Report 1: exploration, data visualization and data pre-processing report

Introduction to the project

Objectives

- **What are the main objectives to be achieved? Describe in a few lines.**

The objective of this project is to try to beat bookmakers' algorithms on estimating the probability of a team winning a match. I'll go through all the steps, like cleaning up data, selecting important features, and making models that can guess the chances of a player winning. At last, I'll try to check if my model can predict the match outcome better than the bookmakers.

- **For each member of the group, specify the level of expertise around the problem addressed?**

I worked on this alone. I'm completely new to Data Analysis.

- **Have you contacted business experts to refine the problem and the underlying models? If yes, detail the contribution of these interactions.**

No.

- **Are you aware of a similar project within your company, or in your entourage? What is its progress? How has it helped you in the realization of your project? How does your project contribute to improving it?.**

Not relevant.

Understanding and manipulation of data

Framework

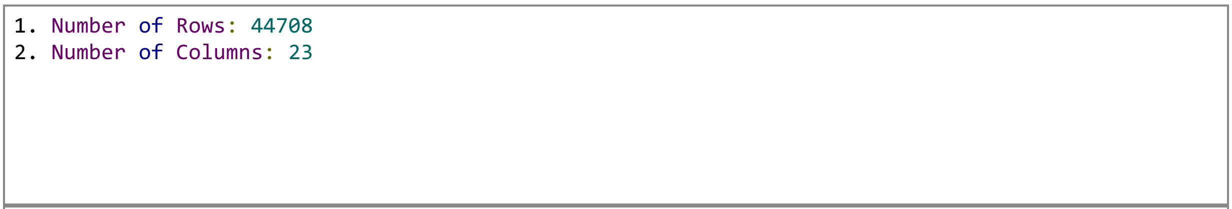
- **Which set(s) of data(s) did you use to achieve the objectives of your project?**

I used the atp_data.csv available from https://www.kaggle.com/edouardthomas/atp-matches-dataset

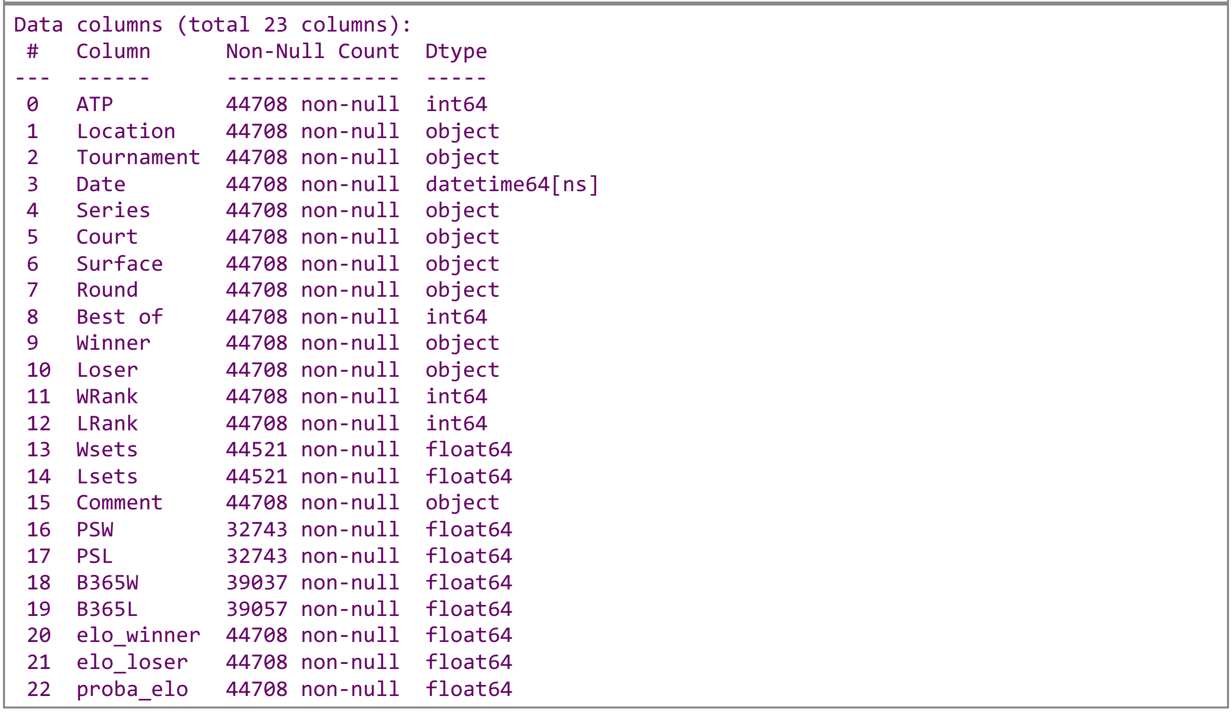
- **Are these data freely available? If not, who owns the data?**

Yes, the data can be freely downloaded.

- **Describe the volume of your dataset?**



```
1. Number of Rows: 44708
2. Number of Columns: 23
```



```
Data columns (total 23 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   ATP         44708 non-null  int64
 1   Location    44708 non-null  object
 2   Tournament  44708 non-null  object
 3   Date        44708 non-null  datetime64[ns]
 4   Series      44708 non-null  object
 5   Court       44708 non-null  object
 6   Surface     44708 non-null  object
 7   Round       44708 non-null  object
 8   Best of     44708 non-null  int64
 9   Winner      44708 non-null  object
 10  Loser       44708 non-null  object
 11  WRank       44708 non-null  int64
 12  LRank       44708 non-null  int64
 13  Wsets       44521 non-null  float64
 14  Lsets       44521 non-null  float64
 15  Comment     44708 non-null  object
 16  PSW         32743 non-null  float64
 17  PSL         32743 non-null  float64
 18  B365W       39037 non-null  float64
 19  B365L       39057 non-null  float64
 20  elo_winner  44708 non-null  float64
 21  elo_loser   44708 non-null  float64
 22  proba_elo   44708 non-null  float64
```

Relevance

- **Which variables seem most relevant to you with regard to your objectives?**

  - ***WRank*** *(Winner's Ranking): A higher ranking can indicate a stronger player.*
  - ***LRank*** *(Loser's Ranking): Similar to the winner's ranking.*
  - ***Wsets*** *(Winner's Sets Won): Number of sets won by the winner.*
  - ***Lsets*** *(Loser's Sets Won): Number of sets won by the loser.*
  - ***elo_winner*** *(Winner's Elo Rating): Elo ratings represent the skill level of the winner and can be highly relevant for predicting outcomes, especially if there's a significant difference in ratings.*
  - ***elo_loser*** *(Loser's Elo Rating): Similar to elo_winner.*
  - ***proba_elo*** *(Elo-Based Probability): This variable directly relates to the project objective of estimating the probability of winning based on Elo ratings.*
  - ***Surface*** *(Match Surface): The type of surface (e.g., clay, grass, hard, carpet) can be a factor that influences match outcomes, as players may have different performance levels on various surfaces.*
  - ***Best of*** *(Match Format): The format of the match (e.g., best of 3 or best of 5) could affect the dynamics of the match.*

**** Additional info on Elo ratings (from the web)*:**

Elo ratings are a system for estimating the relative skill levels of players or teams in two-player games, such as chess and sports like tennis. The Elo rating of a player or team is a numerical value *that quantifies their skill or performance. A higher Elo rating typically indicates a stronger player or team, while a lower rating suggests a weaker one.*

"proba_elo" is a derived variable that uses the Elo ratings of two opponents to estimate the probability of one of them winning a match. In sports analytics, this is often done using a mathematical formula or model that considers the difference in Elo ratings between the opponents. The estimated probability is used to predict the outcome of a match. For example, if "proba_elo" indicates a 0.7 probability for Player A, it suggests that Player A is likely to win the match with a 70% chance.

- **What is the target variable?**

My first choice was to use the estimated probability of a player to win a matc, because the project's objective is estimating match outcome probabilities and challenging bookmakers'. I wrote Report 1 with wanting to use proba_elo as the target variable. But after the preprocessing and analysis I decided to go for a simpler metric, since I decided to treat this as a classification problem. I chose Winner as the main metrics, so that the problem is simplified. We want to predict whether a player will win or lose a match. This is demonstrated in Report 2.

- **What features of your dataset can you highlight?**

Elo Ratings (elo_winner and elo_loser): Elo ratings represent the skill level of players. They seem like important features for estimating match outcomes, as the difference in Elo ratings can provide insights into the relative strengths of the opponents.

Rankings (WRank and LRank): Player rankings are important indicators of performance. These features can help in understanding the historical performance of the participants.

Estimated Win Probability (proba_elo): This feature directly aligns with our project's objective. It represents the estimated probability of a player winning a match based on Elo ratings.

Match Surface (Surface): The type of surface (e.g., clay, grass, hard, carpet) can impact match outcomes. Different players may do great on different surfaces, but play poorly on other ones, so I think this might be an important feature in modeling.

Match Format (Best of): The format of the match (e.g., best of 3 or best of 5) affects the dynamics of the match and can influence predictions.

Number of sets won (Wsets and Lsets): The number of sets won/lost by the winner/loser. We can use these numbers to calculate the percentage of sets won by the winner and loser in previous matches.

Betting Odds (PSW, PSL, B365W, B365L): Betting odds provided by bookmakers can be informative features.

- **Are you limited by some of your data?**

I don’t think so.

