FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Assignment of bachelor's thesis

Title:	Statistical and sentiment-based prediction of UFC results
Student:	Roman Isaev
Supervisor:	doc. Ing. Kamil Dedecius, Ph.D.
Study program:	Informatics
Branch / specialization:	Knowledge Engineering
Department:	Department of Applied Mathematics
Validity:	until the end of summer semester 2023/2024

Instructions

Abstract: The Ultimate Fight Championship (UFC) represents an interesting opportunity to perform statistical and sentiment-based analyses to predict future results. Currently, about 6 thousand well-described results of past matches are recorded and available. In addition, great potential represent the phrases, expressions, and other verbal reactions of the fans and other interested individuals in the online environment. Their analysis will likely provide additional information for the predictions.

The aim of the thesis is to combine the information contained in past statistics and in sentiment to perform a reliable prediction of future UFC results. The steps are as follows:
1) Study the sentiment analysis theory and propose a convenient approach toward the UFC match predictions.
2) Study and propose a method combining past statistics and the results from the previous point. The statistics are available at http://www.ufcstats.com/statistics/events/completed. The source of sentiment information are various social networks (YouTube, Instagram, etc.).
3) Experimentally validate the proposed framework.
4) Discuss achieved results and indicate possible future research directions.

References:
[1] Walaa Medhat et al. Sentiment analysis algorithms and applications: A survey, 2014
[2] Ronen Feldman. Techniques and applications for sentiment analysis. Commun. ACM, 56(4):82-89, 2013.

Electronically approved by Ing. Magda Friedjungová, Ph.D. on 22 January 2023 in Prague.

[3] R.J. Hyndman & G. Athanasopoulos: Forecasting: Principles and Practice.
[4] UFC Statistics. Available online: http://www.ufcstats.com/statistics/events/completed

Electronically approved by Ing. Magda Friedjungová, Ph.D. on 22 January 2023 in Prague.

Bachelor's thesis

STATISTICAL AND SENTIMENT-BASED PREDICTION OF UFC RESULTS

Roman Isaev

Faculty of Information Technology
Department of Applied Mathematics
Supervisor: Ing. Kamil Dedecius, Ph.D.
May 12, 2023

Contents

List of Figures

List of Tables

List of code listings

I would like to sincerely thank my thesis advisor Ing. Kamil Dedecius, Ph.D., who helped me on my journey with the work you witness. I am as deeply grateful to my dearest family, friends and, lastly, Tangus group, for all the support I received from all of them throughout my studies.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis. I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as a school work under the provisions of Article 60 (1) of the Act.

In Prague on May 12, 2023 .....

Abstract

This work explores theory of Sentiment Analysis and investigates the possibilities of utilization of sentiments left on social media platforms for forecasting of UFC fight outcomes. Further, it utilizes already proposed approaches for usage of past statistics for prediction of future fights. This work proves that sentiments left online can be of great value for UFC outcome forecasting, as it achieves high accuracy for the case study which is conducted with the usage of posts from Twitter social media platform.

Keywords UFC, Sentiment Analysis, Fight outcome forecasting, MMA, Twitter sentiment, Machine Learning

Abstrakt

Tato práce se zabývá rešerší teorie oboru jenž se nazývá Analýza Sentimentů a studuje možnosti použití sentimentů uživatelů kteří je nechávají na online platformách ve formě komentářů k predikci budoucích výsledků zápasů v UFC. Dále, tato práce používá již známe metody založené na minulých statistikách a dosahuje výsledků porovnatelných s předchůdci. Nakonec, tato práce slouží jako důkaz toho, že sentimenty můžou mít vysokou hodnotu k predikci výsledků UFC, neboť dosahuje vysoké přesnosti na studovaném případě při predikci pomocí příspěvků na Twitteru.

Klíčová slova UFC, Analýza Sentimentů, Predikce výsledků zápasů, MMA, sentiment na Twitteru, Strojové Učení

Acronyms

MMA	Mixed Martial Arts
UFC	Ultimate Fighting Championship
PPV	Pay-Per-View
ML	Machine Learning
SL	Supervised Learning
NLP	Natural Language Processing
SA	Sentiment Analysis
DT	Decision Tree
RF	Random Forests
RMSE	Root Mean Squared Error
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
URL	Uniform Resource Locator

Chapter 1

Introduction

1.1 Motivation

Mixed Martial Arts (MMA) is one of the most fast-growing sports of our time, surpassing many other combat sports in terms of popularity. Undoubtedly, Ultimate Fighting Championship (UFC) is the most well-known MMA promotion company.

Having a substantial fan base, UFC consistently sells hundreds of thousands up to millions of pay-per-view tickets (PPV) for the most anticipated events. Inevitably, this leads to ever growing interest in forecasting outcomes of the future fights among both betting companies and amateur enthusiasts. Analysis of the statistics of the past fights — using Machine Learning and other methods — is the established approach for predicting the winners of the future fights.

However, it is undeniable that purely numerical approach overlooks certain aspects, such as the emotional state of fighter, which may have a serious impact on the outcome of the fight. Coincidentally, with the invention of social medias, people from all over the world have got a way to share their thoughts, emotions and opinions publicly. Considering the fact that both UFC fighters and their fans are active users of social media platforms, textual opinions left online by those people may — hypothetically — turn out to be a useful source of information for the opinion mining, thus enriching the promisable, yet inexhaustive statistical approach. Remarkably, I could not find any work which takes into consideration the mentioned textual data. Perceiving it as a lost opportunity, I decided to approach this problem myself and explore the sources of that sort of data and the potential predictive benefits associated with them.

1.2 Aim of the Thesis

The primary objective of this work is to research the relevant theoretical background and propose a hybrid way towards UFC fight outcomes forecasting, utilizing the statistical approach and enriching it with the sentiments gained from the textual data left online by UFC fans and other people related to the sport.

Firstly, I should delve into Sentiment Analysis (SA) theory. This includes the research of the fundamental SA concepts, as well as related literature review. Afterwards, focusing on a selection of the past UFC fights, I will do a case study, particular steps of which should be as follows:

- case study problem statement,
- exploration of the sources of data relevant for UFC Sentiment Analysis,
- proposal of the chosen approach for the study,
- acquisition, preprocessing and sentiment annotation of the data.

Secondly, I will carry out an investigation of the relevant application of Machine Learning (ML) methods for the prediction in sports. After that, I should proceed to discussion of the statistical approach for UFC. The practical part of this approach includes:

- acquisition of all past statistics,
- preprocessing of the data,
- assessment of ML classifier built upon this data, to validate the statistical approach per se.

Thirdly, for the hybrid approach, I will focus on the selection of matches investigated in the SA case study. For these fights, comparison of predictive accuracies of SA and statistical approaches should be carried out. The observations will be discussed. In case of findings that these methods complement each other, these experiments should be conducted:

- building ML classification model using sole statistics associated with the chosen fights,
- constructing another ML classifier using both statistics and the associated sentiments,
- comparing the accuracies of these models, in order to evaluate the impact of incorporation of the sentiment data.

Finally, conclusion should be made about the work as a whole, followed by a discussion about potential future work indications.

Chapter 2

Machine Learning

Nowadays, the widespread application of Machine Learning encompasses many data analysis fields, with Sentiment Analysis being one of them. Considering the central role of ML in both theoretical and the practical parts of this thesis, exploration of the ML background will be conducted. As a result, the subsequent chapters should be more comprehensible.

2.1 Core concepts

In essence, ML is a computational process which, with minimal human involvement, creates programs based on the given data. It is utilized in fields such as banking or spam detection [1]. According to Mitchell [2], ML is built upon the fundamental concepts like probability, statistics, and information theory, and has been demonstrated to be of an exceptional value in areas where limited human expertise makes manual development of optimal algorithms a challenging task.

ML can be divided into four categories: supervised, unsupervised, semisupervised and reinforcement learning [3]. Supervised Learning (SL), which is likely the most widespread type of ML, is the main subject of this work. As an input it needs a table of m rows and n columns. Each column represents values of a particular feature (attribute). This table also needs an associated column of m rows which is called a target variable. Such table (also referred to as a dataset [4]) can be described as demonstrated below:

features – commonly named as $x_1, \ldots, x_n$,

target variable – commonly named as y.

