School of Electronic Engineering and Computer Science

Interim Report

Programme of study:
Computer Science and Mathematics

Project Title: Predicting UFC Fight Outcomes using Machine Learning

Supervisor:
Keshav Bhandari

Student Name:
Luke Bransby

Final Year
Undergraduate Project 2023/24

Queen Mary
University of London

Date: 6 May 2025

Abstract

This project develops a machine learning system to predict UFC fight outcomes by incorporating fighter performance data, fight styles, and rankings.

The first step focuses on feature engineering, where relevant fight performance metrics are extracted and pre-processed.

An Elo ranking system is implemented to rank fighters based on their past performances, providing an additional layer of context for the predictions.

The project then incorporates fight style descriptions, using Large Language Model APIs to generate text-based embeddings that capture the nuances of each fighter's performance and strategy.

Finally, an ensemble approach is used to combine multiple machine learning models, such as Random Forest, XGBoost, LightGBM, and CatBoost via stacking and voting techniques. This approach ensures that the strengths of each model are leveraged while minimizing their weaknesses.

The system is designed to predict the fight winner, offering a comprehensive and reliable solution for UFC fight prediction.

Table of Contents

Chapter 1: **Introduction**

1.1 What is MMA?

Mixed Martial Arts (MMA) is a full-contact combat sport which encourages the use of many martial arts, such as wrestling, boxing and Brazilian jiu-jitsu and many more. The Ultimate Fighting Championship (UFC) is the largest organization that hosts MMA fights. High unpredictability of fights intrigues fans across the globe to tune in to watch the action unfold.

1.1.1 Rounds and Time Limits

The fights are typically three 5-minute rounds, with champion bouts and main events being five 5-minute rounds.

1.1.2 Weight Classes

Fighters must compete within a specific weight class, with weigh-ins taking place the day before the fight. Both fighters must make the designated weight for the fight.

1.1.3 Victory Conditions

- Knockout (KO): The opponent is rendered unconscious from a strike
- Technical Knockout (TKO): The referee stops the fight if one fighter is unable to defend themselves
- Submission/ Sub: A fighter taps out either physically or verbally due to a submission hold such as a joint lock or choke
- Decision: If the fight lasts until the end of the designated rounds, the judges score the fight based on criteria such as effective strikes, grappling, control and aggression. Victory is awarded via unanimous, split, or majority decision.
- Disqualification: A fighter is disqualified for purposeful fouls committed against the opponent.
- No Contest: The fight ends without a winner due to accidental fouls that had severe consequences or circumstances beyond control.

1.1.4 Illegal Techniques

- Strikes to the back of the head
- Strikes and grabs to the groin, throat, spine or small joints like fingers and toes
- Kicks and knees to the head of a grounded opponent (any part of body other than feet touching the floor)
- Downward elbow strikes
- Intentional throwing or spiking the opponent on their head or neck
- Grabbing the cage to prevent grappling exchanges

1.1.5 Referee and Judges

The referee ensures fighter safety and stops the fight when necessary to prevent injury. Judges use a 10-point system, awarding the winner 10 points and the loser 9 or fewer, based on performance. A unanimous decision occurs when all judges agree on the winner, while a split decision happens when the majority agrees. If there’s no clear consensus, the fight results in a draw.

1.1.6 Fouls and Penalties

Committing a foul can result in a warning, point deduction or disqualification depending on severity, intent and if the foul is repeated.

1.2 Background

As data analysis has grown, it has been implemented into sports analytics. MMA has many metrics that a fight can be analysed through machine learning. These metrics are used to create predictions for upcoming events. The high unpredictability of MMA fights captivates fans and analysts leading them to tune in to watch fights. Many people participate in betting on what will happen in fights, often using information to support their betting claims.

1.3 Inspiration

The idea for this project stemmed from a YouTube video discussing the shift in UFC champions from strikers to wrestlers over the past six months. The video analysed trends between takedowns and win percentage, showing that higher successful takedowns correlate with better win rates. Intrigued, I downloaded the dataset to explore further and discovered a wealth of data from each match, which I believed could yield valuable insights, as it covered most major UFC fights in the modern era.

1.4 Metrics

During this document, I will be referring to two types of metrics within this document: performance metrics and outcome metrics.

Performance metrics would cover aspects of a fight such as Takedown Attempts, Significant Strikes Landed, Control Time, Submission Success rate, etc. They detail the summation of individual actions that took place during the fight.

Outcome metrics would cover Victory Method, Submission Win, Rounds Completed, Victor etc. They describe the overall outcome of the fight, rather than the performance. Outcome metrics are often what is discussed most about a fight.

1.5 Problem Statement

The problem is to develop an accurate UFC fight outcome prediction model using machine learning, by leveraging fighter performance metrics and historical fight data. A subsequent problem is having the high-quality data to create the highest performing machine learning algorithm.

1.6 Aim

The aim of this projects is to develop a machine learning model to predict fight performance metrics and fight outcome metrics. The model will account for the predicted performance metrics it generates as a part of the overall prediction process.

This will be supplemented by analysing the relationship between individual performance metrics to identify trends. This analysis will help to train the model to better understand the impact of individual performance metrics when trying to predict outcome metrics.

1.7 Objectives

- Build a processed UFC dataset that reflects the official judging criteria by integrating round-by-round and cumulative performance metrics.
- Train and test multiple machine learning models to determine which is the most accurate for determining of fight performance metrics.
- Implement an Elo ranking system to create a better prediction model
- Create an Ensembling Model of the top performing models to create the best possible model

1.8 Research Questions

- How do round-by-round and cumulative performance metrics influence fight outcomes?
- What data is required to make the most accurate machine learning algorithm to predict UFC fights?
- Which ensemble methods provide the best balance between prediction accuracy and model generalizability?
- What are the practical limitations of integrating external data sources like rankings and betting odds?
- How to account for non-numeric influences on fight outcomes which are difficult to quantify but can significantly affect the result and limit the model's predictive power.

1.9 Challenges

- Accounting for subjectivity in judging decisions that aren't always reflected in fight metrics.
- Managing limited data for debuting fighters with no prior history.
- Preventing overfitting while working with a relatively small and imbalanced dataset.
- Preventing data leakage from non-numeric or contextual factors, leading to artificially inflated model performance if not carefully managed.

1.10 Project Plan

Phase 1: Research and Learning (September 30th – November 18th)

Sep 30	Oct 16	Project Foundation and initial research
Oct 16	Oct 29	Data Collection and Initial Exploratory Data Analysis
Oct 30	Nov 20	Feature Engineering, Data Cleaning and Advance ML
Nov 21	Nov 28	Interim Report

Phase 2: Model Development (December 1st – February 18th)

Nov 29	Dec 4	Progress Slides Submission
Dec 5	Dec 9	Progress Presentation
Dec 10	Jan 1	Initial Model Building and Tuning
Jan 2	Feb 18	Deep Dive into Metrics, Relationships and Advance Models

Phase 3: Finalization and Report Writing (February 19th – May 23rd)

Feb 19	Mar 20	Final Model Testing and Validation
Mar 21	Mar 27	Draft Report
Mar 28	Apr 14	Final Report Completion
Apr 15	Apr 30	Final Edits and Project Video Preparation
Apr 15	May 8	Submit Project Video and Final Report
May 9	May 23	Prepare for Viva

Chapter 2: **Literature Review**

2.1 Introduction

2.1.1 Objectives of the Literature Review

This literature review's main goal is to explore how machine learning is applied within the context of predicting UFC fight outcomes. This review covers machine learning in sports, traditional UFC prediction models and machine learning UFC prediction models like my project. Key data sources, data cleaning processes and evaluation metrics will be assessed on impact on prediction performance. Any research gaps are highlighted and expanded upon at the end of the review.

2.2 Background and Theoretical Foundations

2.2.1 Machine Learning in Sports

Match outcome predictions are crucial for coaches and players, influencing tactics and strategy. Beal et al. (2019) found that Neural Networks achieved the highest accuracy (72.2%) for NCAA games, though still fell short of outperforming bookmakers.UFC prediction

Community insights, such as those from Reddit user mmabet69, highlight that records from regional circuits can be misleading compared to UFC experience. They also argued that styles significantly influence outcomes, but are difficult to predict. Accurate bettors distinguish themselves by anticipating each fighter's specific approach, especially when statistical data is limited early in a fighter's UFC career.

2.3 Fight Outcome Prediction

2.3.1 Predictive Analysis of UFC Fights - G. Walsh 2021

Walsh built a predictive model trained on a subset of a dataset found on Kaggle, The Ultimate UFC Dataset. The aim of the project was to predict the winner of each match. Parameter tuning along with other machine learning methods were used to increase the accuracy of predictions.

Walsh applied Neural Networks to his data, producing accuracy of 62% towards predicting the victor of a match. The features that were removed from the original dataset were mainly regarding the rank of each fighter. This meant that the only algorithm could only compare the fighters from their averaged career performance metrics, rather than their relative skills. If a system was implemented to analyse relative skill, there may be room for improvement on the results.

2.3.2 Predicting UFC matches using regression models - S. Apelgren, C. Eklund 2024

The authors aimed to predict UFC match winners using logistic and Bayesian regression but limited their analysis by excluding fighters with fewer than 20 matches or extreme win rates, including top-tier and undefeated fighters. This filtering removed many unpredictable yet crucial cases that a robust model should handle. Additionally, their model was validated on just 20 matches, accounting for only 0.5% of all UFC fights. Without a clear sampling method, accuracy metrics are unreliable and not truly reflective of real-world performance.

2.3.3 Data-Driven MMA Outcome Prediction Enhanced by Fighter Styles: A Machine Learning Approach - J. Yin 2024

Yin used K-means clustering to group UFC fighters into Striker, Grappler, and All-Rounder categories, then applied various machine learning models to predict fight outcomes. Data suitability was confirmed using KMO and Bartlett’s tests. Including fight styles improved model accuracy by 2 - 4%, supporting the hypothesis that style enhances prediction quality. Ensemble learning achieved the highest accuracy (65.52%), outperforming individual models like logistic regression (63.99%). However, the model’s limitation to 10 features likely excluded impactful variables, such as control time, reducing its overall effectiveness.

2.4 Rankings

A common issue in UFC analytics is the inability to accurately measure a fighter’s absolute skill. Official rankings are determined by a private media panel, with unclear criteria, making them unreliable for statistical analysis. To address this, Trixster Productions developed an Elo rating system adapted from chess to track fighters’ abilities based on match outcomes. This transparent, performance-based system offers a more objective foundation for that could be implemented for machine learning predictions.

2.5 Metrics and Performance Evaluation

2.5.1 Accuracy of Model Predictions

When evaluating performance of a prediction model, we can use validation. Providing insufficient training data does not provide enough samples, meaning it does not identify meaningful correlations. This is known as underfitting. However, too much training data can lead to the model memorizing the training data rather than identifying key trends. This is called over fitting. (A. Tokuç, 2024). The Pareto principle is a common split, where 80% of the data is used for testing, and 20% is used for validation.

2.5.2 Evaluation metrics of Models

Since we will have continuous data, such as strikes landed, and discrete data such as number of matches, we will need evaluation metrics for continuous and discrete data.

Continuous data can be evaluated with the following metrics:

Mean Absolute Error, the average of the absolute differences between predicted and actual values.

Mean Square Error, the average squared difference between predictions and actual values. Larger errors are exacerbated.

Coefficient of Determination shows how well predictions align with the actual results with values closer to one being better.

Discrete data can be evaluated with the following metrics:

Precision, true positives / total positives.

Recall, true positive / (True Positives + False Negatives).

F1-score, the mean of precision and recall, providing a balance between the two factors.