Pre-processing and feature engineering

- **Did you have to clean and process the data? If yes, describe your treatment process.**

Yes, data cleaning and preprocessing were integral parts of this project to ensure that the dataset was suitable for machine learning. I took the following steps to treat the data:

- *Column Removal: I identified and removed irrelevant columns, such as 'ATP,' 'Comment,' 'Tournament,' and 'Location'.*
- *Handling Missing Data: I treated missing values in certain by filling them with the median values. This ensured that the data was complete.*
- *Feature Engineering: I created new features, including 'Winner_set_percentage' and 'Loser_set_percentage' (the percentage of sets won by the winner and loser in previous matches). Additionally, the percentage of victories for each player ('Winner_Win_Percentage' and 'Loser_Win_Percentage') in the past.*
- *Sorting by Date: The dataset was sorted by date to ensure that matches were in chronological order.*

In summary, I conducted data cleaning and preprocessing steps to eliminate irrelevant information, handle missing data, engineer new features, and organize the dataset in a chronological order. Before the modeling part, I applied additional preprocessing steps, such as encoding categorical variables and scaling features.

Here is my code for the ***initial*** *preprocessing:*

```
# PREPROCESSING/CLEANING OF DATA
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder

# 1. let's first look at the columns of the dataset and see if some are not relevant and can be dropped
# at first I thought ATP is the ATP ranking of the player but it's actually the Tournament number (because each row is a match),
# which means that ATP can likely be dropped because it seems unimportant
if 'ATP' in df.columns:
    df.drop(columns=['ATP'], inplace=True)
else:
    print("The 'ATP' column does not exist.")

# verify that it's dropped
df.info()

 # 2. Let's check the 'Comment', 'Tournament' and 'Location' columns next, let's see what values they have
columns_to_process = ['Comment', 'Tournament', 'Location']

for column_name in columns_to_process:
    unique_values = df[column_name].unique()
    unique_count = len(unique_values)

    print("Unique values in column '{}' ({} unique values):".format(column_name, unique_count))

    for value in unique_values:
        print(value)

# They have these values:
# Comment: Completed, Retired, Walkover, Disqualified
# Tournament: 207 unique values - tournament names
# Location : 115 unique values -city names
# They seem irrelevant for our models so they will be dropped too
df.drop(columns=columns_to_process, inplace=True)

# verify that they are dropped
df.info()

# 3. let's check how much data is missing per column (in percentage) to see if we can drop some more columns
missing_percentage = (df.isnull().sum() / len(df)) * 100
missing_data = pd.DataFrame({'Column': df.columns, 'Percentage of missing values': missing_percentage})
missing_data = missing_data.sort_values(by='Percentage of missing values', ascending=False)
print(missing_data)

# conclusion: some data is missing in the betting sites (PSW, PSL, B365W, B365L) columns but not enough and we want to keep those columns

# 4. The columns Wsets and Lsets represent the number of sets won/lost by the winner/loser
# Let's use these numbers to calculate the percentage of sets won by the winner and loser in previous matches

# Sort the df by date to ensure matches are in chronological order
df['Date'] = pd.to_datetime(df['Date'])
df.sort_values(by='Date', inplace=True)

# calculate the average percentage of sets won for the winner and loser in the past 50 matches
window_size = 50
df['Winner_set_percentage'] = df.groupby('Winner')['Wsets'].rolling(window=window_size).mean().reset_index(level=0, drop=True)
df['Loser_set_percentage'] = df.groupby('Loser')['Lsets'].rolling(window=window_size).mean().reset_index(level=0, drop=True)

# fill NaN values with 0
df['Winner_set_percentage'].fillna(0, inplace=True)
df['Loser_set_percentage'].fillna(0, inplace=True)

# 5. Let's now use the 'Winner' and 'Loser' columns to calculate the percentage of victories of the winner and loser in the past
# sort the df to ensure matches are in chronological order
df['Date'] = pd.to_datetime(df['Date'])
df.sort_values(by='Date', inplace=True)

# dictionaries to store the cumulative wins for each player
winner_cumulative_wins = {}
loser_cumulative_wins = {}

# lists to store the calculated percentages
winner_percentages = []
loser_percentages = []

# iterate through the df to calculate percentages
for index, row in df.iterrows():
    winner = row['Winner']
    loser = row['Loser']

    # initialize wins for players if not already present
    if winner not in winner_cumulative_wins:
        winner_cumulative_wins[winner] = 0
    if loser not in loser_cumulative_wins:
        loser_cumulative_wins[loser] = 0

    # update cumulative wins
    winner_cumulative_wins[winner] += 1
    loser_cumulative_wins[loser] += 1

    # calculate percentages
     winner_percentage = winner_cumulative_wins[winner] / (winner_cumulative_wins[winner] + loser_cumulative_wins.get(winner, 0))
    loser_percentage = loser_cumulative_wins[loser] / (winner_cumulative_wins.get(loser, 0) + loser_cumulative_wins[loser])

    # append percentages to lists
    winner_percentages.append(winner_percentage)
    loser_percentages.append(loser_percentage)

# create new columns in the DataFrame for the calculated percentages
df['Winner_Win_Percentage'] = winner_percentages
df['Loser_Win_Percentage'] = loser_percentages

# 6. Handle missing values
# let's replace the missing values for the numerical columns with the median values
columns = ['PSW', 'PSL', 'B365W', 'B365L', 'Lsets', 'Wsets', 'LRank', 'WRank']
column_means = df[columns].median()
df[columns] = df[columns].fillna(column_means)

df.head(20)

```

- **Did you have to carry out normalization/standardization type transformations of your data? If yes, why?**

Yes, I did in the next preprocessing step to prepare the data for modeling because it's needed for the machine learning algorithms to ensure that all numerical features are on the same scale.

- **Are you considering dimension reduction techniques in the modeling part? If yes, why?**

No.

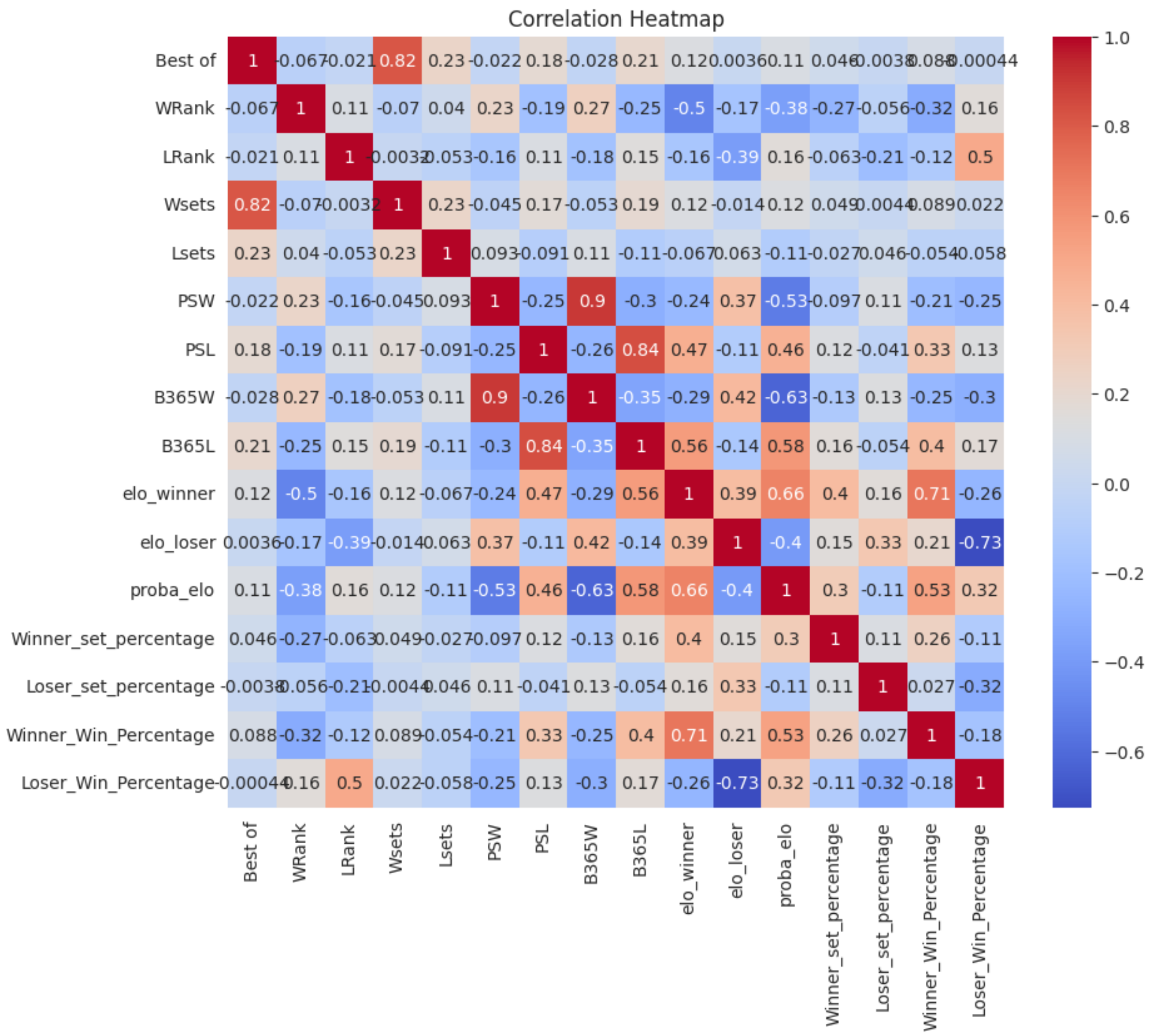
Visualizations and Statistics

- **Have you identified relationships between different variables? Between explanatory variables? and between your explanatory variables and the target(s)?**