Depending on what values our target variable y represents, we deal with two sorts of problems:

classification – in case that y is a categorical variable (e.g., prediction of whether bank should lend money to a client, based on features such as his credit history, income),

regression – in case of that y is a continuous numerical value (e.g., car price prediction, based on features such as car model, engine parameters, production year).

The prerequisities of the prediction process involve a model which is trained on the data $\mathbb{X}$ (table consisting of feature columns) and a target variable y. Afterwards, the model can be fed with the datapoints have same features as $\mathbb{X}$, but no associated target variable value (otherwise, there would be nothing to predict in the first place). Newly created column consisting of the predicted values is usually referred to as $\hat{y}$.

In the following sections I will outline some of the frequently used ML classifiers, as classifiers form a critical part of my thesis.

2.2 Decision Tree

Decision Tree (DT) is a rather straightforward, but an effective ML algorithm, capable of handling both classification and regression problems. Its unique and outstanding property lies in fact that the way model performs classification is interpretable [5].

As an input, it takes table of features and an associated target variable. One of the ways to train the DT model for classification is called Classification And Regression Tree (CART) algorithm [3]. It is a greedy algorithm that grows the DT in a way that starting from a root node in each step it chooses the currently most effective feature for splitting the samples (datapoints) into left and right child nodes, in order to minimize the impurity of classes present in the current node. CART splits samples until some of the following events happen:

- predefined maximum depth of the tree is reached, and so it cannot grow anymore, thus leaving no room for splitting,
- any further splits would no longer be able to lower the impurity.

The latter, impurity, can be measured in two ways:

Gini impurity index $G_i = 1 - \sum_{k=1}^{n} p_{i,k}^2$,

Entropy impurity measure $H_i = - \sum_{\substack{k=1 \\ p_{i,k} \neq 0}}^{n} p_{i,k} log_2(p_{i,k})$,

with $p_{i,k} = \frac{\#\text{class } k \text{ samples}}{\#\text{total samples}}$ for the i-th node.

Once constructed (e.g., Figure 2.1), DT — in layman's terms — works as follows [3]:

1. each datapoint[1] starts at the root node,
2. at each node, impurity-minimizing condition created during DT construction is projected upon one of the datapoint features,
3. in case that this condition appears to be **False**, datapoint proceeds downwards to the right side; otherwise, if the value is **True**, datapoint goes downwards to the left side,
4. this process is repeated until this datapoint reaches a leaf node,
5. the majority class of the node where the datapoint ends-up is assigned to it as its label.

[1] In that case, we refer to datapoint as a vector (or a table row) of n values for each feature $x_1, \ldots, x_n$.

For instance, During DT construction in Figure 2.1, 0 datapoints with target variable of value **setosa** ended up in the green leaf node, followed by 49 datapoints labeled **versicolor** and 5 **virginica** datapoints. This makes **versicolor** the majority class for the green node. Therefore, every datapoint ending up in that leaf node would be classified as **versicolor**.

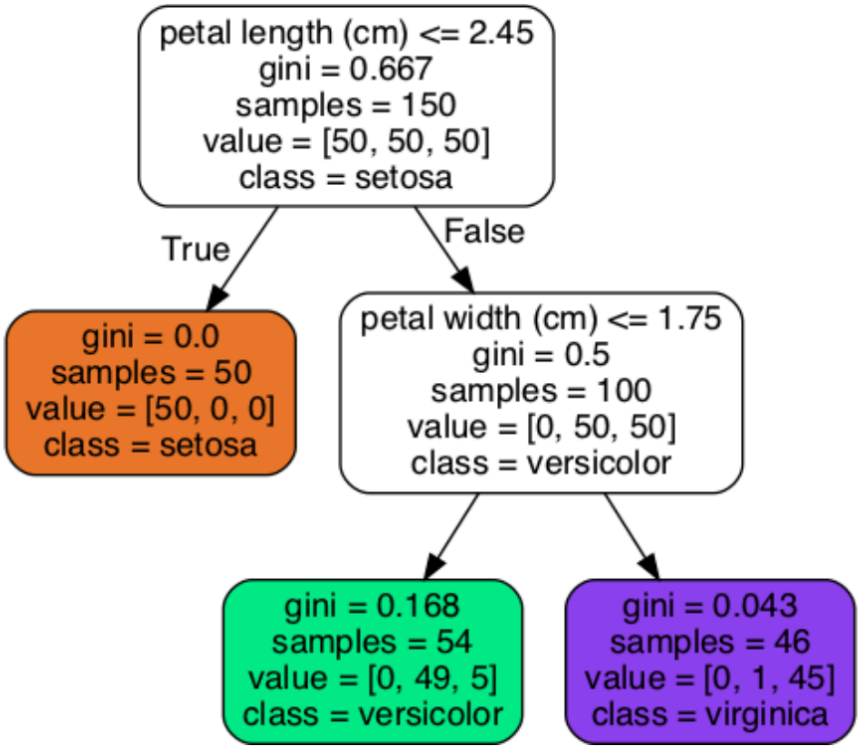

Figure 2.1 Example of DT usage for classification problem from [3]

2.3 Random Forests

Proposed by Breiman [6], Random Forests (RF) is an ensemble model[2] which can be used not only for classification, but for regression as well. The main idea behind the model is that aggregation of many simpler models (Decision Trees) leads to better results. As stated by Géron [3], even if DTs of this ensemble model are "weak learners"[3], when aggregated they can lead to substantial accuracy, outperforming even the best of the DTs.

Its workflow is quite simple. Initially, k DT classifiers are trained, each on different subset of the training data. Subsequent steps are as listed below [3]:

- each datapoint class is predicted by each of the k DTs,
- majority class is then chosen as a label for the datapoint (e.g., if $k = 11$, and 5 DTs vote **setosa**, but 6 DTs vote for **versicolor**, then the latter is considered to be the right class, as majority of the DTs voted for it)

For instance, in Figure 2.2 it is shown that after a datapoint is given to an ensemble model (RF), three simpler models (DTs) vote for class **1**, while one of them votes for class **2**. Since 75% (majority) of the simpler models vote for the former, class **1** is chosen.

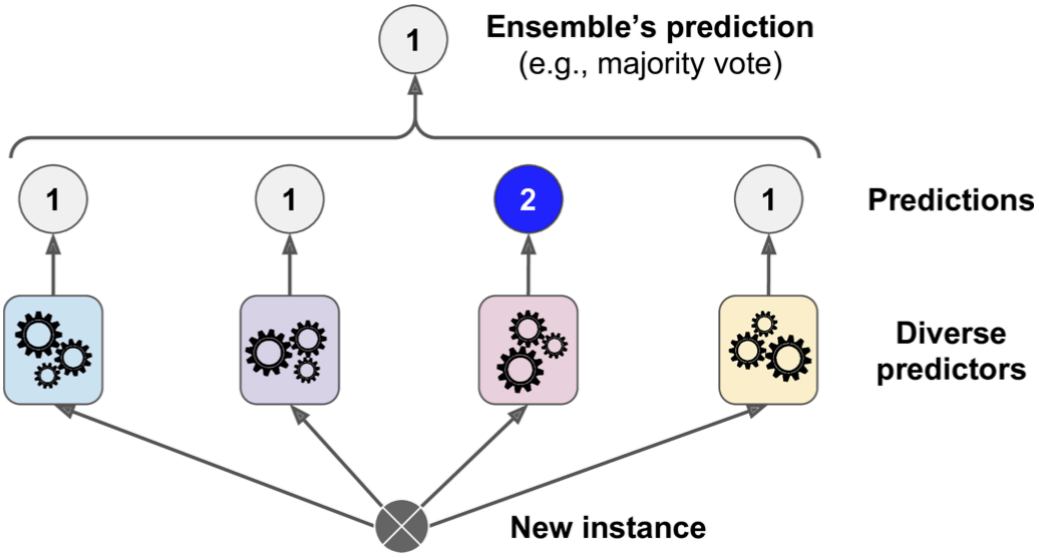

Figure 2.2 Demonstration of voting of simpler models [3]

[2] In simple terms, ensemble model is a model based on k aggregated simpler models.
[3] Weak-learner is a model with predictive accuracy only marginally better than that of random guessing.

Chapter 3

Sentiment Analysis

This chapter is dedicated to the theory behind Sentiment Analysis, which is the conventional method of discovering, understanding, and measuring human opinions expressed in text on a variety of topics. Furthermore, it outlines both the established and state of the art approaches.