2.5.3 Similar Project Performance Evaluation

In the Predicting UFC matches using regression models project, the performance metrics are substantially higher than in the other two projects. However, their performance was evaluated on a very small sample and their data was filtered to have fighters that had easier to predict fights.

In the clustering project by J. Yin, high precision of accuracy was provided. It is likely that the accuracy metrics of the models were generated with an ideal volume of validation data, due to the high knowledge in data science being demonstrated. However, no other forms of performance evaluation were provided.

2.6 Gaps in Existing Research

2.6.1 Integration of Elo rankings in Machine Learning

No ranking systems were used in any of the reviewed projects. This may be due to the UFC organization pairing up very similarly skilled fighters together. However, there is no research of evidence to prove that.

Implementing an Elo ranking system could provide a greater insight to the relative skill of fighters. Since there is no research done on the impact of a ranking system within UFC prediction, it would be worth investigating due to the gap in knowledge.

2.6.2 Lack of Use of Text-Based Fight Descriptions from LLMs

Another unexplored area is the use of large language models (LLMs) to extract qualitative insights about fighter styles from textual descriptions. None of the reviewed research used natural language outputs to generate features beyond structured statistical metrics. This represents a major gap, as LLMs can summarize fight footage or stylistic tendencies in human-like ways, such as describing a fighter as "a pressure-heavy southpaw with a strong clinch game", can then be incorporated into predictive models. Incorporating this type of text-based feature engineering could provide a richer, non-numeric dimension to fighter analysis that mimics how experts and commentators interpret matchups.

2.7 Literature Review Digest

2.7.1 Summary

This literature review has explored various projects of machine learning implemented to predict UFC fight outcomes. Critical analysis was used to determine strengths and weaknesses of the projects, including data sources, evaluation metrics, research gaps. While prior studies have achieved good accuracy, there are several limitations incorporated that if addressed, would allow for much further innovation.

The review has noted the importance of addressing the gaps in knowledge, such as the limited exploration of win method predictions and integration of Elo ranking systems within prediction models. Addressing these gaps would lead to the development of a more robust prediction model.

2.7.2 Future Direction

Enhancing Prediction Capabilities:

Future research should investigate sequential prediction approaches to forecast not only the winner of a UFC fight, but also the win method (e.g., knockout, submission, decision) and the finishing round. This layered approach aligns more closely with how humans and bookmakers analyse fights and would add more depth to model predictions.

Incorporating Elo Rankings:

Implementing an Elo rating system to quantify fighter skill could offer a new dimension of context in fight analysis. By capturing changes in relative ability over time, Elo ratings can improve both model accuracy and interpretability, particularly in scenarios involving newer or rising fighters.

Exploring Additional Features:

Including underutilized performance metrics such as control time, strike accuracy, and fight pace could fill current gaps in modelling fighter dominance. These metrics are especially relevant in decisions, where cumulative performance is critical. Future work should also address the lack of defensive statistics, which are crucial for modelling how fighters neutralize their opponents' strengths.

Integrating Text-Based Fight Style Descriptions:

A promising and underexplored direction is the use of large language models (LLMs) to derive qualitative, text-based features about fighters' styles and tendencies. These can be extracted from analyst commentary, fight previews, or generated synthetically using LLMs like ChatGPT. This allows for the inclusion of non-numeric attributes such as stance, aggression level, or preferred techniques, factors that play a significant role in real-world matchup analysis but are often omitted from structured datasets. Integrating such stylistic descriptors could bridge the gap between human fight analysis and machine learning.

Improving Evaluation Methodology:

Future projects should adopt standardized validation techniques such as 80-20 data splits, stratified sampling, and cross-validation to ensure fair and representative performance assessment. Evaluating models on a broad and diverse dataset helps avoid misleading accuracy from imbalanced or cherry-picked data.

Comprehensive Models:

Combining unsupervised learning techniques (e.g., clustering fighters by style or statistics) with ensemble learning methods could yield models that are both highly predictive and interpretable. This dual benefit makes them more suitable for both academic exploration and practical use in forecasting real fight outcomes.

Chapter 3: **Primary Survey**

3.1 Research Methods

3.1.1 Interviews

Within the interviews, my main point of focus is on inaccuracies of predictions and how certainty can be accounted for. I would like to find out what factors are most impactful with upsets within fights. This will allow me to better create an accurate algorithm that will account for these factors.

3.1.1.1 Interviewees

The main interviewees will be coaches and fighters from KO Combat academy.

3.1.1.2 Interview Questions

1. How well do you believe you can predict any given UFC main event?
    a. Decently confident gave rounds
2. What factors make predicting a UFC match easier or harder?
    a. Styles and age, age can make a great fighter decline a lot
    b. If styles are similar, it becomes tricky to predict
3. When an upset occurs in a match, how much do the following factors contribute to an upset?
    a. Fight style matchup
    b. Game plan executed by either fighter
    c. Training camps of the fighters
    d. Luck / punchers chance
4. When a fighter dominates at the start of their UFC career, how can the fighters' results be predicted against higher skilled opponents if they have not been tested yet?

3.1.1.3 Connection to Project

With the information gathered from the interviews, we can study how upsets be accounted for when predicting a match outcome with machine learning. Some factors that contribute towards upsets, such as how the fight camp goes, is not data that we have access to.

3.1.2 Survey

Within the surveys, my main goal is to find out how accurate fighters and fans can predict outcomes of matches, as well as what contributes to the predictions. As these fights play out, we can compare the reasoning for a prediction to the accuracy of the prediction to tell what makes a prediction accurate.

3.1.2.1 Surveyors

The main surveyors will be fighters and coaches from the KO Combat academy. The survey will also be sent to friends of mine that are hardcore UFC fans.

3.1.2.2 Survey Questions

5. What are your predictions for the following matches?
    a. Bo Nickal vs Paul Craig
    b. Charles Olivera vs Michael Chandler
    c. Jon Jones vs Stipe Miocic
    d. Movsar Evloev vs Aljamain Sterling

   e. Vicente Luque vs Nick Diaz
   f. Ciryl Gane vs Alexander Volkov
   g. Alexandre Pantoja vs Kai Askura
   h. Belal Muhammed vs Shavkat Rakhmonov
6. Explain how you made your predictions
7. Do you believe that the official rankings are a good way to predict fights and why?

3.1.2.3 Connection to Project

Once the fights have occurred, we can compare the result of how the predictions did versus the outcome. Consistently accurate predictions will give a greater insight into how we can predict an outcome. The survey questions can be used to create evaluation metrics when we are testing the performance of the models.

3.2 Research Results

3.2.1 Luck

The interviewees emphasized that luck plays a minimal role in victories. They cited the example of Leon Edwards vs Kamaru Usman, where despite being dominated throughout the fight, Leon landed a knockout in the fifth round. One expert explained that while some might call it a lucky shot, the timing and execution of the kick were deliberate, removing any element of luck.

3.2.2 Injuries and fight camp

Research revealed that injuries and fight camps do not impact the outcome much. One interviewee gave an example of Sean O’Malley won the match against Aljamain Sterling despite being the underdog and having a broken rib going into the fight. Since the O’Malley’s rib was broken, he could not wrestle when preparing for the fight, which happened to be his opponent’s strongest point and his weakest.

3.2.3 Fight Style

The interviewees all agreed that a fighter's style can greatly influence their chances of winning. They referenced the example of Sean Strickland vs Israel Adesanya, where Strickland, initially considered an underdog, won by unanimous decision with all judges scoring 49-46 in his favour. The expert highlighted that Strickland's constant pressure effectively countered Adesanya's strike-and-move style, leaving Adesanya backed up against the fence with no escape.

3.2.4 Fight Plan

Experts highlighted that fight plans play a crucial role in determining fight outcomes. For example, in the Volkanovski vs. Makhachev rematches, Makhachev adapted his strategy in the second bout by exploiting a defensive weakness Volkanovski showed in the first, leading to a knockout via a well-timed head kick. Similarly, Dricus Du Plessis adjusts his approach based on opponents, using pressure and grappling against Adesanya, wild volume striking against Strickland to disrupt his rhythm, and calculated precision against Whittaker. These cases show that tailored fight strategies can decisively influence outcomes. Du Plessis won all three fights, being the underdog in all of them. He achieved these results due to his excellent fight plans which directly counter their opponent by exploiting their weaknesses.

3.2.5 Accuracy of Predictions

The survey average for prediction was 53% accuracy. There was a large tendency to vote for the more well-known fighters that have had a longer career, rather than prospects with a short career. This bias may be due to a lack of knowledge on both fighters, or could be due to bias towards a better accomplished fighter.

Bias was highlighted particularly with Movsar Evloev vs Aljamain Sterling. Evloev is an up-and-coming fighter, whereas Sterling is a previous champion with 3 world champion title defences. All predictions placed Sterling as the winner, despite Evloev winning the fight and being a strong betting favourite. This shows the average fan does not do diligent research compared to a bettor.

3.3 Risk Assessment

ID	Risk	Chance	Impact	Severity	Mitigation Strategy	Backup plan
1	Insufficient data Quality	Medium	High	Medium	Clean and preprocess data thoroughly.	Find a higher-quality data source than the current one.
2	Model under-performance	Medium	High	High	Test and fine-tune multiple machine learning models, using proper validation.	Focus on simpler models to meet baseline performance.
3	Web scraping legal issues	Low	High	Medium	Verify legal terms for web scraping sources.	Use public, crowd-sourced datasets.
4	Time Constraints	Medium	High	Medium	Create a detailed project plan, prioritizing high-impact tasks.	Scale back features for more achievable goals.
5	Technical Issues	Low	High	Medium	Regularly test and debug code, using version control for software management.	Seek supervisor help and use backups to recover progress.
6	Unforeseen ethical concerns	Low	Medium	Medium	Review ethical guidelines from the EECS ethical board.	Consult supervisors and EECS ethical boards if concerns arise.
7	Data processing limitations	Medium	Medium	Medium	Use efficient tools like PySpark for processing performance.	Reduce dataset size and use cloud resources for processing.
8	Data Storage limitations	Low	Low	Low	Optimize storage solutions to avoid excess usage.	Use cloud storage instead of local disk.
9	Not enough interviewees	Medium	Low	Low	Connect with local professionals, such as gym coaches.	Focus on one insightful interview.
10	Not enough participants for survey	Low	Medium	Low	Ensure the questionnaire has a low barrier for participation, focusing on UFC enthusiasts.	Reach out to local gyms for survey participants.
11	Poorly defined requirements	Low	High	Medium	Allocate time to fully define requirements for a proper workflow.	Consult supervisor to redefine requirements if needed.
12	Ineffective Solution	Low	Medium	Medium	Develop efficiently, focusing on solving the core problem.	Evaluate and modify the solution to fit the original problem.
13	Unforeseen requirements	Medium	Medium	Medium	Use an iterative development lifecycle to account for requirement changes.	Focus on new requirements in the next development cycle.
14	Lack of technical skills	Medium	Medium	Medium	Follow online courses (e.g., Kaggle, YouTube) to enhance technical skills.	Scale back features to require fewer technical skills.

15	Poor productivity	Low	Low	Low	Allocate sufficient time for project development and ensure a buffer for unforeseen circumstances.	Ask supervisor for productivity advice.
16	Unforeseen illness and medical issue	Medium	Low	Low	Allow extra time in the project timeline to account for unforeseen circumstances.	Simplify the project scope and set achievable goals within a realistic timeframe.