Yes, relationships between different variables have been identified in the dataset. Here are some visualizations from my analysis:

Between Explanatory Variables (Features):

```
# Correlation Heatmap
# Visualize the correlations between different variables in your dataset,
# including Elo ratings and other features, using a heatmap.
# This can help you identify which variables are most strongly related to match outcomes.

correlation_matrix = df.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title("Correlation Heatmap")
plt.show()

# Between "Best of" and "Wsets" there's a relatively strong positive correlation
# Higher Elo ratings for the winning player are associated with a higher percentage of victories in their past matches.
```

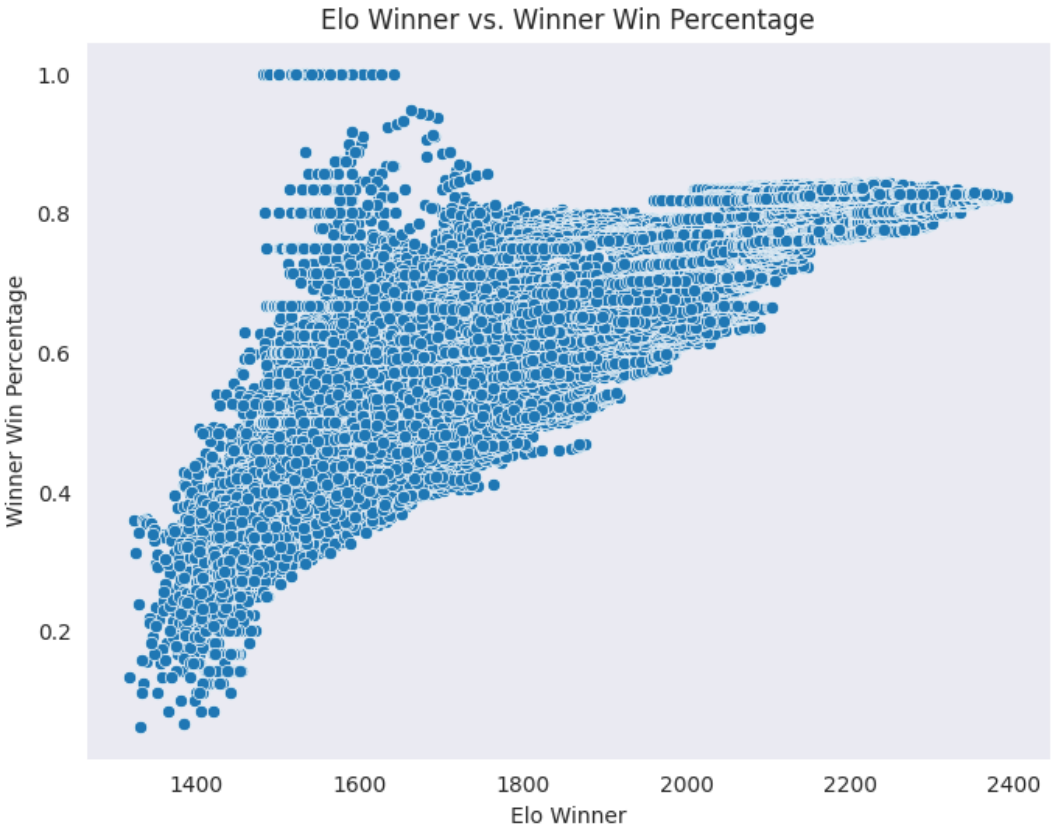


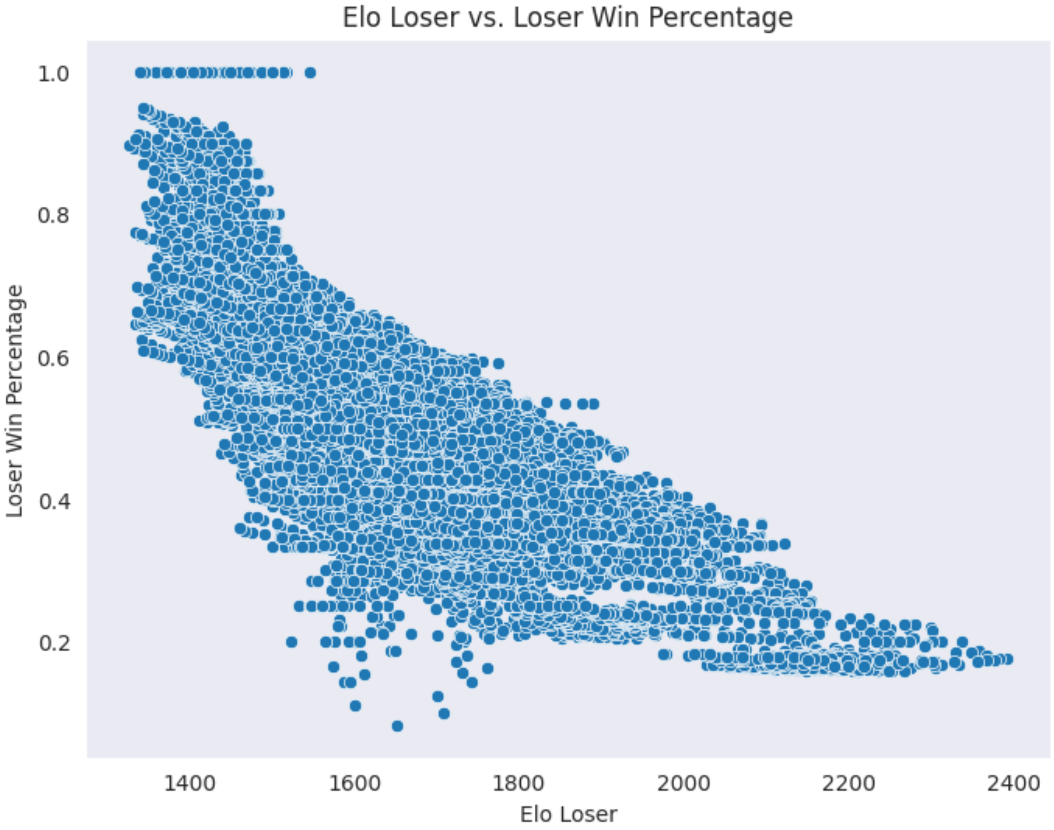
Some conclusions from the matrix:
- Best of and Wsets have a correlation of 0.82, which means that when more sets are played in a match, the winner tends to win more sets. Increasing the number of sets played in a match is associated with the winner winning more sets.

- Elo_winner and Winner_Win_Percentage have a high-ish correlation (0.71). The same goes for elo_loser and Loser_Win_Percentage (-0.73).

```
plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='elo_loser', y='Loser_Win_Percentage')
plt.title('Elo Loser vs. Loser Win Percentage')
plt.xlabel('Elo Loser')
plt.ylabel('Loser Win Percentage')
plt.show()
```

```
plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='elo_winner', y='Winner_Win_Percentage')
plt.title('Elo Winner vs. Winner Win Percentage')
plt.xlabel('Elo Winner')
plt.ylabel('Winner Win Percentage')
plt.show()
```


Elo Winner vs. Winner Win Percentage
1.0
0.8
0.6
0.4
0.2
Winner Win Percentage
1400
1600
1800
2000
2200
2400
Elo Winner


Elo Loser vs. Loser Win Percentage
1.0
0.8
0.6
0.4
0.2
Loser Win Percentage
1400
1600
1800
2000
2200
2400
Elo Loser

elo_loser and Loser_Win_Percentage (-0.73):

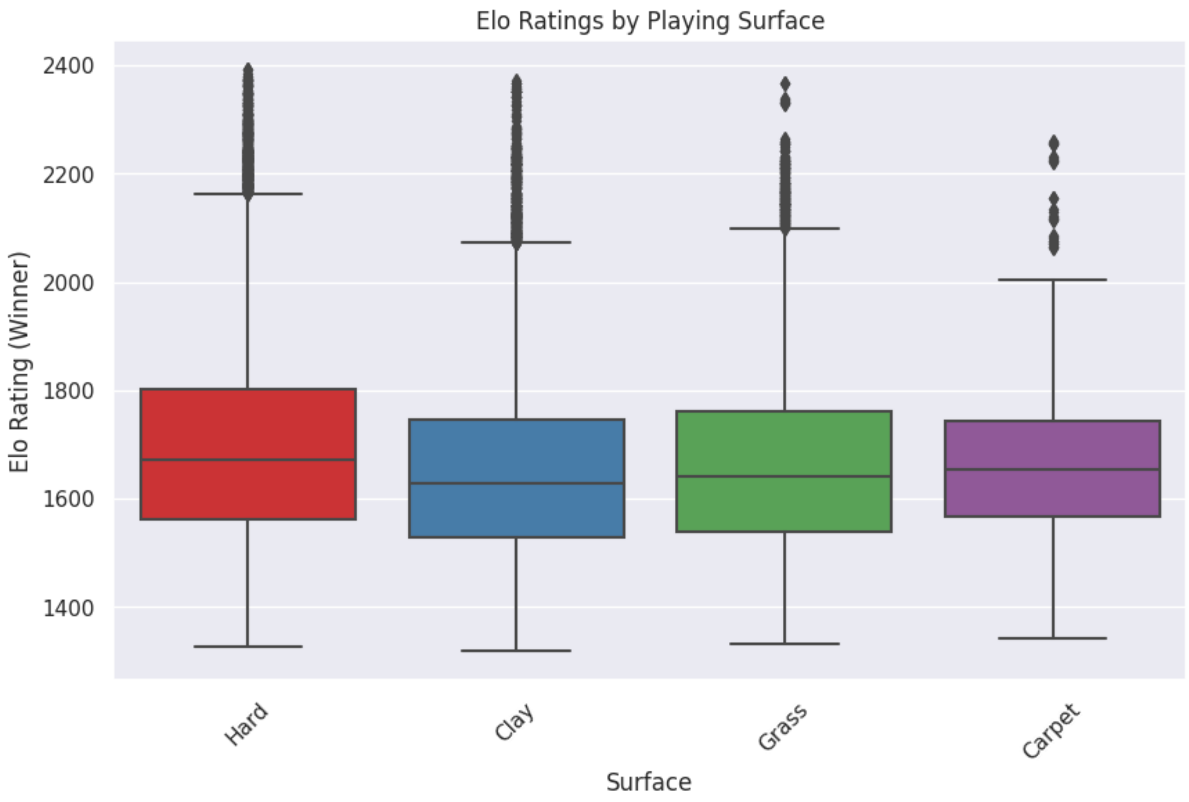
A negative correlation suggests that, as a player's Elo rating (performance) increases, their likelihood of winning (Loser Win Percentage) decreases. This can mean that players with higher Elo ratings may not perform as well in terms of winning matches, which might seem counterintuitive but could be due to various factors such as facing tougher opponents. It means that players (losers in each match) with higher Elo ratings tend to have a lower percentage of match wins.