3.1 Core concepts

Blending the areas of artificial intelligence and linguistics led to the emergence of Natural Language Processing (NLP) in 1950s [7]. Khurana and his co-authors [8] put forth that NLP is centered around the idea of making natural language texts comprehensible to computing machines.

Sentiment Analysis is a subfield of NLP with various applications, from movie reviews sentiment classification [9], stock market news analysis [10], to prediction of elections outcome [11].

Per Medhat et al. [12], the are three primary objectives in Sentiment Analysis. First of all, it is discovery of the opinionated textual data. Secondly, assessment of a standpoint they represent in form of a sentiment. Finally, calculation of the sentiment polarity towards the object of research.

3.2 Opinion

Work of Liu [13] provides us with an overview on what should be concerned when dealing with opinionated documents, outlying following attributes of expressed opinions:

opinion object — target entity on which the opinion is expressed (e.g., president),

opinion holder — a person, or an organization that states the opinion (e.g., electorate),

opinion orientation — it represents a negative, or a positive sentiment of the opinion holder towards the opinion object (e.g., supportive of president),

time — point in time when opinion is stated (e.g., one month before elections).

He proceeds with a statement that an opinion object (e.g., laptop) has components (e.g., speakers, processor) and attributes (e.g., sound quality, workflow fluency). Collectively, components and attributes form the features of the object. Liu notes that all of these features can be expressed with any of the synonyms associated with them (e.g., the way an object looks can be described with synonymous words such as style, elegance). For instance, Figure 3.1 demonstrates features of two cellular phones, which are opinion objects.

3.2.1 Direct opinion

Direct opinion is an opinion, which is clearly states a positive, or a negative attitude towards an opinion object, without focusing on any other objects [14] (e.g., "I like Coca-Cola.", "I hope that president X wins the elections.", "The new book of X is great.").

3.2.2 Indirect opinion

Oftentimes, opinion holders do not share their opinions directly, but rather in form of comparative opinions (e.g., "I like Coca-Cola over Pepsi, because it's much sweeter."). The goal of SA is to find out which object is preferred by a holder. Feldman et al. [15] highlight that according to Jindal and Liu [16] a limited set of words can suffice to capture 98% of comparisons. These words can be divided into:

comparative adjectives and adverbs (e.g., "worse", "more", "less", or words that end with -er, such as "stronger"),

superlative adjectives and adverbs (e.g., "worst", "most", "least", or words ending with -est, for instance, "strongest"),

additional words (e.g., "superior", "beat", "outnumber", "than", "over").

3.3 Sentiment classification

Sentiment classification is a process of deciding what sentiment label should be given to a piece of text. In Sentiment Analysis, we mainly deal with three levels of sentiment classification [12]:

document-level SA — sentiment of the text unit as a whole is measured,

sentence-level SA — text is split into separate sentences carrying their own sentiment,

aspect-level SA — classification based on the target entity features.

Medhat et al. [12] point out that work of Wilson et al. [17] serves as the evidence of minor difference between the document-level and sentence-level approaches, since sentence is a short document. Feldman et al. [15] emphasize that aspect-level SA is "the most fine-grained analysis"; this statement is rather supported by Liu [13], who shows how sentiment towards a cellular phone can be decomposed in terms of its features, giving more insights about opinion (see Figure 3.1).

There are two primary scenarios in Sentiment Analysis: either we have a set of text documents with assigned sentiments, or we deal with unlabeled data. The established and

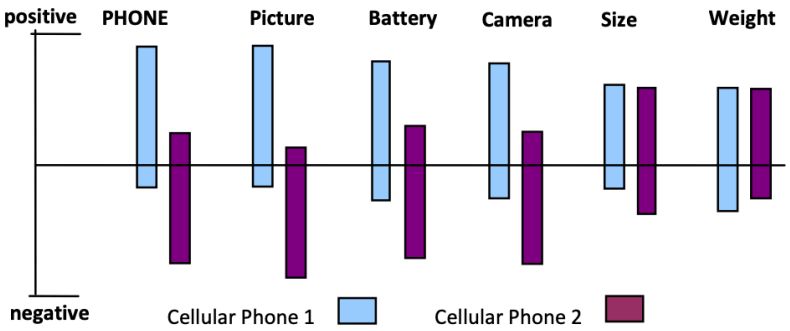

Figure 3.1 Aspect-level SA for phone sentiment classification [13]

as well as the current approaches regarding both of these scenarios will be outlined in the following sections.

3.4 Lexicon-based sentiment classification

Per Buyya et al. [18], lexicon-based approach in SA relies on a prepared sentiment lexicon. This can be represented in form of a dictionary with assigned opinion orientation for each word. They also add that application of same lexicon to different domains is rather meaningless, and that a lexicon should be created specifically for each domain.

There are three ways of creating a sentiment lexicon [15]:

manual technique relies on human annotation of the sentiment lexicon,

dictionary-based approach deals with an initially small pre-annotated dictionary expanded using solutions such as WordNet[1]; synonyms are given the same sentiment as already existing dictionary terms, antonyms are given the opposite opinion orientation,

corpus-based concept was pioneered by [19], it utilizes conjunctions such as "AND", "OR", "NEITHER-NOR", which can be divided into two subgroups [12]:

- preserving sentiment (e.g., for "John is both smart 'AND' kind", in case that we know that "smart" is a positive word, we assume that "kind" — which is not in our lexicon yet — is also positive, as it is connected by "AND" which is usually used for connecting words with similar sentiments),
- changing opinion orientation (e.g., "I was happy to buy this book, 'BUT' reading the reviews, I feel like I wasted my money." is an initially positive sentiment, inverted by what follows the "BUT" conjunction).

Paper of Vicente et al. [20] is suggestive of the fact manual and corpus-based approach effectiveness can be on par. They were able to achieve accuracy of 77% with the manual approach, and accuracy of 77.3% with the corpus-based approach. Hence, the two showed little difference.

[1] More about WordNet in Section 3.6.1.

Survey of Feldman et al. [15] highlights an original solution proposed by Kamps et al. [21]. They utilized WordNet and came up with the following formula for finding sentiment orientation:

$$SO(t) = \frac{d(t, \text{bad}) - d(t, \text{good})}{d(\text{good}, \text{bad})}. \tag{3.1}$$

For Equation (3.1), the following holds true:

- $d(t1, t2)$ is defined as the length of the shortest path between words $t1$ and $t2$ in WordNet,
- if $SO(t) > 0$, sentiment of term t is positive, and it is negative otherwise,
- previous points can be intuitively summarized by saying that if t is closer to the term "bad", its sentiment is negative, and if it is closer to "good", it is positive.

Finally, their approach is graphically represented in the Figure 3.2 provived in their work.

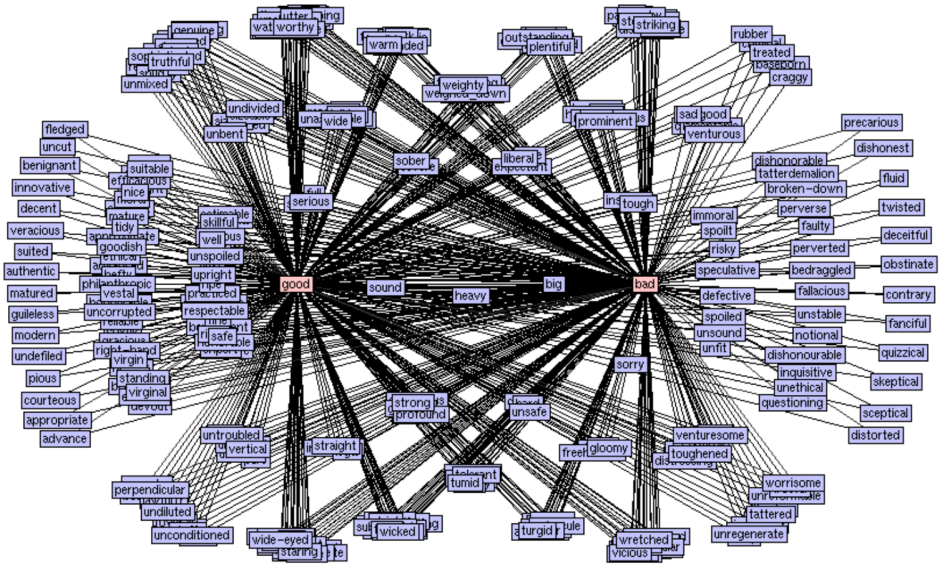

Figure 3.2 Paths between terms and "good" and "bad" in WordNet [21]

3.5 Machine Learning-based sentiment classification

Nadkarni et al. [7] state that despite the skepticism concerning the efficiency of probabilistic language models expressed in 1957 by a leading linguist Noam Chomsky [22], in 1980s Machine Learning methods utilizing probabilities turned out to be prominent tools for the tasks of NLP. For instance, in their work Baid et al. [9] used ML methods Naive Bayes, Random Forest and K-Nearest Neighbours to identify polarity of tweets containing movie reviews. With the named methods they achieved accuracies of 81.45%, 78.65% and 55.30% respectively.