Chapter 4: **Preparation**

4.1 Requirements

4.1.1 Functional Requirements

1. Ingest and preprocess UFC fight data from scraped sources and public datasets, including round-level stats, fight methods, and timestamps.
2. Track and update fighter statistics dynamically across their career using a time-aware dictionary-based system.
3. Integrate a custom Elo rating system, modified by fight outcomes and finish methods to reflect dominance and skill evolution.
4. Engineer features based on the UFC judging criteria, including both offensive and defensive metrics for striking, grappling, and control.
5. Build ensemble machine learning models (e.g., stacking and voting classifiers with Random Forest, XGBoost, etc.) to predict the winner, finish method, and round.
6. Generate predictions in a readable format, including outcome probabilities and supporting model insights (e.g., feature importance).
7. Support flexible experimentation, including toggling feature sets like Elo, round-based stats, or control metrics.
8. Output and store processed datasets and model artifacts for reuse, enabling reproducible results.

4.1.2 Non-Functional Requirements

1. Prediction runtime should be under 10 seconds per fight, supporting near real-time use.
2. Consistent and reproducible results for the same fight input across sessions.
3. Code should be modular, documented, and scalable, with future web deployment in mind.
4. Model should exceed baseline accuracy, with a target of >60% for winner prediction using statistically robust features.
5. Visual outputs such as plots should clearly communicate model behaviour and biases

4.1.3 Data Sources

Kaggle – The Ultimate UFC Dataset

This dataset includes 118 columns of performance and outcome metrics, offering rich information for analysis. However, it is infrequently updated—the last update was in November 2024.

Official UFC Website

Provides round-by-round statistics, allowing for deeper insight into performance trends across a fight. However, it lacks pre-fight fighter rankings, which must be inferred from other sources.

Greco1899/scrape_ufc_stats (GitHub)

This repository offers scripts and tables covering round-by-round stats, fighter physical attributes, and outcome data. It enables up-to-date scraping directly from the UFC site, making it a highly valuable and flexible source.

Chapter 5: **Development**

5.1 Kaggle Raw Dataset

On Kaggle, dataset is available that has been pre-processed. This dataset can be used for a baseline to compare results from my own methodology.

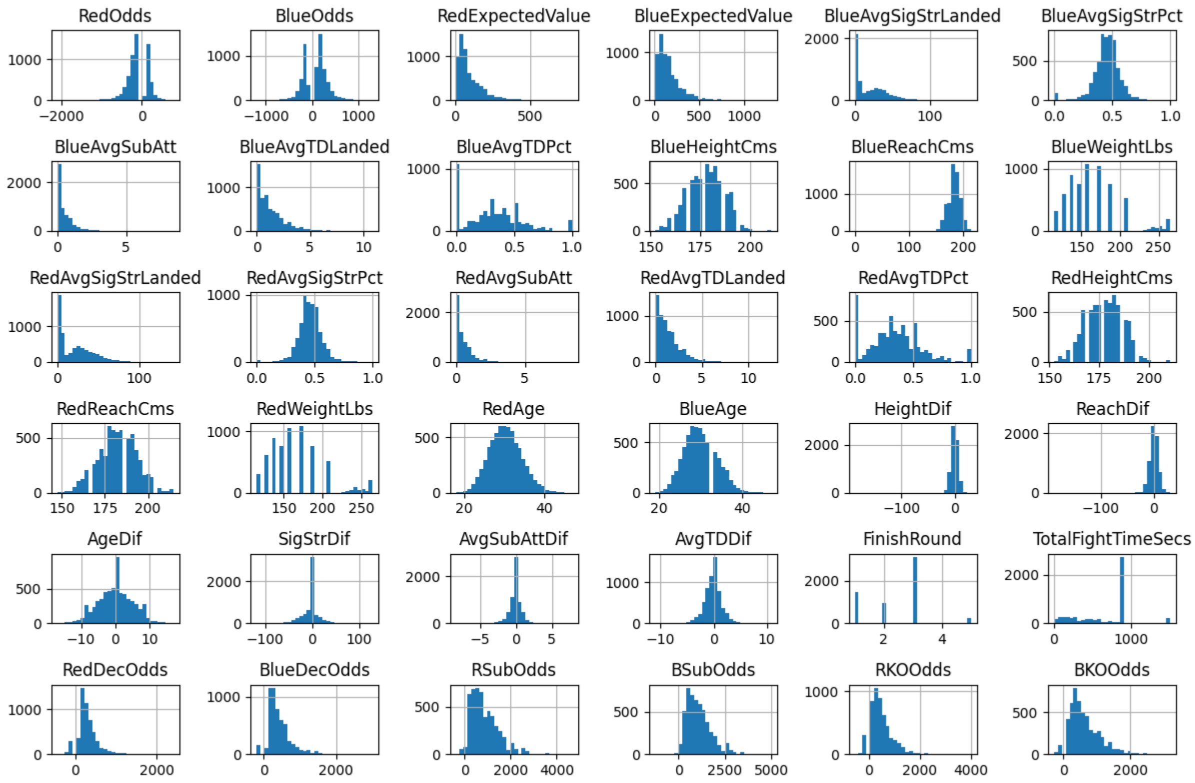
5.1.1 Exploratory Data Analysis

I performed Exploratory Data Analysis with Pandas, reviewing basic details, feature distributions and correlation Matrix.

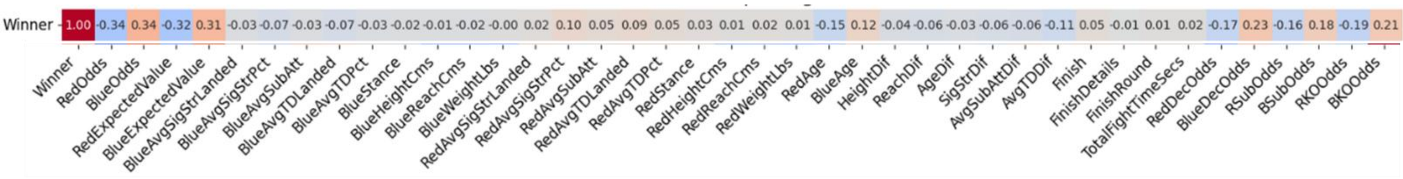
Basic Details:

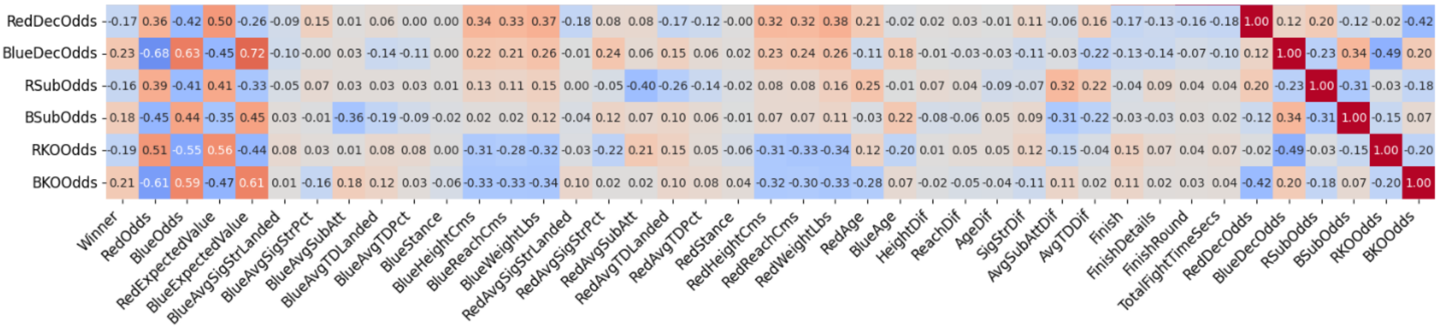
- The dataset contains 51 columns and 6441 rows.
- 36 of the columns are numerical, with the final 15 being strings.
- The strings columns describe fight details such as location, date, country etc.
- Finish Details has 3588 null values. This is because it is left blank for bouts that end in a decision rather than an early stoppage such as a knockout.
- Removing Nan values from all rows don’t have Nan as a valid output leaves 3810 rows

Feature Distributions:



The Correlation heatmap that was generated shows there are almost no correlations between fight performance statistics and winners. The strongest correlation with winners were with betting Odds. These odds were generated from DraftKings Moneyline betting odds, available on *https://www.bestfightodds.com/*. According to





Betting Odds had a stronger correlation with the data rather than the actual winners. This indicates that bettors make decisions based on statistics. One area of interest is the columns that mark the fighters' sizes, with height, weight and reach. The smaller the person is, the more likely for a decision win rather than a win through knockout.

5.1.1.1 Downsides of Kaggle Dataset

key limitation of this project is the incomplete coverage of UFC fights across available datasets. Many recent or lower-profile bouts are either missing or delayed in their availability, particularly when relying on static datasets or public APIs with limited update frequency. As a result, it is impossible to fully validate whether the machine learning models developed here generalize across the entire spectrum of UFC fights, especially for newly announced or upcoming events.

Moreover, since the current dataset has already been pre-processed and lacks a dynamic update pipeline, it cannot be easily refreshed or scaled up for real-time applications. This significantly limits the model's deployment potential in a production environment where live predictions or continual learning would be required. Future work would benefit from integrating a custom web scraper or API-based ingestion system to ensure a continuously updated and complete dataset.

5.2 Baseline Model Predictions

5.2.1 Expert Predictions

Our expert predicted 20 bouts correctly out of 28 total bouts, giving an accuracy of 71%. The expert predicted the events they felt confident and knowledgeable about, ensuring that no random guesses were presented. The predictions were made on each UFC pay-per-view cards, and several UFC Fight Night Events.

5.2.2 Large Language Model Predictions

Before UFC events, ChatGPT was utilised to predict matches based on online articles available the night before the matches. ChatGPT correctly predicted 14 out of 27 bouts, with an accuracy of 52%.

5.2.3 Kaggle Dataset.

The high accuracy score should be interpreted with caution, as the dataset is incomplete, static, and cannot be dynamically updated to include all UFC fights, limiting the model's generalizability and real-world applicability. Random Forest with cross validation was used to generate a baseline accuracy.

Description	Accuracy Score
Red to Win every time (per Kaggle Dataset)	58%
Kaggle Dataset Prediction	64.87%
Surveyor Average	53%
Expert Prediction	71%

5.3 Processed Dataset

5.3.1 The UFC Ruleset

To create the best dataset, features should be based on the UFC's judging criteria from the Unified Rules of Mixed Martial Arts, which prioritize Effective Striking and Grappling, Effective Aggressiveness, and Fighting Area Control (ABC Boxing, 2019).

Effective Striking and Grappling are defined as legal techniques that create an immediate or cumulative impact, contributing to the potential end of the match. Greater emphasis is placed on immediate impact over cumulative damage.

Effective Aggressiveness is defined as the proactive pursuit of a fight-ending sequence, where the emphasis is placed on the effectiveness of these attempts rather than mere forward movement.

Fighting Area Control is defined as determining who is dictating the pace, place and position of the match. This is only assessed if Effective Striking/Grappling is 100% equal for the competitors.

5.3.2 Raw Dataset

The GitHub repository by Greco1899 provides datasets including round by round outcome metrics, something that the raw dataset from Kaggle does not provide. Downloading the latest tables from the scrape_ufc_stats repository would allow for more up to date data. The notebooks are available that show the scraping of the statistics directly from the UFC website.

The data available from the dataset is the same as what is available on the official website. A notebook in the repository shows how the data is scraped from the official site. Since the data is gathered from the official website, we can infer this is the most accurate data.

5.3.3 Feature Selection

Despite the rules emphasizing effective techniques to end the match, preventing opponent effectiveness is equally important. The Ultimate UFC Dataset provides offensive statistics but lacks defensive metrics, neglecting a crucial component of matchups when predicting outcomes in comparison to judges' scoring.

Here are the features I have selected that best represent the rules and give an accurate depiction of matches.

Offensive Striking Metrics: Knockdown Percentage, Strike Percentage, Strikes landed per Minute, and Knockdown per Minute.

