic function to transform the output of a linear regression model into a probability value between 0 and 1.

Feature variables are created using OneHotEncoder, StringIndexer and VectorAssembler.

String Indexer

A data frame's category string columns can be converted into numerical indexes with the support of the StringIndexer. Most machine learning algorithms cannot deal directly with string data; hence this translation is required.

One-Hot Encoder

Since categorical features are transformed into dummy features, one-hot encoding is also known as dummy encoding. One or more categorical characteristics can be converted into numerical dummy features that are helpful for training machine learning models using one-hot encoding.

Vector Indexer

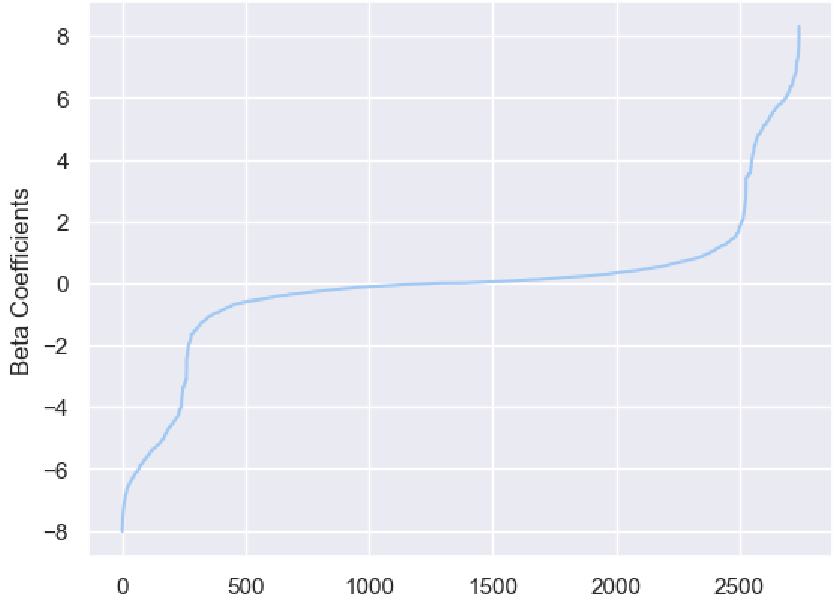
Vector Indexer facilitates the indexing of categorical characteristics within vector datasets. It can convert original values to category indices and automatically determining which features are categorical.

```
root
 |-- label: double (nullable = false)
 |-- features: vector (nullable = true)
 |-- BlackElo: integer (nullable = true)
 |-- ECO: string (nullable = true)
 |-- Opening: string (nullable = true)
 |-- TimeControl: string (nullable = true)
 |-- WhiteElo: integer (nullable = true)
 |-- WhiteRatingDiff: integer (nullable = true)
 |-- Winner: integer (nullable = false)
```

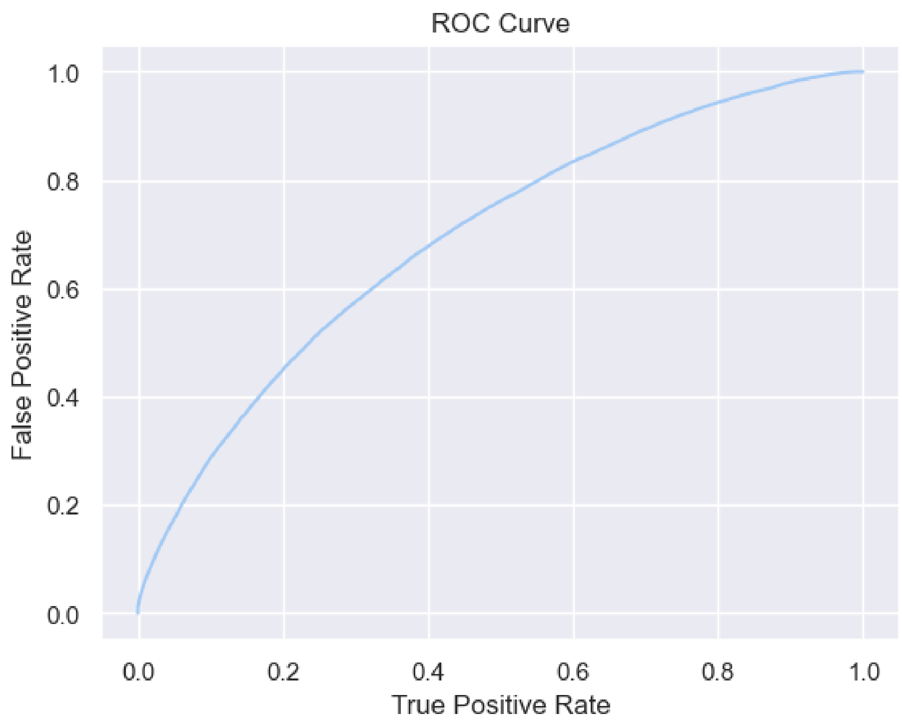
	label	features
0	1.0	(0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
1	1.0	(0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
2	1.0	(0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, ...
3	0.0	(0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
4	1.0	(0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...