In contrast to the lexicon-based method which measures sentiments using dictionaries with pre-assigned opinion orientations for words and phrases, ML-based approach needs a dataset with already assigned sentiment labels to be applicable.

Finally, ML-based approach requires a number of data preprocessing steps. It is caused by the fact that Machine Learning algorithms work with numerical features, and cannot "understand" pure text.

3.5.1 Preprocessing

First and foremost, text document must be cleaned of the noise. Some of the ways to achieve it are demonstrated by Jianqiang and Xiaolin [23] and Pradha et al. [24]:

- text is transformed into its lowercase form, to make sure that words such as "Hello" and "hello" are represented the same way, which reduces the number of features required for ML methods,
- numbers are removed, as they usually do not indicate sentiment,
- words such as "won't", "can't", "don't" are transformed into "will not", "cannot" and "do not" respectively,
- stopwords such as "a", "an", "the", "is" and "at", "not" are removed (per Jiangqiang et al., most researchers see impact of stopwords for SA task as negative [23]),
- punctuation is removed along with other special characters,
- words that have more than 3 consecutive vowels have this sequence shortened down to 3 vowels, which is especially convenient when dealing with data such as posts on Twitter (e.g., "gooooood" becomes "goood", and then it is up to the researcher to decide whether "goood" is meant to be, say, "god" or "good", or leave it as it is)

Finally, tokenization may be applied, during which text documents are transformed into their representations in form of string arrays (e.g., "john loves neapolitan pizza" becomes ["john", "loves", "neapolitan", "pizza"]).

Thus, we end up with data which is ready for the process of feature selection.

3.5.2 Feature selection

Text documents must be transformed into a tabular data representation, where each feature column contains numbers only. Mejova [25] presents some ideas of what kind of features can be engineered for Machine Learning methods:

term presence — 1 if word is present in the text document, 0 otherwise,

term frequency [2] — n, where n is a number of occurrences of this word in the text,

TF-IDF [3] — generally speaking, this metric gives higher weight to terms that appear often in a particular document, while appearing seldom in a dataset as a whole,

[2] Also called BoW — Bag of Words.
[3] Term Frequency-Inverse Document Frequency

n-grams — 1 if n specific words are present consecutively, 0 otherwise.

Applying same methods to three different datasets, in [26] Mejova and Srinivasan demonstrate that among adjectives, verbs, and nouns, adjectives have the strongest indication of the sentiment (see Figure 3.3). Therefore, utilization of Part-Of-Speech (POS) tagging[4] is another thing to consider during feature selection for the problem at hand.

	Pang & Lee		Jindal		Blitzer	
Run	Acc	# features	Acc	# features	Acc	# features
ADJ	0.781	13,546	0.901	21,150	0.772	16,217
VB	0.690	11,845	0.885	20,739	0.748	16,853
NN	0.756	26,965	0.882	84,510	0.758	60,034
ADJ ∪ VB ∪ NN	0.846	43,223	0.921	111,675	0.851	81,095
ACT	0.678	3997	0.902	3997	0.674	3997
SWN	0.819	52902	0.875	52902	0.797	52902
WNA	0.693	2367	0.876	2367	0.656	2367
run 1	0.858	50,917	0.926	218,103	0.864	153,789
majority	0.500	—	0.779	—	0.510	—

Figure 3.3 In stand-alone performance, adjectives are the most effective POS sentiment indicators [26]

Other conclusions that can be drawn from the work of Mejova et al. [26] are demonstrated in Figure 3.4:

- the higher the n in stand-alone n-grams, the worse is accuracy (see runs 1, 6, 7),
- combination of n-grams (1- and 2-grams; 1-, 2- and 3-grams) can improve the accuracy,
- less sophisticated term presence ("bin") does not necessarily lead to lower accuracy than term frequency ("TF"), as in run 1 and run 4 for the "Pang & Lee" dataset they achieve accuracies of 0.858 and 0.859 respectively, with slight advantage for the term presence approach,
- stemming (transformation of word to its base form to save space) can lead to lower accuracy, as shown for run 1 and run 2 for every dataset (affecting "Pang & Lee" the most, with a drop from 0.858 to 0.848 after stemming),
- preserving negations instead of discarding them during stopwords removal — as shown in runs 1 and 2 — can improve accuracy (inspired by Das et al.[27] who appended "--n" at the end of the word which appears after negation, Mejova et al. add "NOT-" before the negated word; e.g., "don't care" leads to construction of feature "NOT-care", and "never been" results in feature "NOT-been").

[4] Depending on the sentence context, POS tagging identifies whether given word is an adjective, verb, noun, etc.

Run #	Stem-ming	TF vs binary	Neg. words	n-gram	Pang & Lee			Jindal			Blitzer		
					Acc	F_n	F_p	Acc	F_n	F_p	Acc	F_n	F_p
1	no	TF	no	–	0.858	0.860	0.856	0.926	0.655	0.959	0.864	0.841	0.881
2	yes	TF	no	–	0.848	0.849	0.847	0.925	0.655	0.958	0.862	0.839	0.880
3	yes	bin	no	–	0.841	0.841	0.841	0.926	0.684	0.958	0.858	0.835	0.875
4	no	bin	no	–	0.859	0.859	0.858	0.925	0.677	0.958	0.859	0.836	0.876
5	no	TF	yes	–	0.866	0.868	0.864	0.929	0.667	0.960	0.867	0.845	0.884
6	no	TF	no	2	0.851	0.858	0.843	0.910	0.496	0.951	0.855	0.825	0.877
7	no	TF	no	3	0.788	0.816	0.751	0.877	0.075	0.934	0.816	0.776	0.832
8	no	TF	no	1,2	**0.875**	**0.879**	**0.869**	0.913	0.547	0.952	0.879	0.856	0.896
9	no	TF	no	1,2,3	0.830	0.843	0.815	**0.947**	**0.748**	**0.970**	**0.896**	**0.876**	**0.910**
10	no	TF	no	phrase	0.767	0.783	0.749	0.881	0.228	0.936	0.813	0.768	0.844
bl	majority rule				0.500	0.500	0.500	0.779	0.126	0.874	0.510	0.430	0.570

Figure 3.4 Performance evaluation with different settings [26]

3.6 State of the art

In following subsections, some of the currently used technologies will be outlined.

3.6.1 SentiWordNet

WordNet[28] is a big English lexical database developed by Miller et al. [29]. Basically, it forms unions called "synsets" between the words that are nearly identical in terms of what concept they represent (e.g., car and automobile would lie in the same synset). Currently, there are 117000 synsets in WordNet. Although words that have the same meaning lie in the same synset, one word can lie in more than one synsets [30]. As, for instance, word "part" can be both a noun (piece, component of something) and a verb (to split, separate something), depending on the context).

Words in WordNet are connceted in differents ways. Some of them are as follows:

synonyms lie in the same synsets,

antonyms link from one word to its opposite in terms of meaning (e.g., from "cold" to "hot"),

hypernyms link from general concept to a concrete instances (e.g, "fruit" is a hypernym of "apple", "banana"),

meronyms connect parts of an object to the object (e.g., "leaf" can lead to "tree").

In most cases, WordNet has relations between words that belong to the same part of speech [28].

SentiWordNet [31] builds upon WordNet. While preserving the connections between the words by WordNet, SentiWordNet adds three sentiment ratings in terms of how positive, how negative and how objective (neutral) it is. The ratings range between 0.0 and 1.0 for each synset. Later it was further enhanced by the authors in [32] and referred to as SentiWordNet 3.0. Figure 3.5 gives us an idea of what synsets look like, and demonstrates 10 most positive synsets and 10 synsets with the most negative sentiments in SentiWordNet 3.0. While most of them are adjectives (e.g., in "pitiful#a#2" the "#a" indicates that it is an adjective), the top positive synset consists of nouns "good" and "goodness".