elo_winner and Winner_Win_Percentage (0.71)

The positive correlation means that players with higher Elo ratings are more likely to win their matches. This trend is intuitive matches my understand of the Elo ratings. Players with higher Elo ratings are typically considered stronger or more skilled, and, therefore, they are expected to have a higher probability of winning matches.

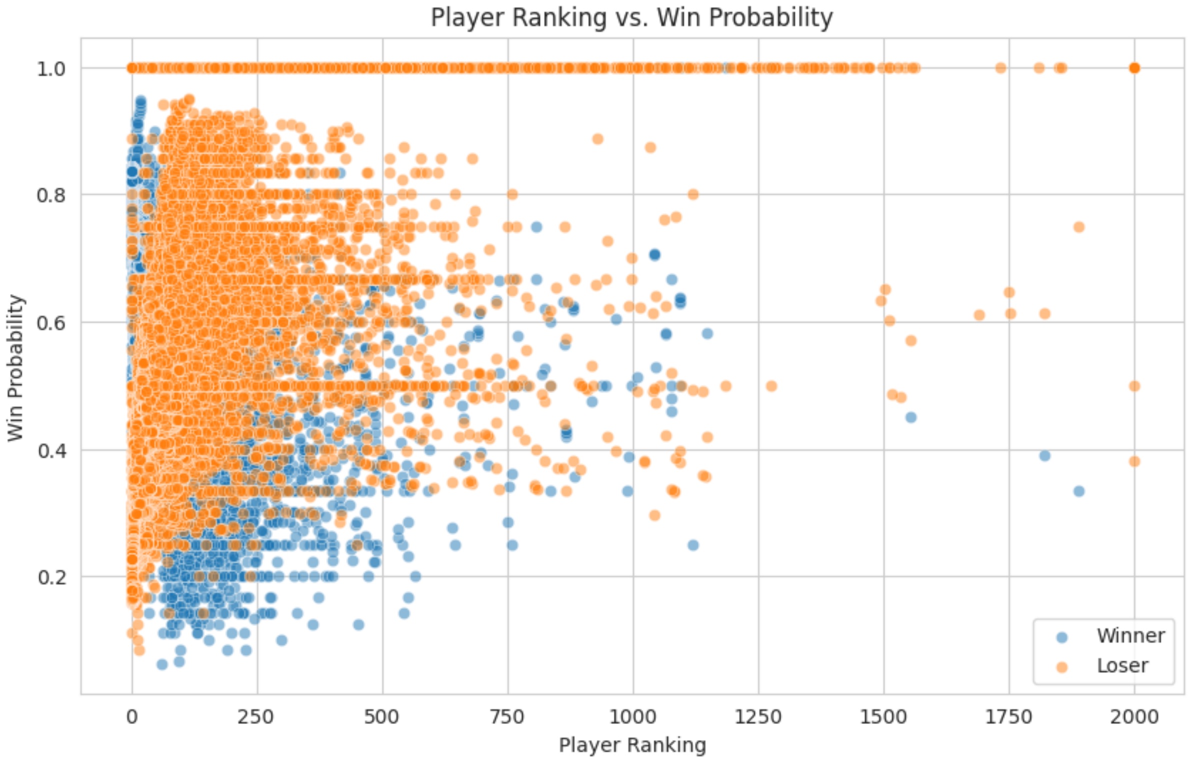
Let's look at some other variables:

```
# create a box plot to compare Elo ratings across different playing surfaces
plt.figure(figsize=(10, 6))
sns.boxplot(data=df, x='Surface', y='elo_winner', palette='Set1')
plt.title('Elo Ratings by Playing Surface')
plt.xlabel('Surface')
plt.ylabel('Elo Rating (Winner)')
plt.xticks(rotation=45)
plt.show()
```



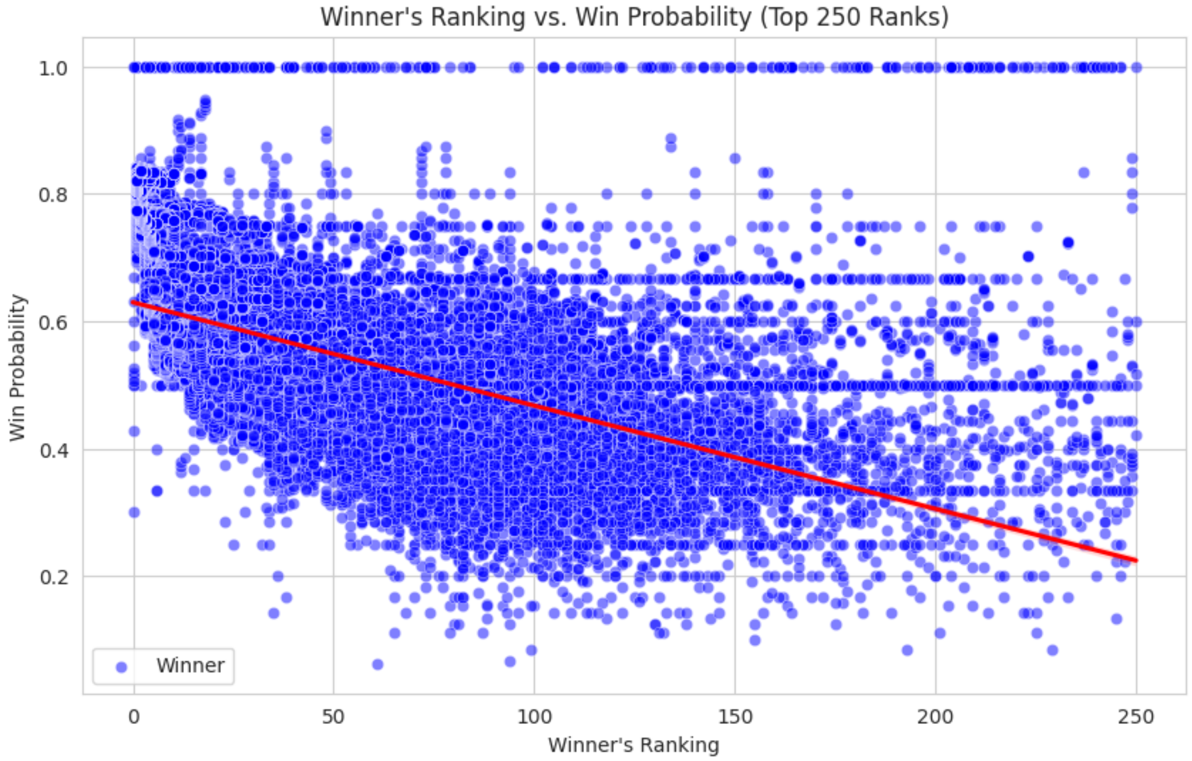
The graph indicates that hard surfaces have the highest median Elo ratings (it is slightly higher than the winner Elo ratings on Grass, Carpet and Clay). There is also a greater *range in Elo ratings for matches played on hard surfaces, compared to other surfaces. All surfaces show outliers for exceptionally high Elo ratings, especially Clay and Grass.*

```python
# Player Ranking vs. Win Probability
# This scatter plot will show the relationship between player ranking and win probability

# Create a scatter plot
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='WRank', y='Winner_Win_Percentage', label='Winner', alpha=0.5)
sns.scatterplot(data=df, x='LRank', y='Loser_Win_Percentage', label='Loser', alpha=0.5)

plt.xlabel("Player Ranking")
plt.ylabel("Win Probability")
plt.title("Player Ranking vs. Win Probability")

plt.legend()
plt.show()