Offensive Grappling Metrics: Takedowns per Match, Takedown Percentage, Submission Percentage, Submission Attempts per Match

Offensive Control Metrics: Offensive Control per match, Offensive Control Percentage

Defensive Striking Metrics: Knockdowns Absorbed percentage, Strikes Absorbed Per Minute, Strike Defence Percentage.

Defensive Grappling Metrics: Takedown Defence Percentage

Defensive Controle Metrics: Defensive Control Percentage

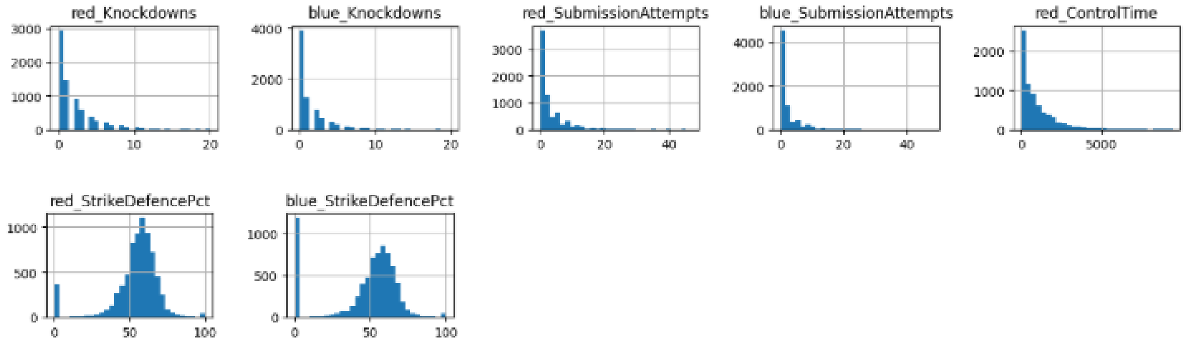
5.3.4 Creating the Processed Dataset

To process the data, a script was created builds up each fighter’s stats over time. It goes through each fight and updates stats like takedowns, strikes, control time, knockdowns, and submissions for both fighters. Before each fight, it saves the fighter's stats at that point. It also calculates how long each fighter spent in the cage. All this information is stored in a dictionary and then written into a CSV file.

5.3.5 Exploratory Data Analysis of Processed Dataset

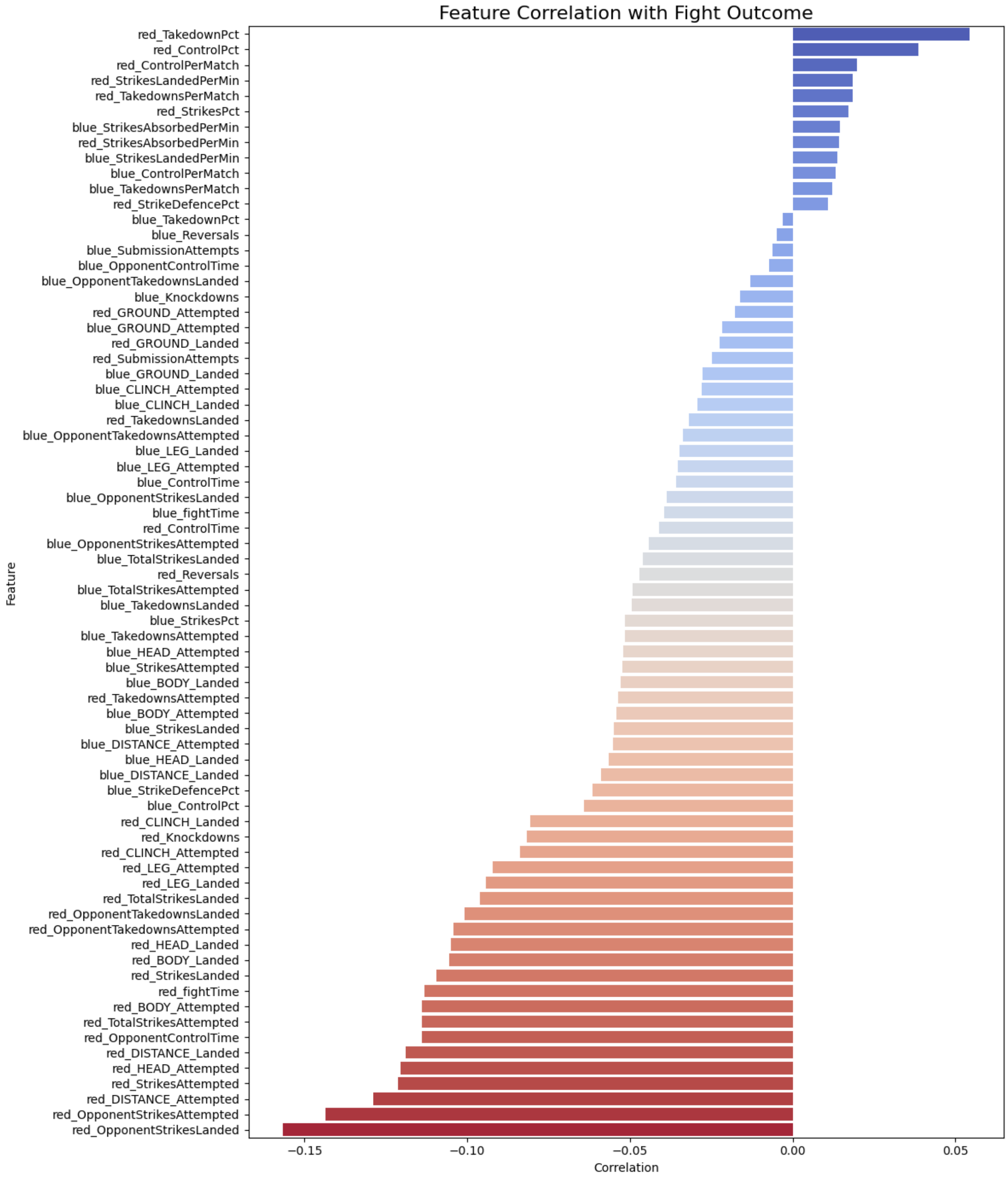
The processed dataset revealed a significant imbalance, with 65% of victories attributed to the red corner. This consistent trend will be referred to as the Red Corner Bias throughout the project.

Features tended to follow normal distribution or an exponential distribution. The type of distribution was determined by if the feature was a percentage, or totalled statistic throughout a fighter’s career.



Within the normal distribution, there are anomalies present at the extreme values of 0 and 100. These anomalies come from fighters in their first match. Looking through the data reveals many debuts are often finished with few punches being thrown and none being blocked, leading to 0% strike defence.

Below are the feature correlations with fight outcomes, calculated using Pandas' correlation method.



An initial inspection reveals that most feature correlations with the outcome are negative, which is unexpected given the assumption of an even 50-50 win distribution between red and blue corners. A possible explanation for this is the presence of Red Corner Bias, which may be skewing the results.

To mitigate these issues, data augmentation can be implemented by swapping the red and blue fighters' statistics. This would result in the gloves colour will have no impact on the result of the bout, however the fighters' statistics will have an impact.

5.3.6 Limitations and considerations for future improvement on pre-processing

While the processed dataset aligns closely with the UFC's judging criteria and captures both offensive and defensive metrics, there are key areas where improvements could enhance performance and scalability.

5.3.6.1 Betting Odds as a Feature

Implementing betting odds as a feature was shown to be highly effective in Section 5.1, since betting odd features had the highest correlation with winners. However, it was not feasible here due to limitations in publicly available APIs. Many fights, particularly older or lower-profile bouts, were missing from API responses. As a result, the odds feature was excluded from this dataset. A possible solution would be to develop a custom web scraper that systematically collects historical odds data from multiple sources. This would ensure broader coverage and offer valuable predictive power across more fights.

5.3.6.2 Manual Downloads for Dataset

While the dataset generation process is effective for research purposes, scalability becomes a concern for real-time or large-scale applications. The current method involves manual downloads and offline processing. For future deployment on a web-based platform, a fully automated pipeline using a custom web scraper would be more appropriate. This would enable regular updates, seamless integration of new fights, and up-to-date fighter statistics without manual intervention. Nonetheless, the current approach was sufficient for this project’s scope and allowed for in-depth analysis with reliable data.

5.4 Data Augmentation

5.4.1 Addressing the Red Corner Bias

By appending the swapped fighter statistics, we gain two large benefits: Doubling the number of entries we have within our dataset, allowing for models to gain a larger accuracy, and to eliminate the Red Corner Bias.

5.4.2 Generated Dataset Augmentation

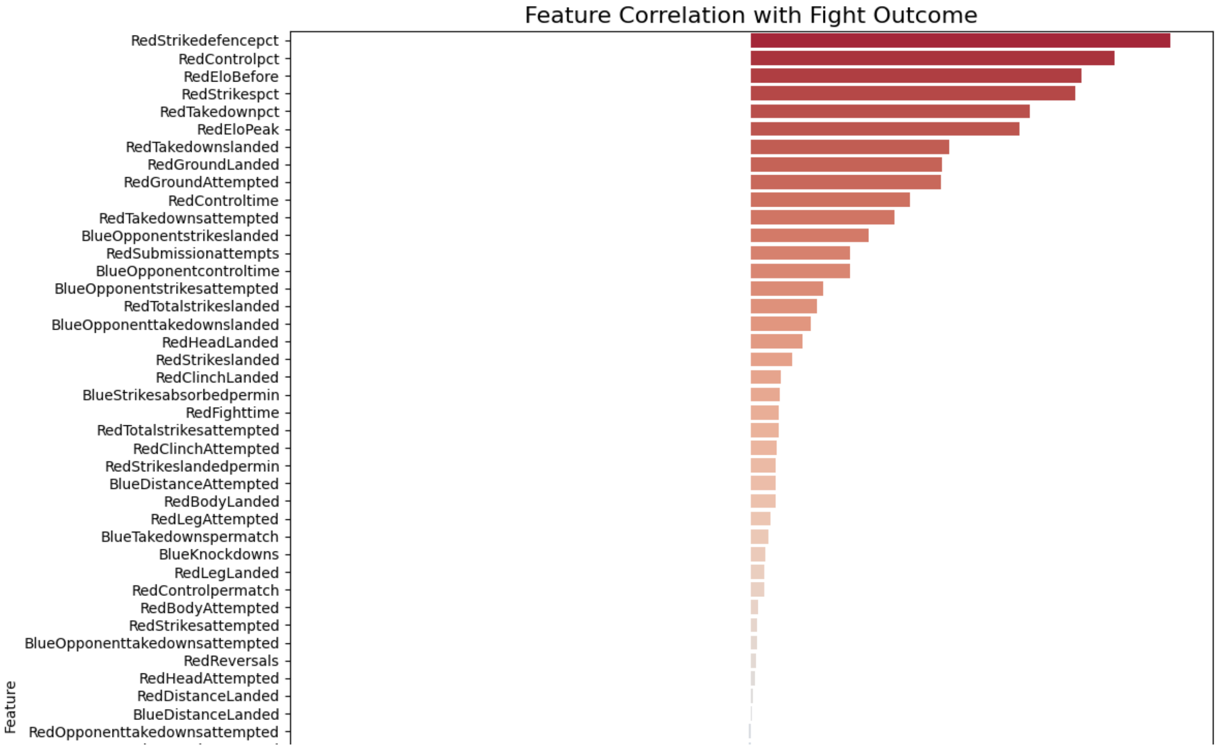
To create the augmented dataset, a duplicate of the dataset is created. The column names and the winner are swapped from red to blue and vice versa. This mirrored dataset is then appended to the original dataset and saved.