Sweeney [33] proposed an elegant approach using SentiWordNet for Twitter Sentiment Analysis. His multi-entity method addresses limits of the document-level SA applied to tweets, as the latter shallowly measures the overall sentiment, omitting sentiments expressed towards each entity of the tweet. For each tweet, his method utilizes Part-Of-Speech tagging and SentiWordNet as follows:

- opinion objects are identified,
- so-called "descriptors" (adverbs, adjectives and verbs lying within a context window of 2 words to the left, and 2 words to the right of the opinion object) are identified,
- using SentiWordNet, sentiments of these adverbs, adjectives and verbs are found,
- final tweet sentiment is the average of sentiments from the previous point.

Rank	Positive	Negative
1	good#n#2 goodness#n#2	abject#a#2
2	better_off#a#1	deplorable#a#1 distressing#a#2 lamentable#a#1 pitiful#a#2 sad#a#3 sorry#a#2
3	divine#a#6 elysian#a#2 inspired#a#1	bad#a#10 unfit#a#3 unsound#a#5
4	good_enough#a#1	scrimy#a#1
5	solid#a#1	cheapjack#a#1 shoddy#a#1 tawdry#a#2
6	superb#a#2	unfortunate#a#3
7	good#a#3	inauspicious#a#1 unfortunate#a#2
8	goody-goody#a#1	unfortunate#a#1
9	amiable#a#1 good-humored#a#1 good-humoured#a#1	dispossessed#a#1 homeless#a#2 roofless#a#2
10	gainly#a#1	hapless#a#1 miserable#a#2 misfortunate#a#1 pathetic#a#1 piteous#a#1 pitiable#a#2 pitiful#a#3 poor#a#1 wretched#a#5

Figure 3.5 Top-10 positive and negative synsets in SentiWordNet 3.0 [32]

3.6.2 Word2Vec

Proposed by Mikoliv et al. [34], Word2Vec is a neural network solution that maps a word to an n-dimensional vector space. Assumption of the model is that words that appear nearby each other in the text documents are most likely to share semantical meaning (e.g., "England", "football" and "queen" are likely to appear close to each other in the vector space, while "China" is likely to appear elsewhere). It requires training on a set of text documents from which it learns associations between the words. This can be done in two ways (Figure 3.6):

Continuous Bag of Words Model (CBOW) is the first one — trained using that technique, model tries to maximize accuracy of classifying a word given the context that surrounds it in a sentence,

Continuous Skip-gram Model (Skip-gram) is the second one — when trained using this method, given a word model tries to maximize how well it predicts the context it appears in.

Per Mikolov et al. [34], algebraic vector operations with vectors created by Word2Vec model make sense, as they state that operations done in Equation (3.2) result in a vector which has vector("Queen") as its closest[5]:

$$\text{vector}(\text{“King”}) - \text{vector}(\text{“Man”}) + \text{vector}(\text{“Woman”}) \approx \text{vector}(\text{“Queen”}), \tag{3.2}$$

where vector(t) is mapping of word t into the vector space using Word2Vec model, and similarity of the resulting vector to the "Queen" vector is measured by means of cosine similarity. Cosine distance, or cosine similarity is measured the following way:

$$cos(\theta) = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\|\|\vec{b}\|} = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}}, \tag{3.3}$$

[5] In terms of cosine distance, as given in Equation (3.3).

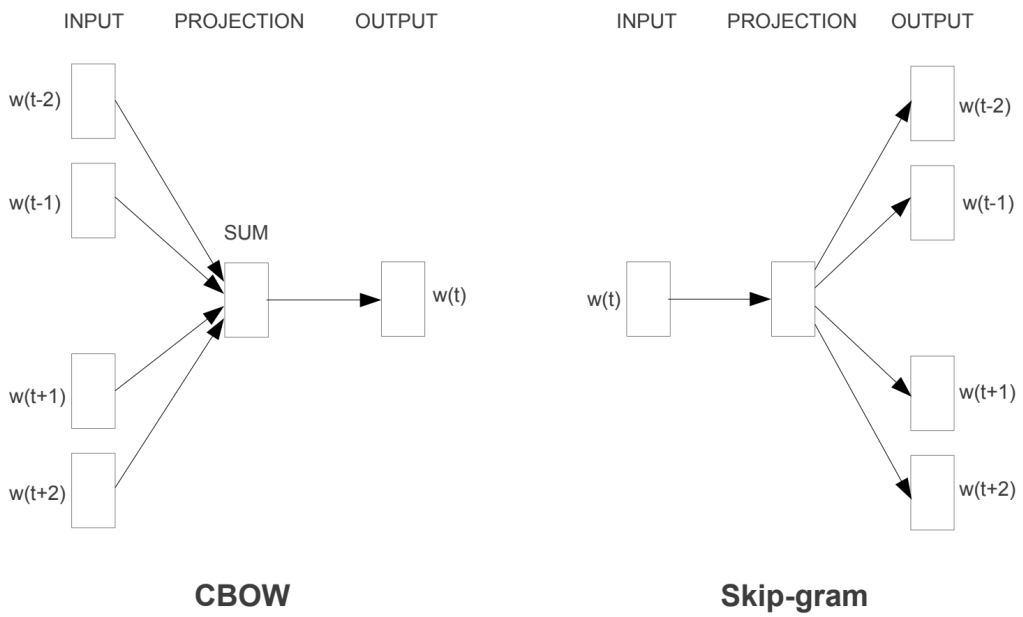

Figure 3.6 Two ways to train Word2Vec model [34]

where $\vec{a}$ and $\vec{b}$ are vectors in the n-dimensional space, θ is the angle between them and a_i is i-th element of vector $\vec{a}$.

Due to Word2Vec model, Ma et al. [35] managed to reduce dimensionality of their dataset by nearly 122 times. Starting with 61189 data features, they integrated Word2Vec and applied clustering method to the resulting vectors, grouping similar vectors into clusters. This way they managed to shrink the number of features down to 500. Despite the substantial reduction in space, the F1-score drop they observed was moderate — from 0.7524 to 0.7506.

3.6.3 BERT

Motivated by suboptimal neural network solutions which would — in simple terms — use only the preceding words to predict the following, Devlin et al. [36] developed a model called BERT. It is bidirectional model which takes into account both left and right context surrounding the word, and can be used for various Sentiment Analysis tasks.

BERT needs to be pre-trained on a large dataset of unlabaled data. One of the training corpuses of the authors consists of 2500 millions words from English Wikipedia. Once trained, the model can be fine-tuned, and thus adjusted to the problem at hand (for instance, during the fine-tuning phase Hoang et al. [37] — who utilized BERT for aspect-based SA — adjusted their model so that it would find a connection between the aspect and the text, so that the model would learn when the text context represents a sentiment).

Training on the unlabeled data is done in two steps [36]:

Masked Language Model (MLM) — during this phase, 15% of randomly chosen words are masked: 80% of them are replaced by `[MASK]`, 10% by the actual word, and 10% by a random word; after that, model learns to predict the masked elements using context,

Next Sentence Prediction (NSP) — as the name of this method suggest, this phase makes BERT learn connections between a sentence and the one that follows it, which is done by replacing the second sentence by a random sentence from the text corpus 50% of time, in order to make the model learn the right connections better.

Comprehensible graphical representation of these methods is given by Nozza et al. [38], and can be seen in Figure 3.7.