```

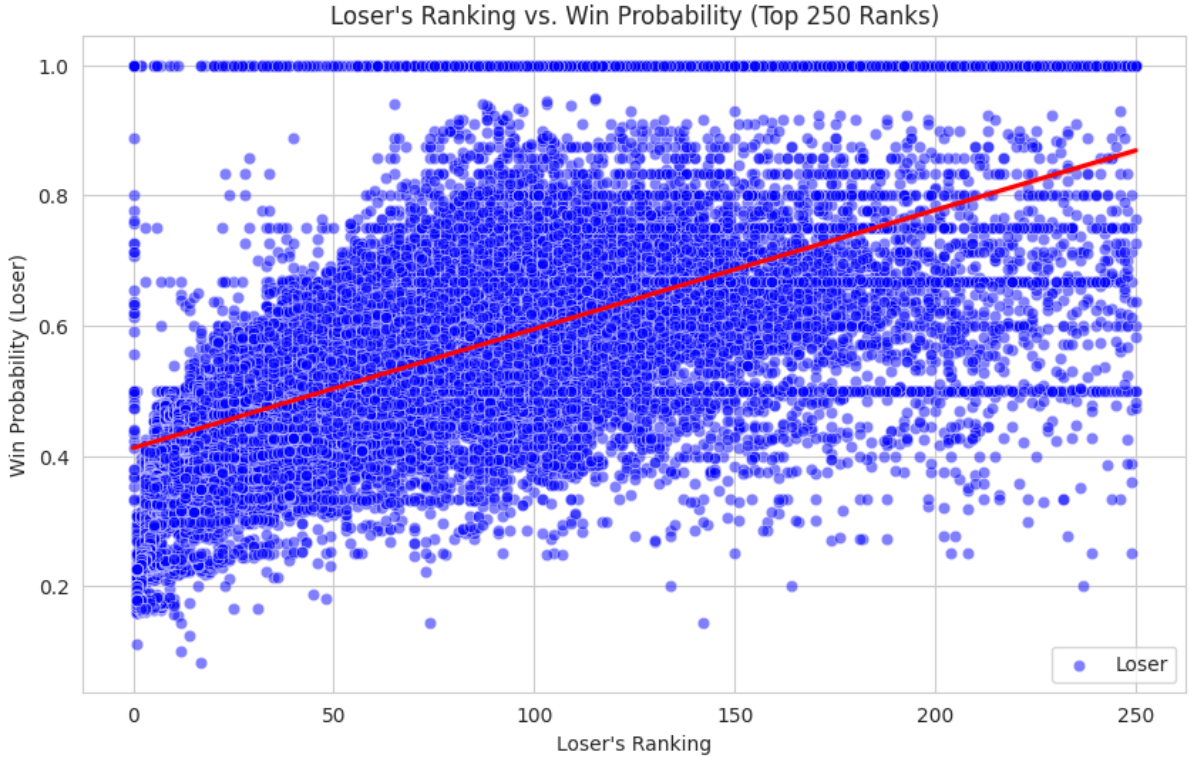


- *Players without Rankings: Players without rankings might be newcomers or lower-profile players who haven't received an official ranking yet.*
- *Gaps in Rankings: There are gaps in the scatter plot between high rankings (for example between 1150 and 1550) so it seems that there's incomplete data in the dataset. Some matches or players within those ranking ranges may not have been included.*

As most of the players are in the lower rankings (e.g., 0-250), let's draw plots for those ranges to see if there is a trend:



- *As the rank decreases (gets better), the plot shows that the probability also increases.*
- *There's matches with win probability of 1 for almost all rankings. This could be explained with the dataset containing matches played under specific conditions or rules where players within this ranking range have a very high likelihood of winning. For example, if these matches are from lower-tier tournaments or involve new players, it could explain the high win probability.*

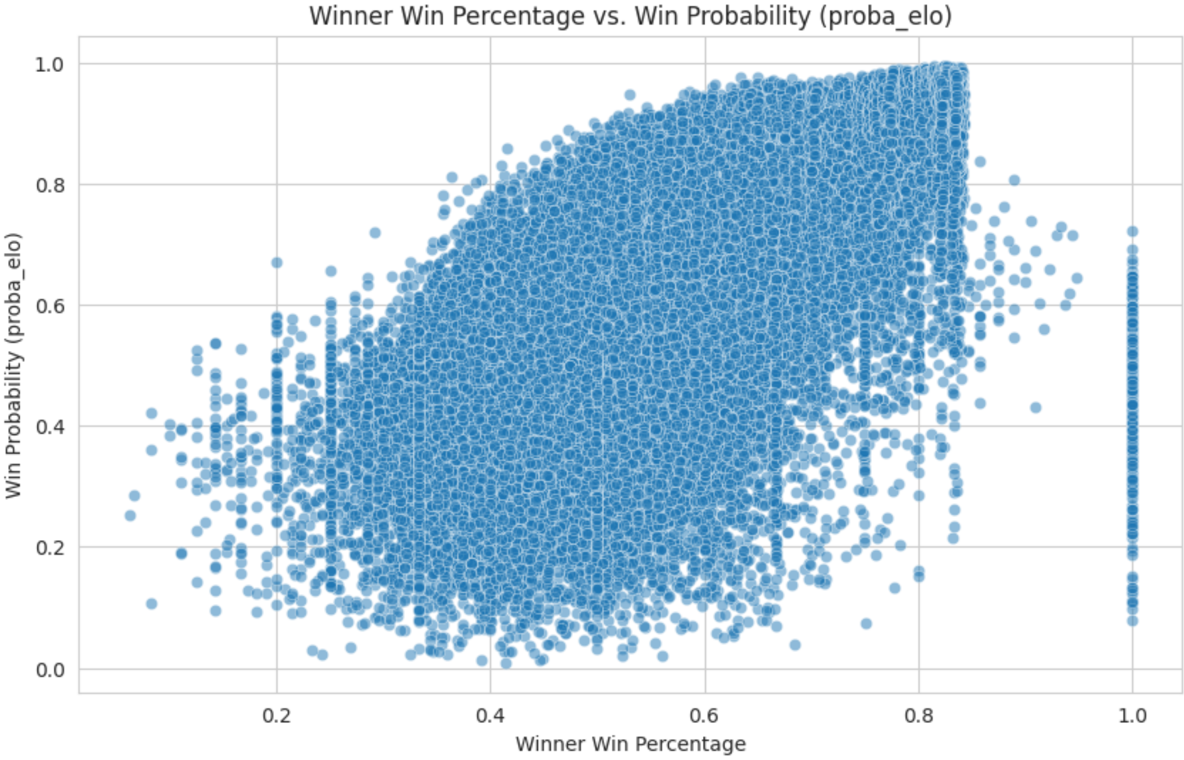


- *I also checked for Round, Series, Court, but they don't seem to be related to the win probability.*

Between Explanatory Variables and the Target Variable (proba_elo):

The "proba_elo" variable represents the estimated win probability of a player. I assume it has relationships with various features, such as player rankings, Elo ratings, and previous win percentages.

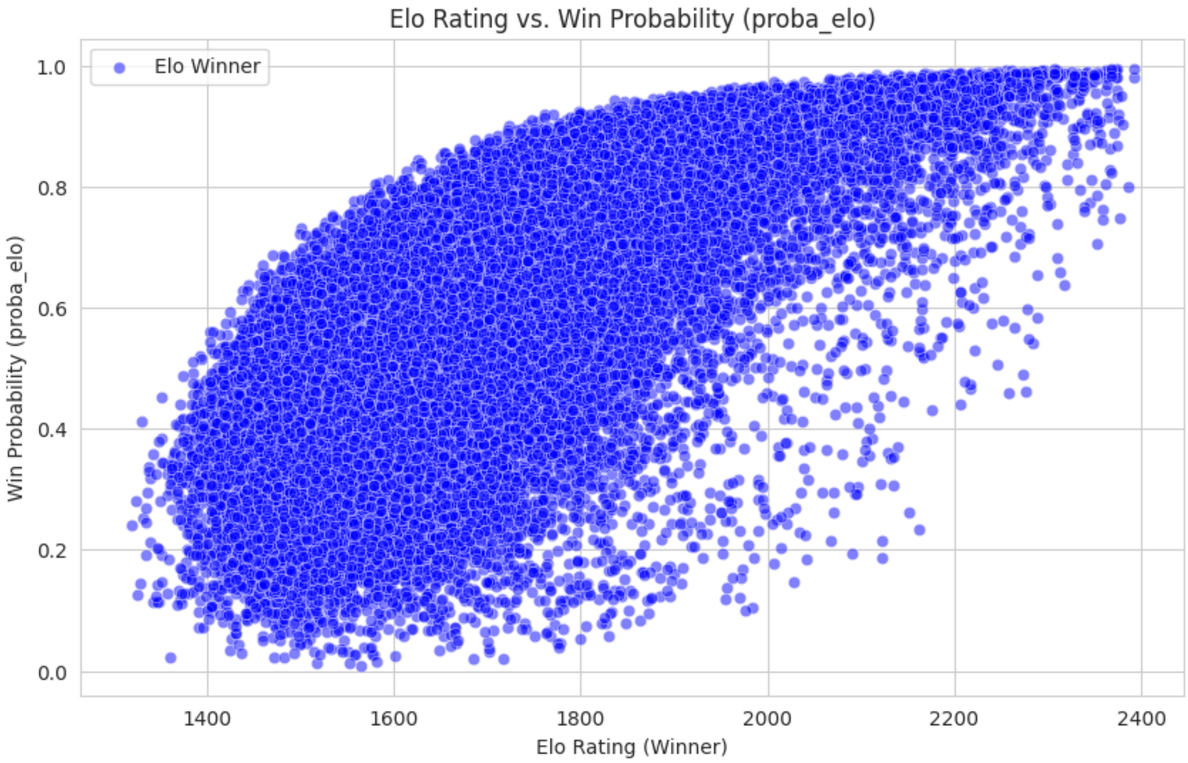
```
# Winner_Win_Percentage vs proba_elo
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='Winner_Win_Percentage', y='proba_elo', alpha=0.5)
plt.title('Winner Win Percentage vs. Win Probability (proba_elo)')
plt.xlabel('Winner Win Percentage')
plt.ylabel('Win Probability (proba_elo)')
plt.show()

```



- positive correlation: as Winner_Win_Percentage increases, proba_elo tends to increase
- the points are more dispersed, so the relationship is not as strong.
- curvy (non-linear) pattern
- correlation coefficient: 0.53
- cluster points on the right could be players who have played only one game and won it (they have Winner_Win_Percentage of 1, which is expected since they have a 100% win rate in their limited history).