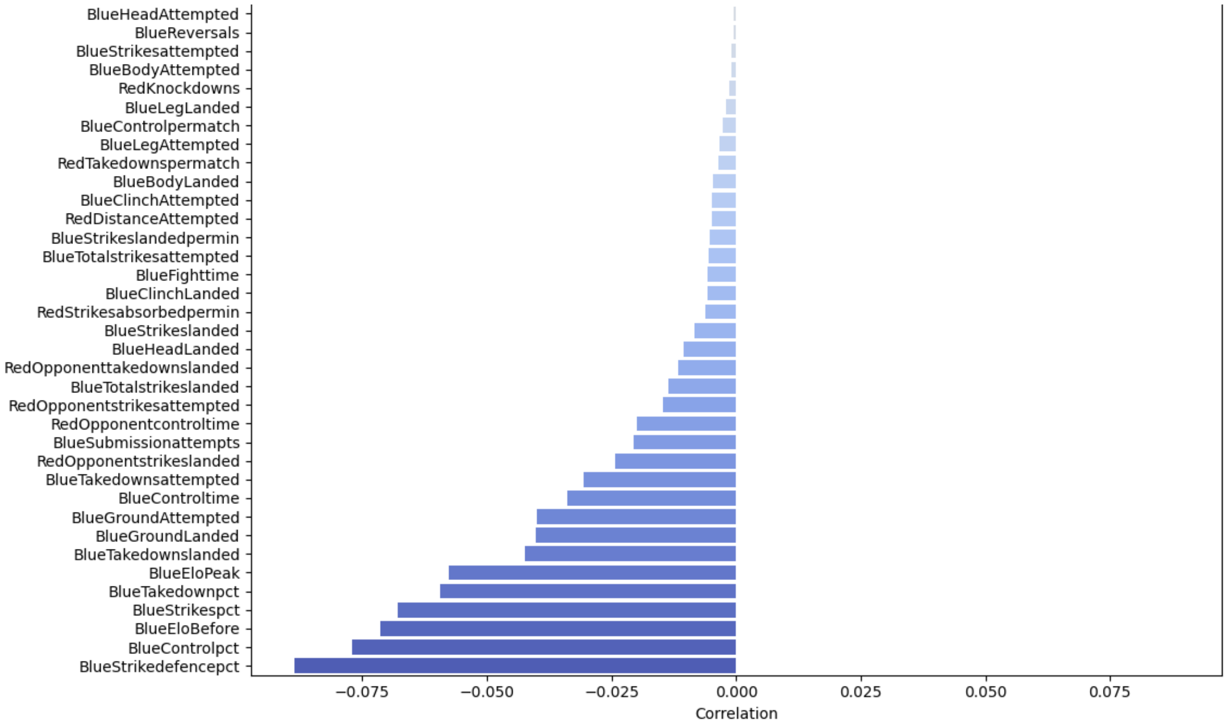
5.4.3 Analysing the Augmented Dataset

Feature correlations with outcomes were plotted for the pre-augmentation dataset, showing lower correlations for some features, such as RedOpponentStrikesLanded, which dropped from 0.16 to 0.04. This may indicate Red Corner Bias, with better strikers often placed in the red corner. Despite smaller correlation coefficients, the plots reveal that both defensive and offensive abilities increase the chance of winning. The augmented dataset proved to be more effective than the pre-augmentation version.

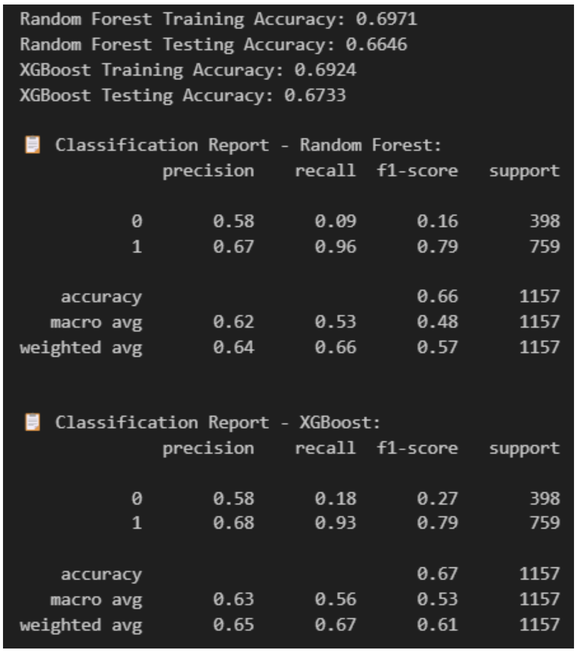
5.4.4 Comparing Machine Learning Performance Between Augmented and non-Augmented Datasets

To compare the performance of the augmented dataset and non-augmented dataset, I ran both through the same model and observed classification report, training and testing values. These can give key insights as to how the models are performing, such as if models are overfitting to data or contain bias.



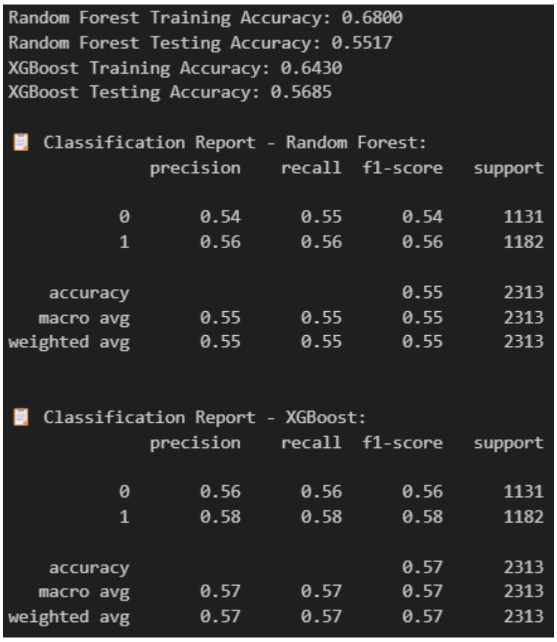


Non-Augmented Dataset



```
Random Forest Training Accuracy: 0.6971
Random Forest Testing Accuracy: 0.6646
XGBoost Training Accuracy: 0.6924
XGBoost Testing Accuracy: 0.6733

📋 Classification Report - Random Forest:
              precision    recall  f1-score   support

           0       0.58      0.09      0.16       398
           1       0.67      0.96      0.79       759

    accuracy                           0.66      1157
   macro avg       0.62      0.53      0.48      1157
weighted avg       0.64      0.66      0.57      1157


📋 Classification Report - XGBoost:
              precision    recall  f1-score   support

           0       0.58      0.18      0.27       398
           1       0.68      0.93      0.79       759

    accuracy                           0.67      1157
   macro avg       0.63      0.56      0.53      1157
weighted avg       0.65      0.67      0.61      1157
```

Augmented Dataset



```
Random Forest Training Accuracy: 0.6800
Random Forest Testing Accuracy: 0.5517
XGBoost Training Accuracy: 0.6430
XGBoost Testing Accuracy: 0.5685

📋 Classification Report - Random Forest:
              precision    recall  f1-score   support

           0       0.54      0.55      0.54      1131
           1       0.56      0.56      0.56      1182

    accuracy                           0.55      2313
   macro avg       0.55      0.55      0.55      2313
weighted avg       0.55      0.55      0.55      2313


📋 Classification Report - XGBoost:
              precision    recall  f1-score   support

           0       0.56      0.56      0.56      1131
           1       0.58      0.58      0.58      1182

    accuracy                           0.57      2313
   macro avg       0.57      0.57      0.57      2313
weighted avg       0.57      0.57      0.57      2313
```

Note: class 0 is the blue fighter and class 1 is the red fighter

While the augmented dataset shows lower raw accuracy than the original, the non-augmented model is biased, predicting red corner wins 96% of the time due to class imbalance (65% red wins). This inflates accuracy without meaningful learning. In contrast, the augmented model's balanced 56% accuracy suggests more informed and fair predictions across both classes.

5.5 Elo Ranking System

This section explores implementing the Elo Ranking System algorithm to address the issue of a bias ranking system implemented by the UFC.

5.5.1 What is the Elo Ranking System?

Elo (1978) explained that a player's rating changes based on match outcomes, with winners gaining points from losers. The point exchange depends on the ranking difference: a higher-ranked player winning results in a small point gain, while an upset win by a lower-ranked player causes a larger point shift. Over time, the rating reflects the true skill levels of players, even after upsets.

5.5.2 How can the Elo Ranking System assist the project?

The System can be used to rank fighters before matches, since the UFC does not have a proper ranking system. They rank the top 15 fighters in each division, through popular vote. Since there are 1817 currently active athletes as per the UFC website in April 2025, and only 180 total ranked fighters, under 10% of the fighters have a rank. A rank can be a significant indicator on who wins the fight. By addressing this gap, we can provide a measure of relative skill of a fighter.

5.5.3 How the Elo Ranking System works

The expected score of a player is calculated with the following formula. The expected score gives a value between 0 and 1 as a probability of winning the match.

$$\underbrace{E_A}_{\text{Expected score of player A}} = \frac{1}{1 + 10^{\underbrace{(R_B - R_A)}_{\text{Difference between Elo score of B and A}}/400}}$$

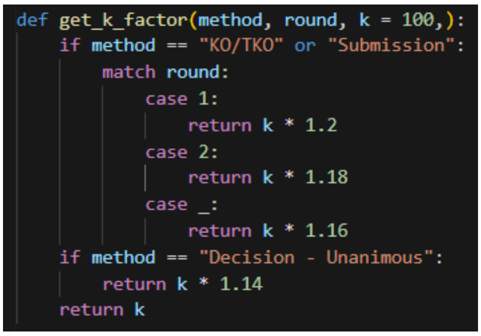
The new Elo score is the sum of the current score and the adjustment rating, which is multiplied by the difference of the performed and expected score. (Unity Technologies 2022)

$$\underbrace{R'_A}_{\text{New Elo score of player A}} = \underbrace{R_A}_{\text{Current Elo score of A}} + \underbrace{K}_{\text{Adjustement Rating (K-factor)}}(\underbrace{S_A}_{\text{Performed score of A}} - \underbrace{E_A}_{\text{Expected score of A}})$$

Scale factor of 400 is a constant that has been used throughout history, typically the Adjustment rating is altered based on the experience of the player, where the default value is typically 32 as explained by Mazzola (2020), but can be adjusted in more complex scenarios. The more experienced the player is, the lower the adjustment rating. Chess players often play daily. On the contrary, according to MMA Fight Report (2024), most UFC fighters compete 2–3 times per year. This discrepancy leads us to indicating that we should use a larger adjustment rating.

5.5.4 Implementation of the Elo Ranking System

The system was updated to adjust Elo ratings based on the method and timing of the fight's finish. More dominant outcomes like early knockouts or submissions increased the adjustment, while decisions (especially split) had less impact. A cap of 1.2 was set on the multiplier to prevent overvaluing quick finishes, which can result from luck rather than consistent skill.