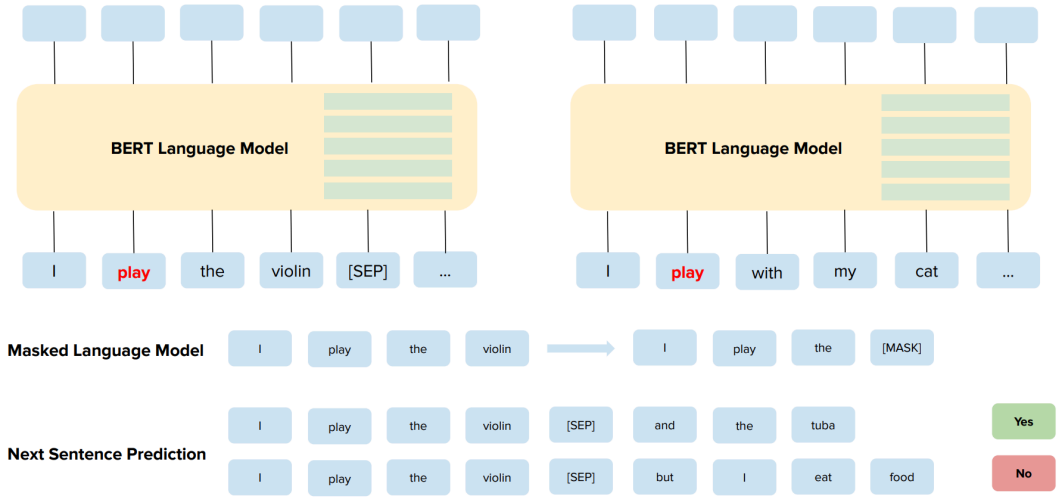

Figure 3.7 Pre-training BERT in two ways [38]

Later, Liu et al. [39] introduced a more effective version of BERT and called it RoBERTa. Among the changes they incorporated are training done on larger dataset for longer time, omitting the NSP step and using dynamic masking, which masks parts of the data every time they are given to RoBERTa during training (BERT uses static masking, where it applies masking during data preprocessing phase). Per comparisons done in [39], RoBERTa performs with higher accuracy than BERT.

3.6.4 VADER

Created by Hutto and Gilbert [40], VADER is a rule-based model intended for social media Sentiment Analysis. Per authors, in some cases VADER performs better than human for the task of sentiment classification.

VADER has a lexicon of around 7500 words, where each word has a sentiment value which spans from -4.0 to $+4.0$. Human raters were engaged for the creation of VADER. Various techniques were applied to assure the quality of ratings. For instance, during lexicon construction all words were split into the batches of 25 words. In case that for some batch rater's sentiment score for so-called "golden items" (words with pre-defined sentiment) was more than one standard deviation away from the mean in distribution of ratings by other workers, all ratings of this rater were dumped for this batch. Ratings were done in similar manner as shown in Figure 3.8.

In contrast to other lexicon-based solutions such as SentiWordNet, VADER can assign sentiment score to emojis like ":-)". Authors underline that another advantage of VADER

is that it requires less computational power than ML-based approach, without drastically sacrificing accuracy. More so, they compare their approach with Machine Learning methods and demonstrate that for 3 out of 4 datasets VADER outperforms all the other approaches. For Twitter dataset, they achieved F1-score accuracy of 0.96, while the best accuracy produced by ML methods is 0.84 for Naive Bayes, and the worst is 0.54 for SVM classifier.

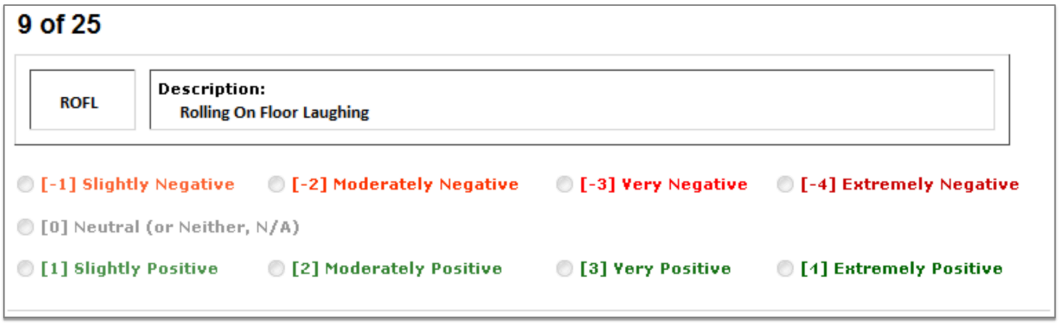

Figure 3.8 Interface similar to the one used for sentiment rating during VADER construction [40]

Chapter 4

Machine Learning in UFC Forecasting

The study done in 2018 by Hitkul et al. [41] examines the efficacy of ML methods for the UFC fight outcome prediction. Although this publication is not openly accessible, the abstract part and the references are available to the general public. As the abstract says, in UFC much more experienced athlete is not guaranteed to win against a newcomer. The sport is complex and many variables play their roles, which makes forecasting a complex problem. Another issue they point out is that there is no convenient raw data about the athletes provided. Lack of that data makes the forecasting even harder. Finally, the authors say that they employed a variety of ML methods. Despite the fact that the evaluation of applied framework is not explicitly provided freely, McQuaide [42] references this exact paper of Hitkul et al. in his work, and states that the accuracies Hitkul and his colleagues achieved are as given in Table 4.1.

Classifier	Accuracy
Perceptron	55.7%
Random Forests	58.4%
Decision Tree	51.7%
Stochastic Gradient Descent	55.1%
Support Vector Machine	57.8%
K-Nearest Neighbor	55.7%

Table 4.1 Accuracies achieved by Hitkul et al. [41]. Provided in [42]

McQuaide [42] acquires data from UFCStats[1]. She states that her own study was with the "fight-centric intention". Therefore, she excluded features like maximum number of rounds per fight and fight location. She decided to utilize statistical features such as average strikes landed or attempted, numbers of wins and losses of fighters and other statistics in his dataset. From the feature importance plot provided in his paper we can see that she not only constructs features such as average strikes landed by fighter, but also features like average

[1] http://www.ufcstats.com/statistics/events/completed

amount of strikes landed by his opponent. Although not stated explicitly, a couple of times McQuaide states that there are more than hundred features to analyze, hence we can assume that her own dataset probably has that many columns. She further points out that across whole dataset fighter in the red corner wins 62.6% of time. Employing four different models, she achieves the best results using Gradient Boosting Classifier. The average accuracy of this model was 60%. Thus, simply guessing red corner fighter to win would give an overall higher accuracy over the whole dataset. Remarkably, Decision Tree that McQuaide trains (see Figure 4.1) considers the amount of significant strikes landed by a fighter in the red corner as the most important feature, with age of red corner fighter and blue corner fighter being second and third most important features respectively (as discussed in Section 2.2, in each node DT chooses the currently most effective feature for splitting the datapoints).

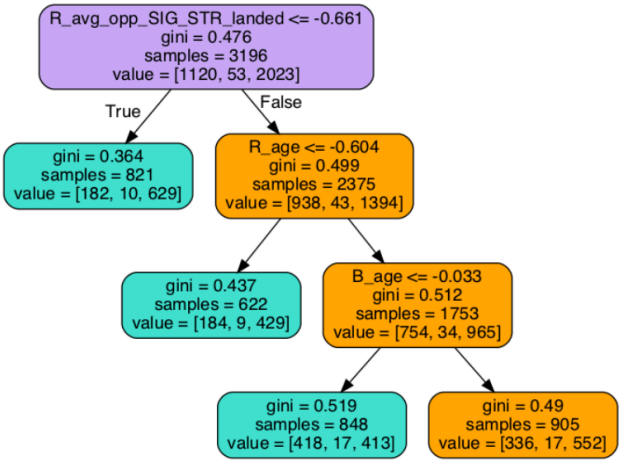

Figure 4.1 Decision Tree construction for UFC dataset by McQuaide [42]

Johnson [43] demonstrates an alternative way of looking at fight statistics variables — in his work he utilizes so-called differentials. For instance, he talks about a variable called "Strike Differential per Minute" which he calculates by finding a difference between amount of strikes a fighter lands opposed to the amount of strikes landed by his opponent. This statistic is then divided by minutes that the fight lasted. Johnson explores possibilities of constructing many different "second level" features from the basic features, as in Equation (4.1) which follows:

$$\text{Power Rating} = \frac{\text{Knockdowns} + \text{Knockouts/Technical Knockouts}}{\text{Total Strikes Landed}}, \tag{4.1}$$

where "Power Rating" is a "second level" feature. Johnson refers to variables on the right side of the Equation (4.1) — such as "Knockouts" — as "count" variables. He conducts experiments with a linear regression model and states that in the stand-alone performance "count" features are more effective than the "second level". He further concludes that the highest accuracy is reached with a combination of the simple "count" features with the "second level" variables.