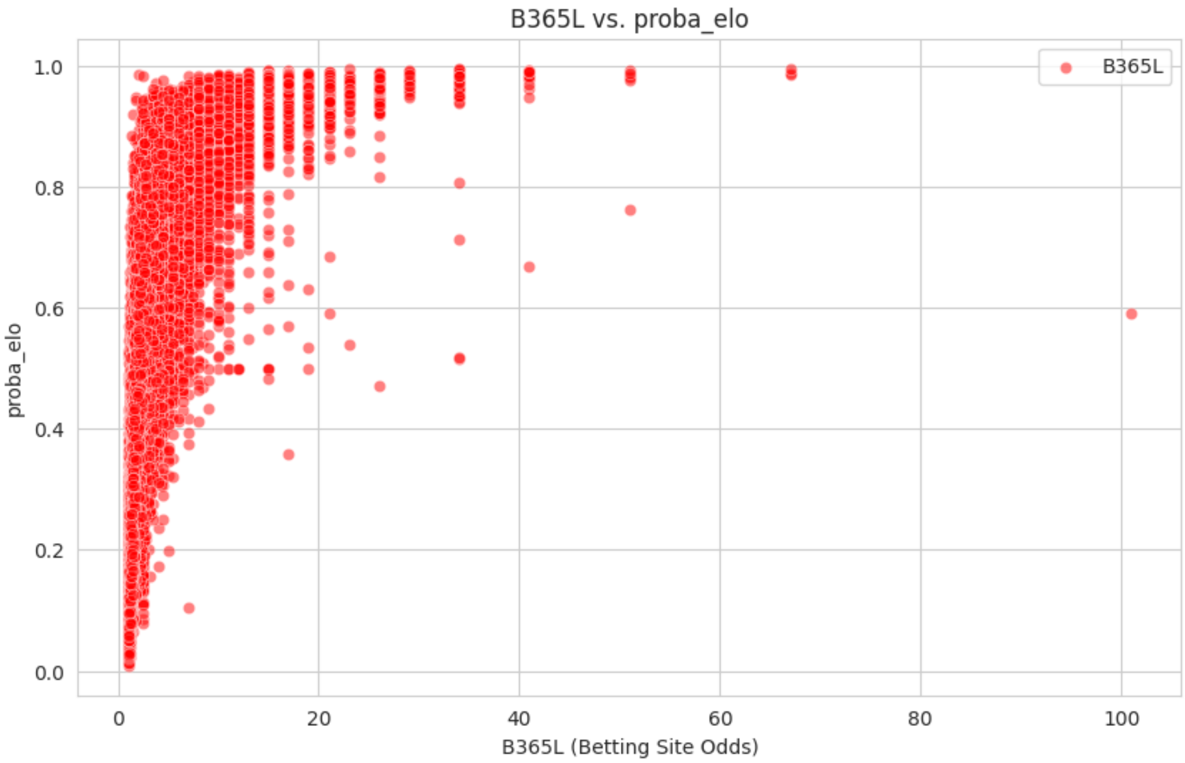
```
# Elo Winner vs proba_elo
```

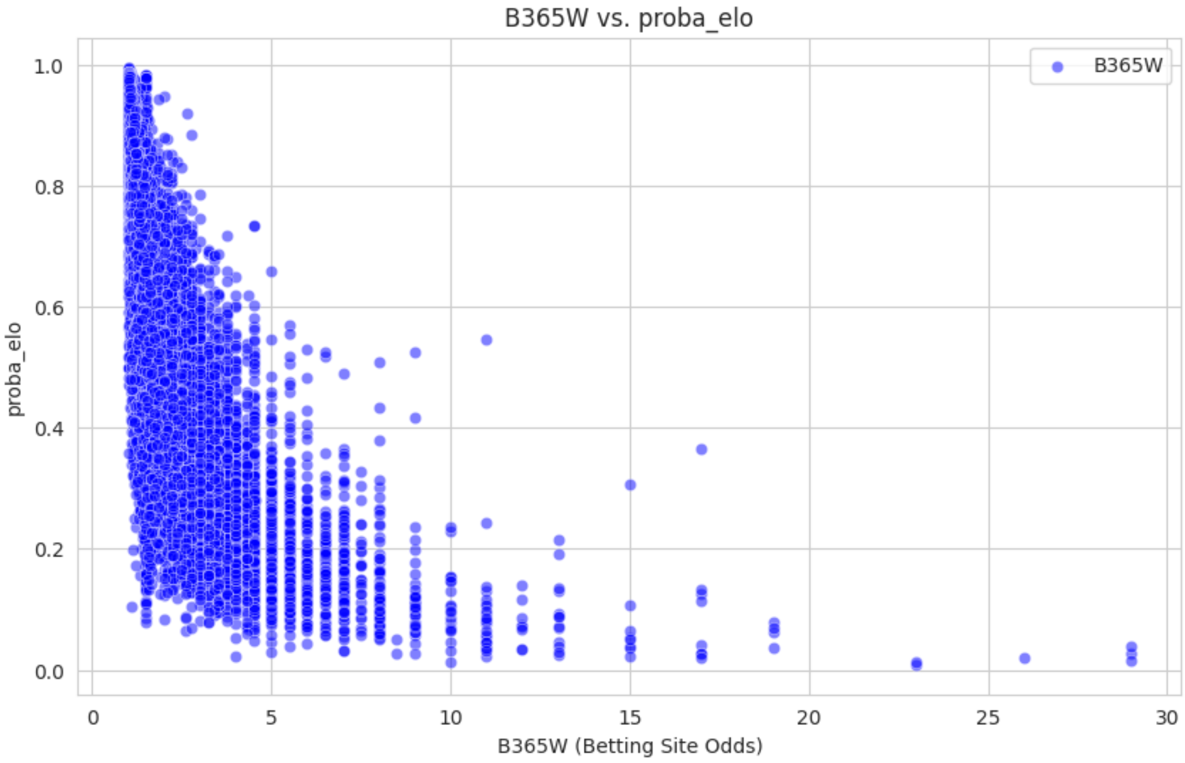
```
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='elo_winner', y='proba_elo', alpha=0.5)
plt.title('Elo Winner vs. Win Probability (proba_elo)')
plt.xlabel('Elo Winner')
plt.ylabel('Win Probability (proba_elo)')
plt.show()

```



- *positive correlation*
- *stronger relationship than proba_elo has with winner_win_percentage*
- *curvy (non-linear) pattern*
- *correlation coefficient: 0.66*
- *higher elo ratings suggest a higher win probability*

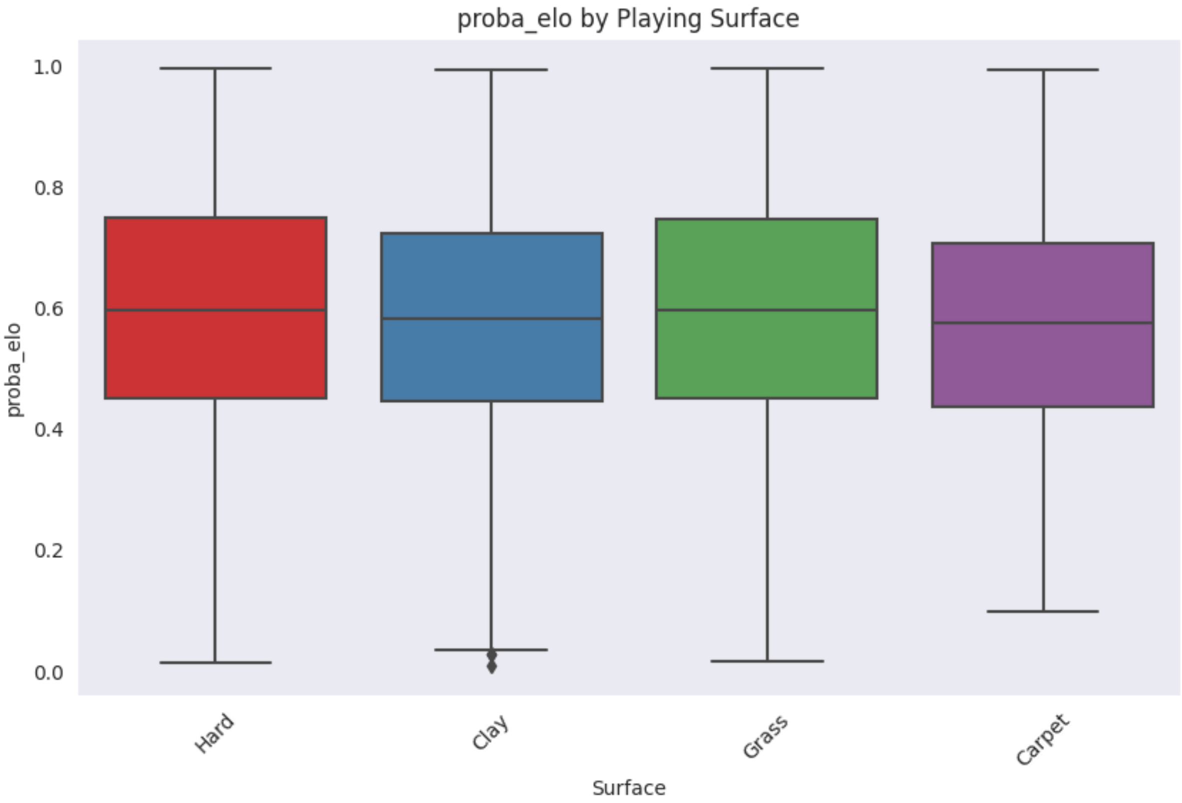
```
# Create scatter plots for B365W vs. proba_elo
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='B365W', y='proba_elo', alpha=0.5, label='B365W', color='blue')

plt.xlabel('B365W (Betting Site Odds)')
plt.ylabel('proba_elo')
plt.title('B365W vs. proba_elo')
plt.legend()
plt.show()

# Create scatter plots for B365L vs. proba_elo
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='B365L', y='proba_elo', alpha=0.5, label='B365L', color='red')

plt.xlabel('B365L (Betting Site Odds)')
plt.ylabel('proba_elo')
plt.title('B365L vs. proba_elo')
plt.legend()
plt.show()