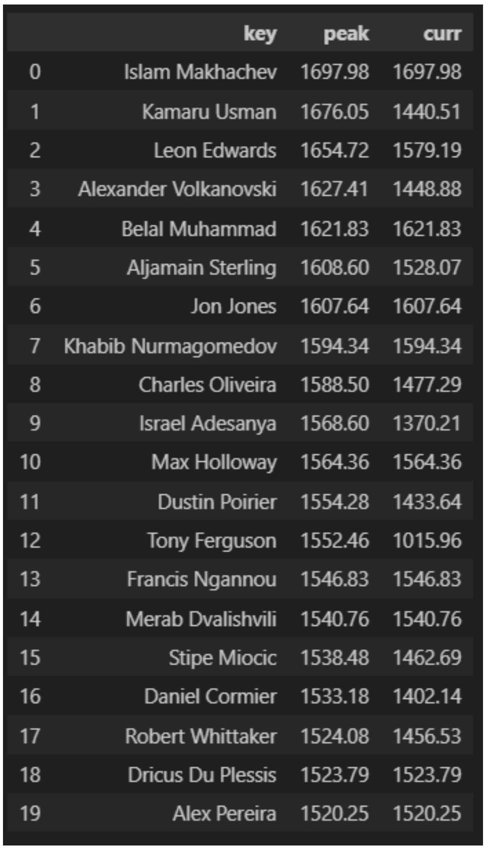


```python
def get_k_factor(method, round, k = 100,):
    if method == "KO/TKO" or "Submission":
        match round:
            case 1:
                return k * 1.2
            case 2:
                return k * 1.18
            case _:
                return k * 1.16
    if method == "Decision - Unanimous":
        return k * 1.14
    return k
```

5.5.5 Assessment of Elo Ranking System implementation

Creating a ranking system introduces a degree of subjectivity, so to assess the validity of the Elo scores, a dictionary of fighters and their all-time peak ratings was compiled and sorted to identify the top performers. Interestingly, most of the top-ranked fighters were currently active, which could be explained by either rating drift or the overall rise in athlete skill level as MMA continues to grow as a sport. All the fighters in the top rankings had held championship titles—except Tony Ferguson and Dustin Poirier—though their presence is justified given the strength of competition in the lightweight division during the mid-2010s. Fighters like Khabib Nurmagomedov, Charles Oliveira, Max Holloway, and Islam Makhachev were also part of this era, highlighting the exceptionally high level of competition.

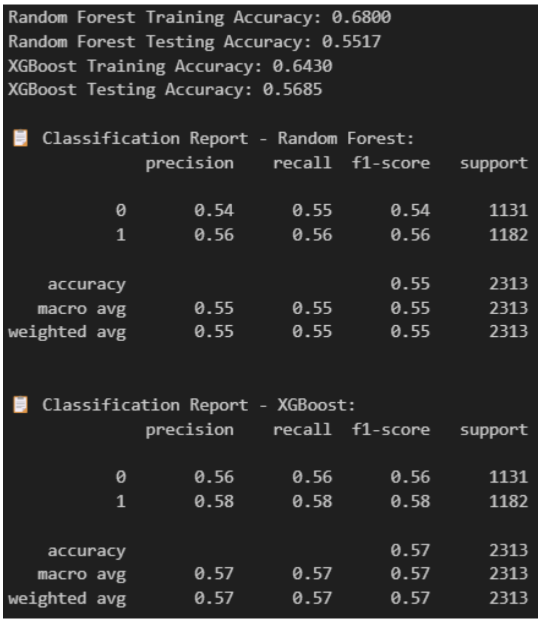
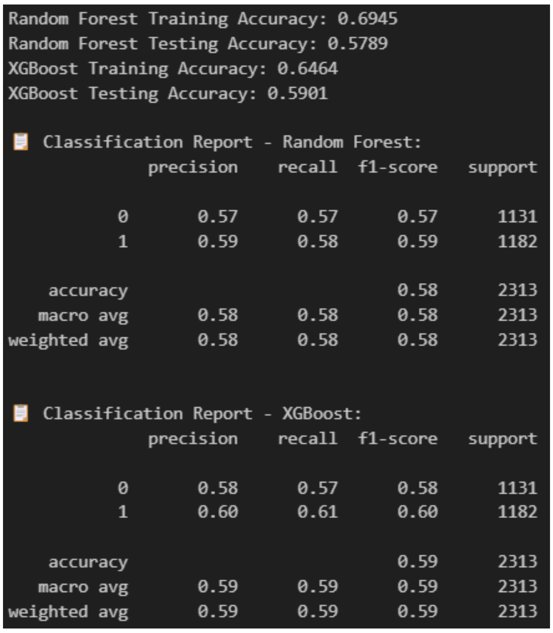
While this outcome suggests the Elo rankings are a good indicator of skill, some notable legends—such as Demetrious Johnson, Georges St-Pierre, and Anderson Silva—did not appear in the top 20 despite long championship reigns. This may be attributed to lower overall competition in earlier eras or to rating drift, where scores gradually deflate due to lack of consistent high-level opposition. As noted by Mazzola (2022), this is a known limitation of the Elo system when applied over long timeframes and across generational shifts in skill.



	key	peak	curr
0	Islam Makhachev	1697.98	1697.98
1	Kamaru Usman	1676.05	1440.51
2	Leon Edwards	1654.72	1579.19
3	Alexander Volkanovski	1627.41	1448.88
4	Belal Muhammad	1621.83	1621.83
5	Aljamain Sterling	1608.60	1528.07
6	Jon Jones	1607.64	1607.64
7	Khabib Nurmagomedov	1594.34	1594.34
8	Charles Oliveira	1588.50	1477.29
9	Israel Adesanya	1568.60	1370.21
10	Max Holloway	1564.36	1564.36
11	Dustin Poirier	1554.28	1433.64
12	Tony Ferguson	1552.46	1015.96
13	Francis Ngannou	1546.83	1546.83
14	Merab Dvalishvili	1540.76	1540.76
15	Stipe Miocic	1538.48	1462.69
16	Daniel Cormier	1533.18	1402.14
17	Robert Whittaker	1524.08	1456.53
18	Dricus Du Plessis	1523.79	1523.79
19	Alex Pereira	1520.25	1520.25

5.5.6 Machine Learning results of implementing Elo Ranking System

To analyse the impact of the system, the same models were run on the pre-processed dataset. The only difference in the datasets is that the peak Elo rating and the current Elo rating before each fight are added to the dataset.

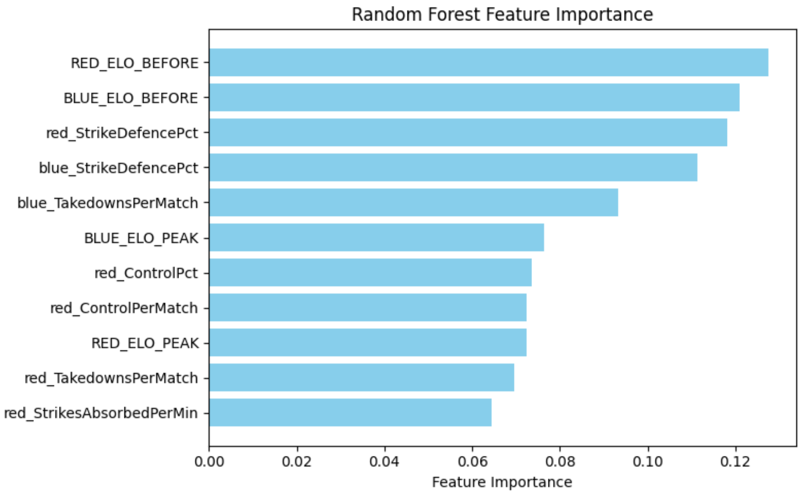
Before Elo ranking	After Elo Ranking
	

```
Random Forest Training Accuracy: 0.6800
Random Forest Testing Accuracy: 0.5517
XGBoost Training Accuracy: 0.6430
XGBoost Testing Accuracy: 0.5685

📋 Classification Report - Random Forest:
              precision    recall  f1-score   support

           0       0.54      0.55      0.54      1131
           1       0.56      0.56      0.56      1182

    accuracy                           0.55      2313
   macro avg       0.55      0.55      0.55      2313
weighted avg       0.55      0.55      0.55      2313


📋 Classification Report - XGBoost:
              precision    recall  f1-score   support

           0       0.56      0.56      0.56      1131
           1       0.58      0.58      0.58      1182

    accuracy                           0.57      2313
   macro avg       0.57      0.57      0.57      2313
weighted avg       0.57      0.57      0.57      2313
```

```
Random Forest Training Accuracy: 0.6945
Random Forest Testing Accuracy: 0.5789
XGBoost Training Accuracy: 0.6464
XGBoost Testing Accuracy: 0.5901

📋 Classification Report - Random Forest:
              precision    recall  f1-score   support

           0       0.57      0.57      0.57      1131
           1       0.59      0.58      0.59      1182

    accuracy                           0.58      2313
   macro avg       0.58      0.58      0.58      2313
weighted avg       0.58      0.58      0.58      2313


📋 Classification Report - XGBoost:
              precision    recall  f1-score   support

           0       0.58      0.57      0.58      1131
           1       0.60      0.61      0.60      1182

    accuracy                           0.59      2313
   macro avg       0.59      0.59      0.59      2313
weighted avg       0.59      0.59      0.59      2313
```

The testing accuracy increased by on average 2.5% between the two models, which is substantial in this regard. In addition, plotting feature importance shows Elo Rating had the highest importance when the model is generated which further proves the effectiveness of the ranking model.

Peak Elo ratings did seem to matter, but not as much as current Elo Ratings. A fighter could have had an unlucky match, and their peak rating is a better indicator of their skill, giving the feature high importance. However, if a fighter has aged, their peak rating is a smaller indicator opposed to their current rating since they are not as physically capable as they once were.



5.5.7 Issues with current implementation of Elo System

The Elo Ranking System was implemented to provide a more objective and dynamic method of ranking UFC fighters, but like any model, it has limitations that warrant critical analysis and highlight areas for potential improvement.

5.5.7.1 Using Judges' Scorecards for Adjustment Factor:

The current system assigns an adjustment factor based on the method of victory (e.g., knockout, submission). However, a more nuanced approach could involve using judges' scorecards as a better indicator for adjusting rankings. Fighters who win via decision, especially unanimous decisions, show a different level of dominance than those who win by split decision. The weight of each judge's scorecard could be integrated into the Elo adjustment factor to better reflect a fighter's overall performance and dominance in a fight, regardless of how it ends.

5.5.7.2 Rating Drift and Historical Context

The Elo system struggles with rating drift, as demonstrated by the current rankings heavily favouring active fighters. Historical champions like Georges St-Pierre and Anderson Silva did not rank as highly due to the skill level of their opponents. This suggests that Elo may not fully capture the evolving skill level in MMA. Incorporating an adjustment mechanism that accounts for the historical context, such as the relative strength of opponents during a fighter’s reign, could provide a more accurate reflection of a fighter’s true legacy.

5.5.7.3 Adjustment Factor for Fighters with Long Layoffs

The Elo system works well for fighters with consistent activity but may not account for fighters who take long breaks or fight infrequently. These fighters can experience fluctuations in their rankings that do not necessarily correlate with their actual skill level. An adjustment factor for activity levels or long layoffs would allow the system to better reflect a fighter's skill in relation to their fight frequency.

5.5.7.4 Dynamic Adjustment of Elo Ratings

The current adjustment factor is dynamically altered based on the fight's finish method, but it may be too simplistic. As noted, a fighter who finishes quickly might be awarded more points, but this could favour fighters who win by chance rather than skill. Future work could include additional layers to the Elo system, such as a dynamic adjustment based on fight duration, the skill of the opponent, and how decisively the victory was achieved.

5.6 Large Language Models

Fight performance metrics do not portray a full picture on what occurs during a fight. An example would be a fighter getting tired throughout the fight. It cannot be measured, although it would certainly make an impact, and models should account for conceptual and subjective factors.

5.6.1 How large language models can assist

Large Language Models such as ChatGPT use the internet as a source of information. Online articles discussing conceptual and subjective fight features. Since ChatGPT has access to such knowledge, conceptual factors could be implemented when predicting fight outcomes.

5.6.2 How can we implement Large Language Models

The text output from a large language model can be embedded, which is a process of turning the text into numbers. Embedding will produce thousands of numbers, which would need to be reduced to avoid noise. We can do this by implementing Principal Component Analysis. PCA transforms a large set of variables into a smaller one that still contains most of the information in the large set. (Jaadi 2024)

5.6.3 Testing accuracy of Large Language Model outputs

To test the effectiveness of large language models at giving an conceptual analysis on a fighter, we prompted the OpenAI API to explain a fighters fight style. For very well-known fighters, the output was excellent and very well detailed. These popular fighters have many analysis articles written about them, strengthening the API's response.