Martinez-Ríos [44] acquires data from UFCStats database and applies ML methods to the constructed dataset. In contrast to the previous works, Martinez-Ríos tries to approach

this as a multiclassification problem, with the target variables he uses being “Red”, “Blue” and “Draw”. However, almost none of the models he builds happen to ever predict a draw, except one run of Decision Tree model which, however, has a rather low overall accuracy compared to the other runs — 56.63%. This finding is suggestive of the fact that it is more effective to approach the problem in binary way, predicting either victory, or defeat. Besides already mentioned McQuaide [42], Martinez-Ríos uses features that say how much damage the fighter has done to the opponent, along with those that indicate how much damage the opponent has done to the fighter. He also uses features which describe the number of wins by fighter using specific technique, be it submission or a knockout. His analysis of feature importances plot for Random Forests model shows that age of the red fighter is the most important feature, with average accuracy of significant strikes landed by red fighter's opponent being the second, and average number of significant strikes landed by red fighter's opponent being the third most imporant variable. Martinez-Ríos decides to remove from his dataset categorical variables indicating referee name, fight location and the match date, as they have a lot of unique values and applying One Hot Encoding would possibly make his solution prone to the curse of dimensionality. As for the classifier used, he utilized Decision Tree, K-Nearest Neighbor, Support Vector Machine and Random Forests. Contrary to the previously discussed works, his publication provides us not only with the measure of accuracy, but also with F1-score. This helps him to determine how well his model performs for prediction of both red and blue fighter victories — the motivation behind that is that the dataset is quite unbalanced, with blue corner fighter winning only approximately 35% of time. This can make a ML model too good at predicting red fighter's win, and increase rate of false positives, while overlooking victories of the blue fighters. He tries to experiment with feature selection techniques, such as Principal Component Analysis (PCA) or feature selection using the feature importances analysis. This approach has varying results. No apparent advantage of any of those is demonstrated. Nevertheless, author concludes that although models built upon whole dataset perform as effectively as those models built upon dataset with feature importances-based reduction, sometimes the latter outperforms the former. For instance, evaluation of the accuracies he achieves with Random Forests is demonstrated in Table 4.2, while the F1-scores for the same model can be seen in Table 4.3. Per author, the results are in accordance with the fact that the dataset is unbalanced, as in total red corner athletes won up to 63% of the fights, while blue fighters came out victorious only 35% of the time.

Dataset	Accuracy
All features	65.83%
PCA reduction	65.37%
Feature importances reduction	67.75%

Table 4.2 Random Forests accuracies achieved by Martinez-Ríos [44]

Dataset	F1-score (Red)	F1-score (Blue)	F1-score (Draw)
All features	78%	32%	0%
PCA reduction	78%	16%	0%
Feature importances reduction	79%	30%	0%

Table 4.3 Random Forests F1-scores per class achieved by Martinez-Ríos [44]

Likewise Martinez-Ríos [44], Turgut [45] scrapes statistics from UFCStats. He decides to cut off the information about fights which happened before October 1998, since a lot of data is missing in these records. He shares the finding that many features have their cells with missing values filled with "--". He replaces these fillers with "NaN" value. Inspired by the work of McCabe et al. [46] in which they calculate average performace for n previous games, Turgut decides to construct his features as the average of the matches prior to the match for which the prediction happens. Turgut recognizes the fact that simply averaging fight statistics (e.g., average number of strikes landed in all previous matches) would wrongly assume that all matches have the same length, and so all fighters have the same time window to demonstrate their craft. This approach is misleading, since not all fights last their full alocated time and end by so-called decision (judges vote for the winner), as some of them end via knockout or submission, and thus fight can end at any moment. As an alternative to this inefficient approach, Turgut normalizes the variables by calculating the average per minute (e.g., average number of strikes landed per minute of all previous matches). This way, all the variables are projected onto the same scale. He builds two models. Evaluation of his approach is demonstrated in Table 4.4.

Model	Accuracy
Random Forests	58.98%
Artificial Neural Networks	59.11%

Table 4.4 Accuracies achieved by Turgut [45]

Chapter 5

Sentiment-based Approach

Following chapter begins with a discussion about social media platforms which can be potentially utilized for UFC Sentiment Analysis task. Subsequently, choice of platform is discussed, along with the approach applied on its data.

5.1 Data sources exploration

5.1.1 Reddit

Reddit[1] is a social media platform which uniqueness stems from the fact that it divides its users into so-called "subreddits". These can be thought of as groups of people which share their opinions on a specific topic. For instance, "r/UFC" is a subreddit dedicated to UFC, while "r/books" is a subreddit where users discuss books. Subreddits' workflow is as follows:

1. person creates a post on a subreddit, which can be thought of as a start of discussion,
2. users of the subreddit get notified that a post is uploaded on the subreddit, others can find it via search bar at reddit website,
3. if users open the post, they can:
   - upvote or downvote it,
   - and, most importantly, leave a comment, which can potentially serve as a unit for Sentiment Analysis.

In idealistic scenario we expect to see comments which are strongly relevant to the text of the post. However, people are more creative than that — they often write comments which are just loosely related to the post. This leads to a threaded discussion on the topic which is rather irrelevant for the Sentiment Analysis polarity classification.

Additionally, as reddit user named "prikshet" demonstrates [47], comments under posts can have more than 30 levels of nesting. And — recursively — each nested comment can get nested again. This presents a problem for SA task, since comments in threads appear

[1] https://www.reddit.com/

in a "top-down" context[2], and it is somewhat appropriate to know what the context is, in order to understand what child comments refer to. Being a complex task in itself, it would need a separate study dedicated to it.

Lastly, after I browsed "r/UFC" and "r/MMA" subreddits, I came to the conclusion "r/MMA" is more popular and presents better opportunities for SA. I witnessed a pattern of post titles on this subreddit, which goes "[Official] UFC" and then is followed by names of fighters and, for example, text saying that this post serves as a place for discussion of the pre-fight press-conference of the fighters. However, if we type "r/MMA [Official] UFC" into the search bar of reddit, we can see that only 80-100 fights — which can be thought of as one of the most anticipated — get the privilege of having an official discussion dedicated to them on the "r/MMA" subreddit. Among these 80-100 fight posts, not all discussed fights are necessarily unique (e.g., for Islam Makhachev vs Alexander Volkanovski fight we can see 3 posts, for Islam Makhachev vs Charles Oliveira fight we can see 2 posts). Thus, we can rightfully guess that the upper bound for unique fights count would be around 40-50. That is too little, since there are more than 6000 fights in UFC as of now. For instance, I could not find a discussion for the fight between Petr Yan and Sean O'Malley[3], which was not expected, considering the fact that in total these athletes have 5 millions of Instagram followers, which makes them popular.

Moreover, in posts which serve the purpose of discussion of press-conferences, people tend to talk about the unimportant details of the conference and other subjects irrelevant to the fight. These comment threads do not provide insight on preference of . One of such examples taken from [48] is demonstrated in Table 5.1. Another issue shown in Table 5.1

Comment order	Comment level	Text
1	1	"nobody is going to talk about Conor's troll laugh that could just barely be heard on the mic?"
2	2	"Lol that was so cringey. He looked like a total retard when he did that tbh."
3	2	"I thought this dude is the joker And it was awesome"
4	2	"It was the magical moment Conor went full leprechaun"

Table 5.1 One thread of comments from Khabib Nurmagomedov vs Conor McGregor post [48]

is that 1st comment refers to Conor implicitly, stating that "**He** looked like a total retard". The opinion holder also describes his attitude towards "Conor's troll laugh" by saying "**that** was so cringey". The same phenomenon of implicit reference can be seen in the rest of the comments. Although this particular example of comment thread has only 1 level of nesting under the first comment (thus, it has 2 levels of comments) and it makes it quiet simple in its nature, SA task already gets hard. It is given by the complexity of "connecting the dots" and correctly assigning each opinion statement to each opinion object. Also, users can Reddit users can delete their accounts or their posts. After that, `[deleted]` is going

[2]As we go deeper from the first root comment down, each additional level of nesting further modifies the current context, while still preserving the context of the previous levels.

[3]Fight between Petr Yan and Sean O'Malley is interesting case for the Sentiment Analysis, since the odds of betting companies favoured Petr Yan. In that sense, victory of O'Malley was surprising.

to be displayed in place of comment. However, comments which are nested under it will be preserved. Hence, the latter lose context in which they appeared in the first place.

Summing up, despite the fact that Reddit is a scraping-friendly platform which allows its users to scrape it freely using Reddit API[4], it does not seem like either a universal or a convenient solution for the UFC Sentiment Analysis.