```





- the graphs indicate a strong correlation between betting site odds and proba_elo
- stronger relationship than proba_elo has with other features
- curvy (non-linear) pattern

- correlation coefficient: B365W vs proba_elo: -0.63 and B365L vs proba_elo: 0.58
- some outliers (not relevant)

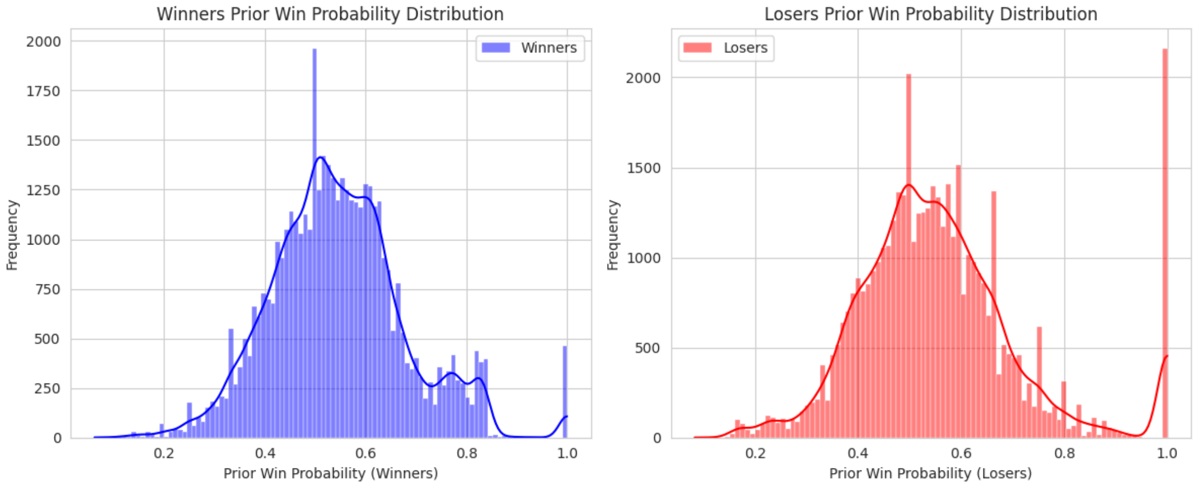


The box plot shows the relationship between proba_elo and the playing surface. It looks very similar for all surfaces and the proba_elo ratings in the data set.

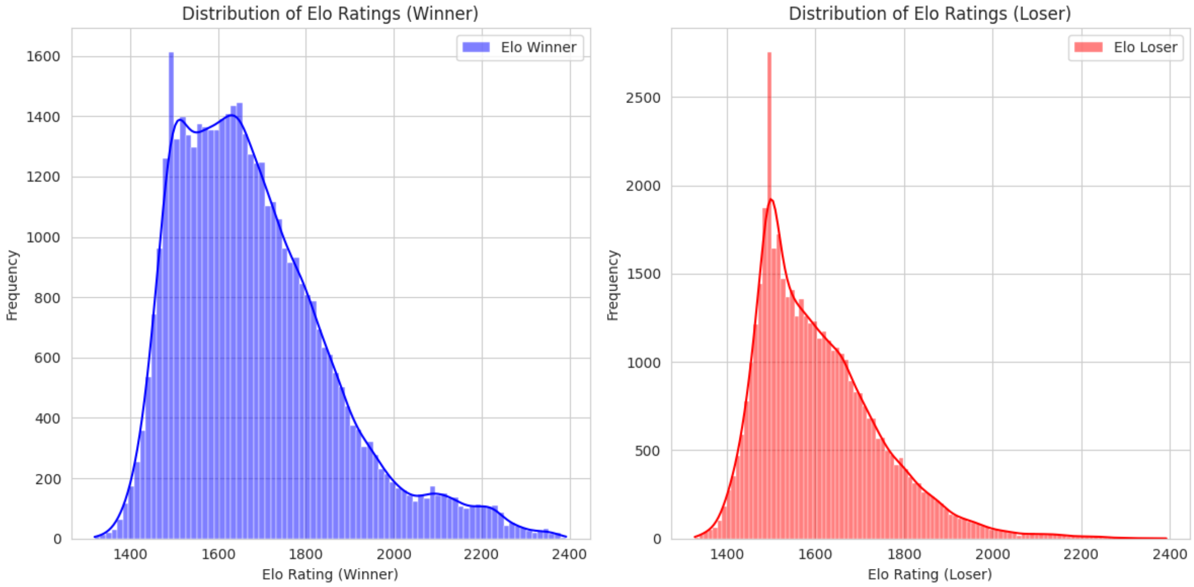
- **Describe the distribution of these data, distribution, outliers.. (pre/post processing if necessary)**

```
# 1. Winners and Losers Win Probability Distribution

import matplotlib.pyplot as plt
import seaborn as sns

# Set the style
sns.set_style("whitegrid")

# Create a figure and axis
fig, ax = plt.subplots()

# Plot the histograms for winners and losers
sns.histplot(data=df, x='Winner_Win_Percentage', kde=True, color='blue', label='Winners')
sns.histplot(data=df, x='Loser_Win_Percentage', kde=True, color='red', label='Losers')

# Set labels and title
ax.set_xlabel("Prior Win Probability")
ax.set_ylabel("Frequency")
ax.set_title("Winners and Losers Prior Win Probability Distribution")

# Add a legend
plt.legend()

```

```
# Show the plot
plt.show()