Logistic Regression Model

After splitting the dataset, the model is built. The results for the model are shown below.



This graph above shows the relationship between the beta coefficients of the dataset. It develops an interesting trend where the curve starts increasing steadily until around (2000,4) before it starts increasing more sharply until it reaches 8 as a beta coefficient.



The roc curve is decent however could be better. The logistic regression model accuracy of 65% and 67% after hyper tuning. The logistic regression model was compared with other models but has the highest accuracy. This concludes that Elo rating may not be the main variable to influence a win.

References

Binkhonain, M. & Zhao, L., 2019. A review of machine learning algorithms for identification and classification of non-functional requirements. *Expert Systems with Applications: X.*

Gudivada, V. N., Apon, A. & Ding, J., 2017. Data Quality Considerations for Big Data and Machine Learning: Going Beyond Data Cleaning and Transformations. *International Journal on Advances in Software,* 10(1), pp. 1-20.

Hernandez, L. & Quinteiro, J., 2019. *Binary Classification with PySpark and MLlib.* [Online]
Available at: https://www.kaggle.com/code/palmer0/binary-classification-with-pyspark-and-mllib
[Accessed 15 Oct 2023].

Jankowska, K., 2020. *Binary classification - multiple method comparison.* [Online]
Available at: https://www.kaggle.com/code/klaudiajankowska/binary-classification-multiple-method-comparison
[Accessed 15 Oct 2023].

Lubis, A. M., 2020. *Chess Game (EDA + Predict Winner Modelling).* [Online]
Available at: https://www.kaggle.com/code/codingan/chess-game-eda-predict-winner-modelling/notebook
[Accessed 15 Oct 2023].

Muller, A. C. & Guido, S., 2018. *Introduction to Machine Learning with Python.* Forth ed. Sebastopol, CA: O'Reilly Media, Inc.

Nick, 2020. *Chess dataset with EDA and Logistic Regression.* [Online]
Available at: https://www.kaggle.com/code/zingo3245/chess-dataset-with-eda-and-logistic-regression/notebook
[Accessed 15 Oct 2023].

Ray, S., 2020. *Improve Your Model Performance using Cross Validation (in Python and R).* [Online]
Available at: https://www.analyticsvidhya.com/blog/2018/05/improve-model-performance-cross-validation-in-python-r/
[Accessed 17 Oct 2023].

SARAHG, 2017. *Titanic Analysis_Learning to Swim with Python.* [Online]
Available at: https://www.kaggle.com/code/sgus1318/titanic-analysis-learning-to-swim-with-python/notebook
[Accessed 15 Oct 2023].

Sharma, S., 2020. *Chess_dataset (100,000 games) Lichess.* [Online]
Available at: https://www.kaggle.com/datasets/sahit2509/chess-dataset-100000-games-lichess
[Accessed 15 Oct 2023].

Sharma, S., 2020. *Chessmeta.* [Online]
Available at: https://www.kaggle.com/code/sahit2509/chessmeta/notebook
[Accessed 15 Oct 2023].

Sherif, A. & Ravinda, A., 2018. *Apache Spark Deep Learning Cookbook.* Birmingham: Packt Publishing.

TRUSTTHEDATA, 2018. *Bank Customer Churn Prediction.* [Online]
Available at: https://www.kaggle.com/code/kmalit/bank-customer-churn-prediction
[Accessed 15 Oct 2023].

Turing, 2023. *Different Types of Cross-Validations in Machine Learning and Their Explanations.* [Online]
Available at: https://www.turing.com/kb/different-types-of-cross-validations-in-machine-learning-and-their-explanations
[Accessed 17 Oct 2023].

Verma, N., 2017. *Comparing Various ML models(ROC curve comparison).* [Online]
Available at: https://www.kaggle.com/code/nirajvermafcb/comparing-various-ml-models-roc-curve-comparison
[Accessed 15 Oct 2023].