The model gpt-4o-mini was used for its fast output, good accuracy and low cost. We thought that since the prompt did not require high intelligence since we were only asking for a description of the fighter's style.

5.6.4 Testing confidence of Large Language Model outputs

To test the confidence of the API, the same prompts were generated on numerous occasions on well known fighters. Outputs from the API were very consistent with slight changes within the wording, but emphasis was placed on same concepts consistently.

5.6.5 Generation and optimisation of API calls

The process of generating fight style analysis was slow, with a single analysis taking a couple seconds each. This number would tally up quickly. With a test batch of 30 fighters, it took 3 minutes to run. Since over 4000 fighters were in the dataset, a simpler method had to be implemented. A list of unique fighter names was saved to a list. The list was broken down into batches of 5 fighters to decrease overhead between API calls. The response was then stored into a list to cache the fighters fight style. Each fighter would only have a single request. This prevents fighters with many bouts having their fight style re-generated.

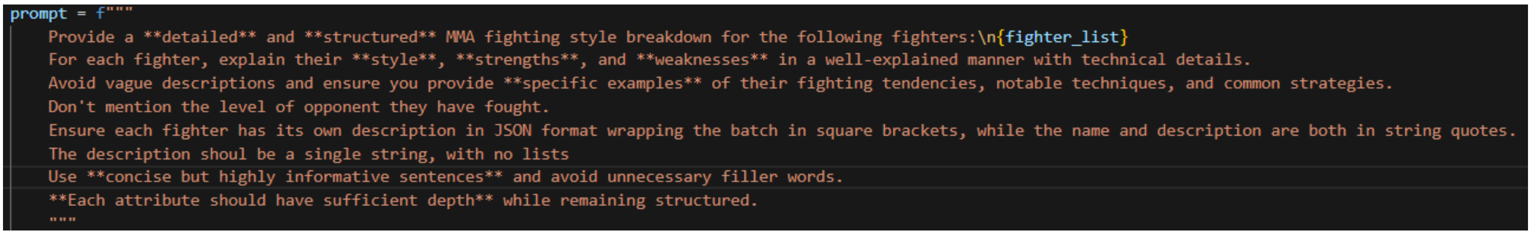
The prompt details for the output to be written into JSON format to allow for simpler conversion into an csv file with each fighter having their own dictionary. To add the fight styles to the dataset, the fight style analysis is added as an extra column for the red and blue fighters.

Batching and caching process saved the total time from 7 hours to 2 hours and 15 minutes.

One downside to this methodology is that a fighter's style may adapt over the course of their fighting career. This adds complexity to the prompt generation as the LLM would need to only use data available as of the fight date for each fight. This loses the time save of caching.

A further optimisation included was to lower the tokens from 800 to 500 per fighter. Tokens are basic units of text that LLM's compute, this could be a word, or a part of a word. Lowering the tokens comes with the trade-off of less information, but a lower computational cost when generating prompts, embedding and running machine learning models.

Here is the prompt used :



```
prompt = f"""
    Provide a **detailed** and **structured** MMA fighting style breakdown for the following fighters:\n{fighter_list}
    For each fighter, explain their **style**, **strengths**, and **weaknesses** in a well-explained manner with technical details.
    Avoid vague descriptions and ensure you provide **specific examples** of their fighting tendencies, notable techniques, and common strategies.
    Don't mention the level of opponent they have fought.
    Ensure each fighter has its own description in JSON format wrapping the batch in square brackets, while the name and description are both in string quotes.
    The description shoul be a single string, with no lists
    Use **concise but highly informative sentences** and avoid unnecessary filler words.
    **Each attribute should have sufficient depth** while remaining structured.
    """
```

5.6.6 Embedding Fight Style Analysis

To embed the fight styles, the model text-embedding-3-small was used available from OpenAI's API. This model is the latest version with the highest performance and lowest cost.

To apply the embedding, a function to receive the function is applied to each fight analysis. This is then saved as another column in the dataset. The embedding output is a list of 1536 various floats between -1 and 1.

5.6.7 Setting up Principal Component Analysis

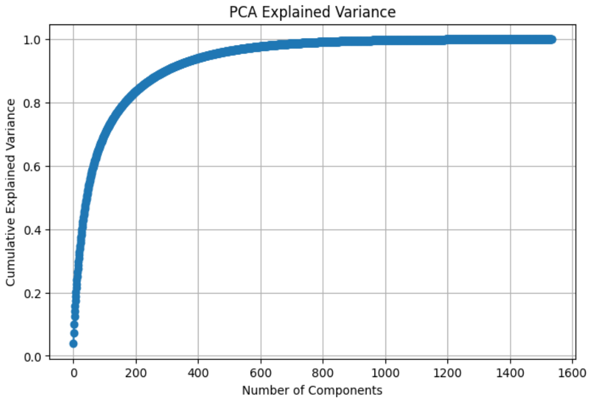
Principal component analysis (PCA) is a dimensionality reduction and machine learning method used to simplify a large data set into a smaller set while still maintaining significant patterns and trends. (Jaadi 2024)

The embeddings were stored as a single list, but for PCA, each float needed to be a separate feature. Each list was split into new features named according to the index (e.g., blue_emb_0, blue_emb_1, etc.), and these features were saved into separate data frames for the red and blue fighters.

5.6.8 Applying Principal Component Analysis

After separating the data frames, scikit's PCA is applied, and the cumulative explained variance is plotted against the number of components. The optimal number of components is found at the "elbow" of the curve, balancing variance capture and simplicity. We selected 200 components, achieving a cumulative explained variance of 0.83.

This reduces dimensionality, and the data is then fitted with the PCA transformer and concatenated to the original dataset.



5.6.9 Appending Text-Based Fight Style analysis to pre-existing dataset

To build the base model, the fight statistics and Elo ranking system were implemented. The fight styles were then appended to a separate data frame. The text-based fight style data was then embedded and ran PCA to reduce the number of components. The fight style data frame was then all combined with the base model to include the reduced fight style embeddings. The entire new data frame was then doubled and mirrored, to circumvent the red corner bias.

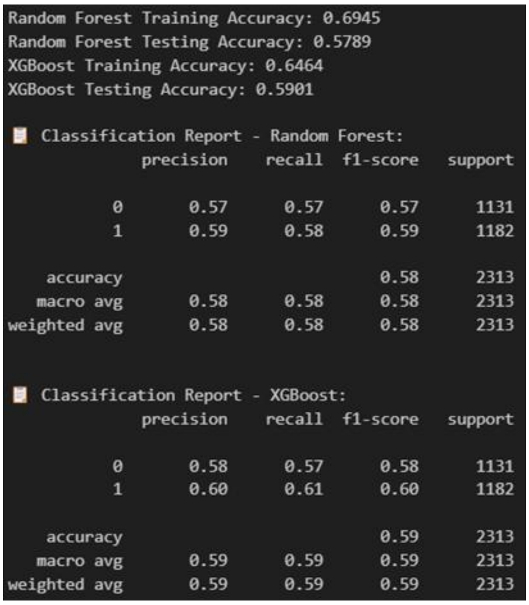
5.6.10 Models used for testing predictive performance

To achieve the results above, an exhaustive manual search was conducted to find the best parameters for XGBoost and Random Forest.

5.6.11 Comparison to baseline performance

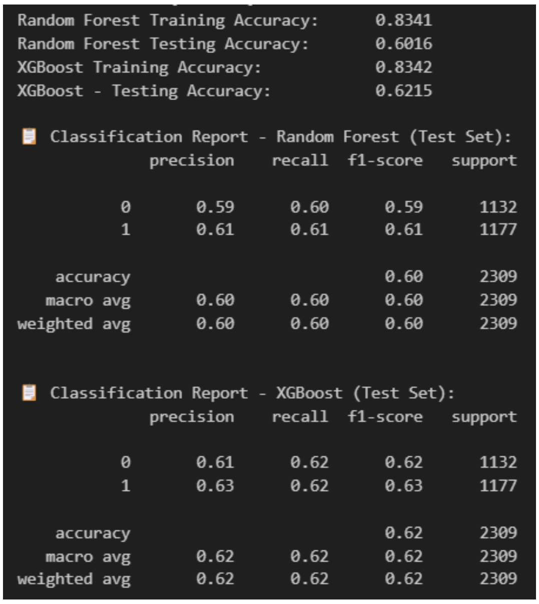
Before LLM Fight Style Analysis

(Contains Elo Ranking)



```
Random Forest Training Accuracy: 0.6945
Random Forest Testing Accuracy: 0.5789
XGBoost Training Accuracy: 0.6464
XGBoost Testing Accuracy: 0.5901

📋 Classification Report - Random Forest:
              precision    recall  f1-score   support

           0       0.57      0.57      0.57      1131
           1       0.59      0.58      0.59      1182

    accuracy                           0.58      2313
   macro avg       0.58      0.58      0.58      2313
weighted avg       0.58      0.58      0.58      2313


📋 Classification Report - XGBoost:
              precision    recall  f1-score   support

           0       0.58      0.57      0.58      1131
           1       0.60      0.61      0.60      1182

    accuracy                           0.59      2313
   macro avg       0.59      0.59      0.59      2313
weighted avg       0.59      0.59      0.59      2313
```

After LLM Fight Style Analysis



```
Random Forest Training Accuracy:      0.8341
Random Forest Testing Accuracy:       0.6016
XGBoost Training Accuracy:            0.8342
XGBoost - Testing Accuracy:           0.6215

📋 Classification Report - Random Forest (Test Set):
              precision    recall  f1-score   support

           0       0.59      0.60      0.59      1132
           1       0.61      0.61      0.61      1177

    accuracy                           0.60      2309
   macro avg       0.60      0.60      0.60      2309
weighted avg       0.60      0.60      0.60      2309


📋 Classification Report - XGBoost (Test Set):
              precision    recall  f1-score   support

           0       0.61      0.62      0.62      1132
           1       0.63      0.62      0.63      1177

    accuracy                           0.62      2309
   macro avg       0.62      0.62      0.62      2309
weighted avg       0.62      0.62      0.62      2309
```

The results show a 3% increase in performance through implementing text-based fight style analysis. These results are very impressive.

Particularly for XGBoost, implementing the text-based fight style analysis was very beneficial, boasting a 5% increase in recall, meaning the model is better at spotting the correct winner in closer or less obvious matchups.

5.6.12 Training Accuracy and Testing Accuracy Gap

A large gap between training and testing accuracy suggests overfitting, where the model memorizes training data instead of learning patterns. In this stage, the gap increased to 20-22%, compared to 12-15% in earlier stages. While this typically indicates overfitting, the complexity of the model, due to added fight style components, means the test accuracy is maximized, and the overfitting is not harmful.

5.6.13 Drawbacks and potential solutions of current LLM Implementation

There are many aspects of the current pipeline that could be altered. There are several conflicts which we have discovered which need mentioning.

5.6.13.1 Potential Data Leakage

IBM defines data leakage as when external data unintentionally influences the training process, giving the model prior knowledge it wouldn't normally have. This can lead to overly optimistic predictions, making them appear more accurate than they are (IBM, 2024).

The LLM generates output based on a fighter's entire career, which can cause leakage when analysing untested skills. For example, a fighter's wrestling ability may be included even if it hasn’t been tested until a later fight. To address this, sourcing information only up to the fight date could prevent leakage but would lead to rudimentary fight style analysis.

5.6.13.2 Changing fight styles over careers

The model doesn't account for changes in a fighter's style over time. For instance, a fighter like Dustin Poirier evolved from a brawler to a pressure fighter, and early fights may not reflect this shift. A solution could be using data from dates before each fight, though this would increase processing time. However, this leads to the next issue.