```

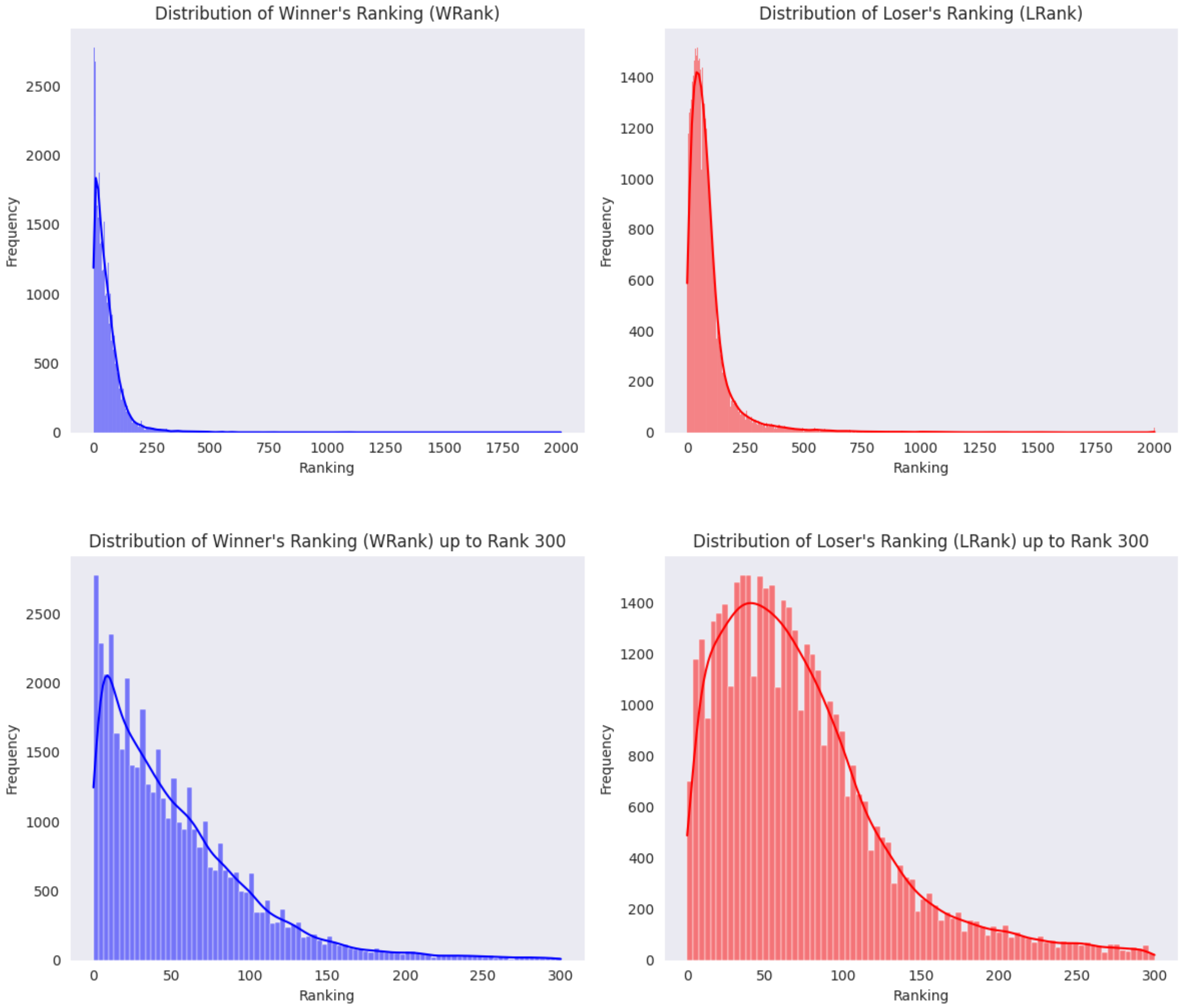


These histograms show the distribution of win probabilities for both winners and losers. The distributions are similar, clustering around 0.5, with a significant spread, with some having very low probabilities (close to 0) and a few with high probabilities (close to 1). It follows a relatively normal distribution with a central value but with some variation. The probabilities of 1 can be interpreted as probabilities for newcomer players with only 1 match in their history (which they won).

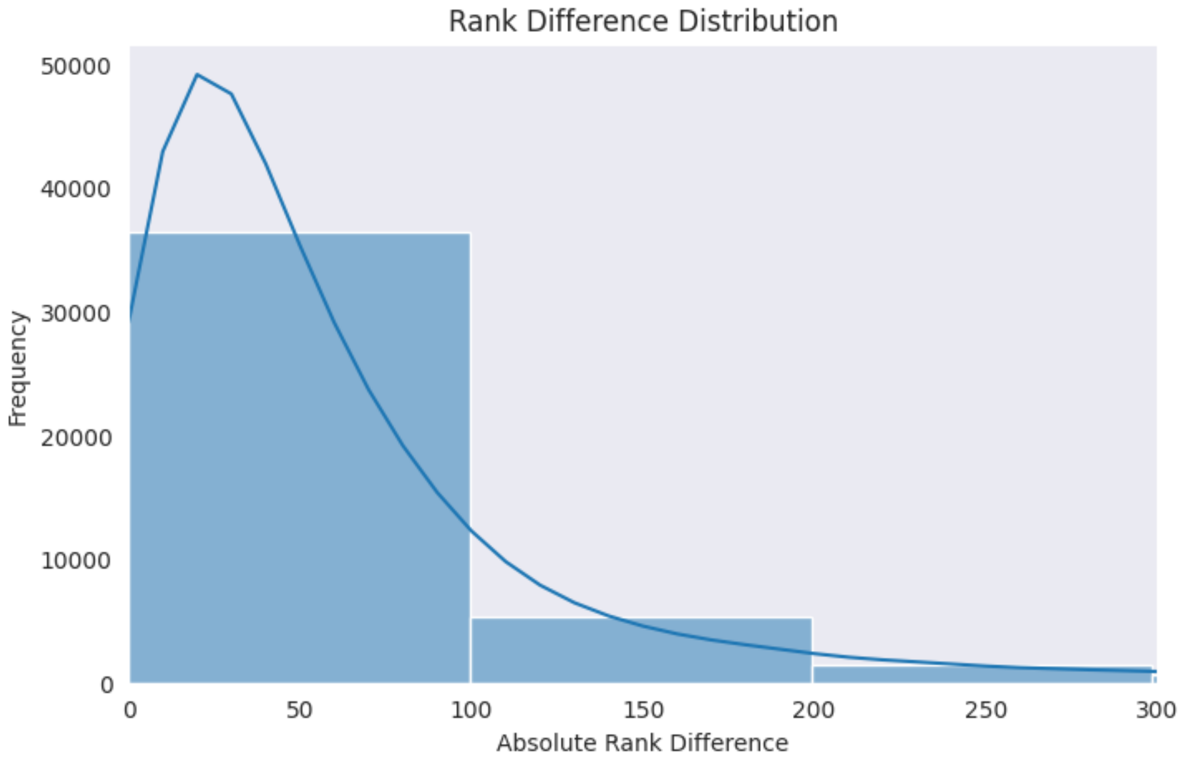


Loser Distribution: More players have Elo ratings below the center point, which implies that they might be considered less skilled or have lower ratings, while fewer players have higher ratings.

Winner Distribution: The distribution is more normal/even, winners tend to have a higher Elo rating.



The ranking (WRank, LRank) distribution shows that Winner's ranking are more tightly concentrated in the lower (better) ranking, while the Loser's ranking have a more normal distribution in comparison, having more rankings that are greater (worse).



Most matches have relatively low absolute rank differences, meaning that they involve players with rankings close to each other. Matches with larger absolute rank differences are less common, suggesting that matches between players with significantly different rankings are rarer in our dataset.

- **Present the statistical analyzes used to confirm the information present on the graphs.**

Here's a summary (most of it was already presented above):

- *Best of and Wsets Correlation (0.82): The strong positive correlation means that as more sets are played in a match (Best of), the winner tends to win more sets. An increase in the number of sets played in a match is associated with the winner winning a higher number of sets.*
- *Elo Winner and Winner Win Percentage Correlation (0.71): This positive correlation suggests that players with higher Elo ratings are more likely to win their matches. It aligns with the concept of Elo ratings where players with higher ratings are expected to play better.*
- *Elo Loser and Loser Win Percentage Correlation (-0.73): The negative correlation indicates that as a player's Elo rating (performance) increases, their likelihood of winning (Loser Win Percentage) decreases. Players with higher Elo ratings tend to have a lower percentage of match wins, which could be due to facing tougher opponents.*
- *Players without Rankings: Players without rankings are probably newcomers or lower-profile players who haven't received an official ranking yet.*

- *Gaps in Rankings: The gaps in the scatter plot, especially between higher rankings (e.g., 1150 and 1550), suggest potential incomplete data in the dataset. Matches or players within those ranking ranges may not have been included.*
- *Ranking Trends for Lower Rankings (0-250): There's a clear trend: as the rank decreases (gets better), the probability of winning increases. Matches with a win probability of 1 for almost all rankings could be explained by specific conditions or rules in lower-tier tournaments, where players within this ranking range have a very high likelihood of winning.*
- *Round, Series, Court Relationship: Round, series, and court type don't seem to be strongly related to win probability.*
- *Winner Win Percentage vs. Proba_elo: There's a positive correlation (0.53) between Winner Win Percentage and Proba_elo, showing that as Winner Win Percentage increases, Proba_elo tends to increase. The curvy pattern suggests a non-linear relationship, and there are cluster points on the right, maybe representing players who have played only one game and won it.*
- *Elo Winner vs. Proba_elo: Elo Winner has a stronger positive correlation (0.66) with Proba_elo than Winner Win Percentage. The curvy pattern means a non-linear relationship where higher Elo ratings suggest a higher win probability.*
- *Betting Site Odds vs. Proba_elo: The graphs indicate a strong correlation between betting site odds and Proba_elo, with a curvy pattern. B365W has a correlation of -0.63, while B365L has a correlation of 0.58. Outliers exist but are not particularly relevant.*
- *Winner and Loser Win Probability Distributions: The distributions of win probabilities for both winners and losers are similar, clustering around 0.5, but with lots of variability. The presence of probabilities of 1 could be explained that they might be newcomers or specific match conditions where players have a very high likelihood of winning.*
- *Elo Ratings Distribution: Elo ratings are more centered for winners, suggesting that they tend to have more consistent ratings. In contrast, losers' Elo ratings have a wider distribution, meaning that they include both lower and higher ratings.*
- *Ranking (WRank and LRank) Distribution: The distribution of winner's rankings is concentrated in the lower (better) rankings, while the distribution of loser's rankings is more evenly distributed, which means more varied rankings.*
- *Rank Difference Distribution: Most matches involve players with relatively low absolute rank differences, meaning that they have rankings close to each other. Matches with larger rank differences are less common, implying that matches between players with significantly different rankings are rarer in the dataset.*

● Draw conclusions from the elements noted above allowing them to project themselves into the modeling part

- *Feature Importance: The correlation analysis revealed that certain features are more strongly related to the target variable (proba_elo) than others. The Elo ratings of both winner and loser had a significant impact on win probability. These ratings should be considered as important features in the model.*

- *Rankings: WRank and LRank showed a strong relationship with win probability. Matches involving players with significantly different rankings are less common.*
- *Number of Sets: Best of and Wsets were highly correlated, suggesting that the number of sets played in a match impacts win probability.*
- *Betting Odds: Betting site odds, particularly B365W and B365L, showed a strong correlation with proba_elo. This indicates that odds provided by bookmakers hold predictive power for match outcomes. The model can use these odds as features.*
- *Win Percentages: The relationship between Winner_Win_Percentage and proba_elo suggests that players with a higher historical win rate tend to have higher win probabilities in future matches. The model should incorporate this historical win percentage as a feature.*
- *Surface and Round: Not as much related to win probability as Elo ratings and rankings, I still want to consider the type of playing surface and the round of the tournament when building the model. These factors can affect the player performance.*
- *Handling Outliers: Outliers, such as players with very high win probabilities, may be linked to specific conditions or new players.*