5.6.13.3 New fighter rudimentary fight style analysis

Fighter analysis for less experienced fighters is often rudimentary due to limited data. This issue can be mitigated by comparing LLM-generated analysis to expert reviews of fighters with few fights. More data leads to more accurate results, but inexperienced fighters lack sufficient data for detailed analysis.

5.6.13.4 Too simplistic prompts

Prompts may be too simplistic, limiting depth in analysis, especially for new fighters. More detailed prompts for experienced fighters could improve analysis, but for newer fighters, this may lead to overly complex results. A solution could be adjusting response length based on the LLM’s confidence level.

5.6.13.5 Low explainability

LLMs provide less explainability compared to earlier models due to the use of embeddings and natural language output. Understanding how the LLM derives its conclusions can be difficult, and investigating better formats for explainability falls outside the scope of this project.

5.7 Ensembling

5.7.1 What is Ensembling and How can it help?

Ensemble learning combines multiple machine learning models to improve predictive capability. Combining several diverse machine learning models can yield greater results than using a single machine learning model. (IBM 2024)

5.7.2 Data used for Ensembling

The data used for Ensembling is the same data that was produced in the previous chapter about text-based fight style analysis. Included in the data is the original fight statistics, the Elo Rankings and the text-based fight style analysis. The dataset has been doubled then mirrored to prevent red corner bias, and to depict a more realistic display of predictive performance.

5.7.3 Models used for Ensembling

Random Forest is a collection of decision trees that averages or votes on predictions to reduce overfitting and improve accuracy.

XGBoost is a fast and regularized gradient boosting algorithm that builds trees sequentially to correct previous errors and optimize performance.

LightGBM is a gradient boosting framework that grows trees leaf-wise (instead of level-wise), making it faster and more memory-efficient on large datasets.

CatBoost is a gradient boosting algorithm designed to handle categorical features natively and prevent overfitting with ordered boosting.

HistGradientBoosting is a scikit-learn implementation of gradient boosting that bins continuous features into histograms to speed up training and reduce memory use.

5.7.4 Model Parameters used for Ensembling

An exhaustive manual search was conducted to find the best parameters for all parameters.

5.7.5 Ensembling Voting Classifier Implementation

Voting is an Ensembling method where the final prediction is the majority vote between the various learning models.

The models used for the voting process were Random Forest, LightGBM, CatBoost and HistGradientBoosting. XGBoost was not working included because of an incompatibility with the scikitlearn Voting Classifier.

5.7.6 Ensembling Stacking Classifier Implementation

Stacking is an Ensembling method which uses a meta-model can be used to combine the outputs of other models. Typically, a logistic regression classifier is used for this role.

The models used for the stacking classifier were Random Forest, LightGBM, CatBoost, HistGradientBoosting and XGBoost. A Logistic Regressor was used as the meta model.

5.7.7 Stacking versus Voting

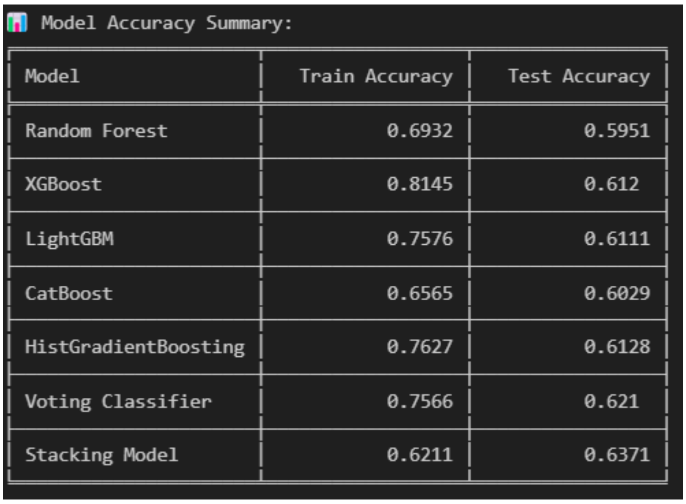
Stacking is often more accurate since it learns from model mistakes, but it’s more complex and prone to overfitting. Voting is simpler and faster, but may not capture relationships between models as effectively. Both will be implemented to find the best version.

5.7.8 Comparing Ensembling performance to baseline performance

The graph on the right displays the various accuracies of the different models. XGBoost and Random Forest do have lower performance than in the LLM’s performance test. This is from detuning the model so that it runs quicker for testing purposes.

Voting Classifier had a greater result over any individual model by almost 1%, which is again significant.

The stacking model proved to be even more effective. Stacking model boasted a 2.4% more accurate model for the most accurate individual mode, and being a further 1.5% more accurate than the voting classifier.



```
📊 Model Accuracy Summary:

| Model               |   Train Accuracy |   Test Accuracy |
|---------------------|------------------|-----------------|
| Random Forest       |           0.6932 |          0.5951 |
| XGBoost             |           0.8145 |          0.612  |
| LightGBM            |           0.7576 |          0.6111 |
| CatBoost            |           0.6565 |          0.6029 |
| HistGradientBoosting|           0.7627 |          0.6128 |
| Voting Classifier   |           0.7566 |          0.621  |
| Stacking Model      |           0.6211 |          0.6371 |
```

5.7.9 Training Accuracy and Testing Accuracy gap

The gap between training and testing accuracy for the base models is explained in section 5.7.12. The voting and stacking models also show some overfitting, but less than the base models. Notably, the stacking model has a lower training accuracy than testing accuracy due to the L2 Regularization in the meta model, which slightly reduces training accuracy but improves generalization (IBM, 2025).

5.7.10 Areas of improvement

While the results achieved are promising, there is potential for further enhancements that could lead to even better performance.

5.7.10.1 Hyperparameter Tuning

Though the parameters were manually tuned for the ensemble models, a more systematic approach using GridSearchCV could have yielded better results. GridSearch would allow for a comprehensive search across multiple hyperparameters to find the optimal configuration for each model, improving performance consistency.

5.7.10.2 System Limitations

Running GridSearchCV at scale would require significantly more computational power than my current setup allows. Using cloud computing resources would provide the necessary processing power to perform more extensive hyperparameter searches efficiently, potentially leading to higher accuracy.

5.7.10.3 Feature Subset Testing

I could have explored different feature subsets (e.g., fight stats vs. LLM-generated features) to evaluate their individual contributions to model performance, which might have improved the model's overall efficiency and interpretability.

Chapter 6: **Summary**

6.1 Key Findings

ELO Ratings Outperform Traditional Rankings
By adapting the chess-style ELO rating system to MMA, a more objective and dynamic way to rank fighters can be implemented to avoiding media bias and better reflecting real-world performance.

Fight Stats Can Predict Outcomes (To an Extent)
Using metrics like takedown accuracy, control time, and striking volume. Machine learning model achieved around 55% accuracy. Good improvement over chance (50%) in a highly unpredictable sport.

Colour Bias Needed Correction
Initial models learned to pick the red corner (typically the favourite), revealing hidden biases in the data. This was solved by mirroring fight data, forcing the model to learn from performance, not corner colour.

Adding ELO Boosts Accuracy
Integrating ELO scores into the model improved performance by 1.5%, reinforcing that relative fighter strength is a key factor in predicting outcomes.

Text-Based Fighter Descriptions Add Contextual Insight
GPT-o4-mini was used to generate fighter summaries, then embedded those into numerical vectors. After PCA compression, this added real-world insight into models and improved accuracy, reaching a final rate of 62%.

Ensemble Learning Enhanced Performance Further
By combining multiple models (Random Forest, LightGBM, etc.), the final ensemble system outperformed any individual one, providing more balanced, accurate predictions across a wider variety of fight types. Providing a further 2.5% increase over base models used.

6.2 Future Work

For future work, there are several potential areas of improvement. One key aspect is the use of cloud computing to enhance model performance, particularly for tasks like hyperparameter tuning with GridSearchCV. Cloud resources would allow for more extensive experimentation and faster computation, which is limited by the processing power of my current system. This would help achieve more optimal model parameters and potentially improve predictive accuracy.

Another area of development is creating my own web scraper to gather up-to-date fight data and expand the dataset, incorporating more diverse and recent fight statistics. This would provide a more comprehensive view of each fighter’s performance and contribute to a more accurate prediction model.

Additionally, putting an interactive version of the model online would enable real-world testing and validation. By allowing users to input fight data and view predictions manually, I could gather valuable feedback to refine the model. This would also open the possibility for further model optimization based on user interaction and results.

6.3 Final thoughts

This project demonstrated the feasibility of predicting UFC fight outcomes using ensemble machine learning techniques. It highlighted the importance of feature selection, model tuning, and interpretability. While performance was strong, real-world deployment would require further testing, real-time data, and ethical considerations. Overall, the project was a valuable learning experience in applying data science and machine learning, fields I have never learnt about previously.

6.4 Contributions

This project was carried out solely by Luke Bransby. Project supervision, guidance, and feedback were provided by Keshav Bhandari.

Chapter 7: **References**

Ultimate Fighting Championship. (n.d.) *Rankings*. Available at: https://www.ufc.com/rankings (Accessed: 15 October 2024).

Welch, L (2022) “Implementing the CMS+ Sports Rankings Algorithm in a JavaFX Environment”, Available at: https://scholarworks.uark.edu/ineguht/82/ [Accessed: 25 November 2024]

Ryan Beal, Timothy J. Norman and Sarvapali D. Ramchurn (2019). Artificial intelligence for team sports: survey Available at https://www.cambridge.org/core/journals/knowledge-engineering-review/article/artificial-intelligence-for-team-sports-a-survey/2E0E32861D031C022603F670B23B55B3%20%5b [Accessed: 26 November 2024]

A. Aylin Tokuç (2019). https://www.baeldung.com/cs/ml-underfitting-overfitting

Elo, A. (1978). *Ranking of Chess Players Past and Present*. New York: Batsford.

Best Fight Odds (2025) UFC Betting Odds. Available at: https://www.bestfightodds.com/ [Accessed: 26 March 2025]

Dabbert, M. (2021). Ultimate UFC Dataset. Available at: https://www.kaggle.com/datasets/mdabbert/ultimate-ufc-dataset [Accessed: 17 October 2024]

Greco, L. (2021). scrape_ufc_stats. Available at: https://github.com/Greco1899/scrape_ufc_stats [Accessed: 17 October 2024]

ABC Boxing (2019). Unified Rules of Mixed Martial Arts. Available at: https://www.abcboxing.com/wp-content/uploads/2020/02/unified-rules-mma-2019.pdf [Accessed: 26 March 2025]

Unity Technologies (no date) ELO Rating System. Unity ML-Agents Documentation. Available at: https://unity-technologies.github.io/ml-agents/ELO-Rating-System/ [Accessed: 18 April 2025]

Mazzola, M. (2020) ‘Implementing the ELO Rating System’, Medium, 21 July. Available at: https://mattmazzola.medium.com/implementing-the-elo-rating-system-a085f178e065 [Accessed: 18 April 2025]

Jaadi, Z., 2023. Step-by-step explanation of principal component analysis (PCA). [online] Built In. Available at: https://builtin.com/data-science/step-step-explanation-principal-component-analysis [Accessed 23 Apr. 2025].

IBM, 2023. Ensemble learning. Available at: https://www.ibm.com/think/topics/ensemble-learning [Accessed 23 Apr. 2025].

IBM. 2023. Regularization: What is it and why is it important in machine learning? Available at: https://www.ibm.com/think/topics/regularization [Accessed 2 May 2025].

IBM. 2024. What is Data Leakage?. IBM. Available at: https://www.ibm.com/think/topics/data-leakage-machine-learning?utm_source=chatgpt.com [Accessed 4 May 2025].