5.1.2 YouTube

YouTube[5] is a social media platform where people and various organisations share videos. Workflow on this platform is as follows:

1. user uploads a video on his channel,
2. subscribers of the user get notified, non-subscribers can search for the video by the keywords they expect to be present in its title,
3. if users click on a video, subsequently they can:
   - write comments under the video,
   - leave replies under the comments.

In contrary to Reddit which — as discussed earlier — can have more than 30 nested comments for each level of nesting, YouTube allows to nest only once in form of replies to a comment. This, however, does not prevent its users from doing what Reddit users do with the nesting — YouTube users compensate the absence of the nesting feature by tagging each other in the section of replies. Thus, the same problem with tracking the context of the conversation as with Reddit, but in different form.

What makes YouTube more appealing than Reddit is the fact that it has a larger userbase of "over 2 billion monthly active users" [49], compared to Reddit's "over 430 million monthly active users" [50]. This applies to UFC use-case as well. For instance, while "r/MMA [Official] Ferguson Oliveira" query does not return any results for the fight between Tony Ferguson and Charles Oliveira, if we search for "ferguson oliveira weigh-in" on YouTube we get relevant results. YouTube certainly covers more UFC fights than Reddit.

The common way of scraping YouTube is to use YouTube API[6]. The limit of the latter tops at 10000 scraped units per day [51].

[4] https://www.reddit.com/dev/api/
[5] https://www.youtube.com/
[6] https://developers.google.com/youtube/v3

5.1.3 Instagram

Instagram[7] is a social networking platform. Typical use-case scenario of Instagram is as follows:

1. user uploads a photo or a video on their profile,
2. after clicking on the uploaded item, users can:
   - leave comments under it,
   - reply to the comments.

None of the fighters have their own Reddit profiles (at least none that I have personally heard of or witnessed during my research), and most of them likely do not have YouTube channels (the only one I myself discovered is Sean O'Malley[8]). With Instagram it is different, as nowadays 500 million people use it daily [52], and many UFC fighters are active users too [53].

Several attempts were made to scrape the data from Instagram profile of Conor McGregor[9] via Instaloader[10]. Each time Instagram suspended my accounts after some comments were scraped (as far as I remember, in total around 100 comments were scraped). Despite the efforts to contact Instagram developers and ask for permission to scrape some data for the research purposes, the outcome was unsuccessful. I learned that while scraping Instagram is not illegal, their Terms of Service (ToS) are against it, and they use sophisticated algorithms to detect non-human behavior of the user.

Summing up, as of now no Sentiment Analysis study of scope similar to UFC SA task can be conducted on that platform, as Instagram — unlike Reddit and YouTube — does not provide any API and prevents all forms of scraping.

5.1.4 Twitter

Twitter[11] is a social networking platform with a daily userbase of 206 million users [54]. The characteristic which makes it the most appealing platform for Sentiment Analysis among all the mentioned social media platforms is the fact that posts on Twitter called "tweets" are atomic in terms of length (the most frequent length of tweet is 28 [55]). It is very simple for a user to open the app, express how he feels about X in one sentence and share it with the world.

In contrast to Instagram where main posts are videos or photos and thus cannot be used for (textual) opinion mining, tweets themselves already express opinions which can be used for Sentiment Analysis. Once published, all tweets can be found through execution of a query on Twitter using the keywords relevant for our study (e.g., "McGregor vs Khabib" is a legitimate query for Twitter, returning all tweets containing that sequence). Furthermore, Twitter query algorithm allows us to modify various search filters such as words we do not want to be present in tweets and specification of time interval of tweets publication.

[7] https://www.instagram.com/
[8] https://www.youtube.com/@SugaSeanUFC/featured
[9] https://www.instagram.com/thenotoriousmma/?hl=en
[10] https://instaloader.github.io/
[11] https://twitter.com/

Although Twitter has Twitter API[12], only 1500 tweets per month can be scraped for free via the API. Nevertheless, frameworks such as *snscrape* [56] allow us to scrape tweets with no limits.

5.2 Data acquisition

5.2.1 Platform and tools

Jiang et al. [57] state that target-independent approach[13] for sentiment classification is appropriate for the cases when an opinion is expressed about a movie or a product, as these opinions carry the overall attitude of its holders towards the one particular entity. However, they point out that online users may mention many targets in their posts, which would misclassify the sentiment for the target of our interest. They back up their statement with the following arguments:

- "People everywhere love Windows & vista. Bill Gates" — per Jiang et al., this tweet would most likely be treated as positive towards the target "Bill Gates", but as we can see it holds no opinion towards him; more than that, he is the opinion holder, not the opinion target,
- "Windows 7 is much better than Vista!" — here they show us a case of a comparative opinion, which would also be classified as positive for both "Windows 7" and "Vista" if treated as target-independent.

As Liu [13] puts it, while "for some individual sub-problems researchers have annotated data for benchmark testing, there is still not a comprehensive public domain corpus that can be used to evaluate all tasks in a unified way.".

Mentioned works made me cautious and not willing to blindly apply solutions such as SentiWordNet or VADER when we cannot say how good our final accuracy actually is, having no frame of reference and dealing with a domain-specific problem. After many reflections, I decided to experiment with a lexicon-based SA approach. With that in mind, I focused on Twitter, for the atomic nature of tweets.

The tools I used for data collection are *Python*, *snscrape* [56] for tweets acquisition and *pandas* [58] for storing the tweets. The code was executed in *Jupyter Notebook* [59].

5.2.2 Fights sample

To conduct a case study, I decided to focus on a sample of 23 fights. The reasoning behind choosing specifically these fights was uncomplicated — those are some of the fights I have either heard of or watched myself. If it turns out that Sentiment Analysis shows insufficient predictive accuracy for the outcomes of the bouts between fighters that an occasional UFC viewer like me has heard of, the less likely it would be effective for some less popular fighters which are known by the dedicated fans exclusively.

[12] https://developer.twitter.com/en/products/twitter-api

[13] Target-independent approach calculates the overall sentiment of a text, without differentiating between sentiments towards different entities of the opinion.

I created structures which would store name of the fight, names, usernames and nicknames of the fighters and date of the fight in `YYYY-MM-DD` format. Examples of these structures for storing fight information are given in Code listing 1. Also, in Code listing 1 we can see that fighter named "Alexander Gustafsson" has neither Twitter account, nor a nickname — I have not found none of those during my research. Thus, he is the only fighter who has no nickname and no Twitter username provided, as other fighters have at least one of these.

```
[
    'Adesanya vs Pereira',
    [
        'Israel Adesanya (@stylebender, Izzy)',
        'Alex Pereira (@AlexPereiraUFC)'
    ],
    '2022-11-13'
],
[
    'Usman vs Masvidal',
    [
        'Kamaru Usman (@USMAN84kg)',
        'Jorge Masvidal (@GamebredFighter, Gamebred)'
    ],
    '2021-04-25'
],
[
    'Alvarez vs McGregor',
    [
        'Eddie Alvarez (@Ealvarezfight)',
        'Conor McGregor (@TheNotoriousMMA)'
    ],
    '2016-11-13'
],
[
    'Jones vs Gustaffson 2',
    [
        'Jon Jones (@JonnyBones)',
        'Alexander Gustafsson'
    ],
    '2018-12-30'
]
```

Code listing 1 Examples of structures providing fight information

For instance, let us decompose the sequence "Israel Adesanya (@stylebender, Izzy)" in the first item of Code listing 1 ("Adesanya vs Pereira" fight). "Israel" is the first name of the fighter, "Adesanya" is his surname, "@stylebender" is his Twitter username and, finally, nickname "Izzy" is the way fans can refer to him on social media platforms. Furthermore, in Code listing 1 we can see that fighter named "Alexander Gustafsson" has neither Twitter account, nor a nickname — I have not found none of those during my research.

5.2.3 Sentiment annotation

In order to create a dictionary for the lexicon-based approach I did the following:

1. scrolled thousands of relevant UFC tweets on Twitter, in order to identify the patterns using which users tend to express their attitude towards fighters before the fight,

2. annotated a lexicon with either $+1$ or -1, depending on whether a pattern expressed a positive or a negative opinion respectively towards the fighter.

In total, this lexicon contains around 190 items, with only around 50 of them indicating negative sentiment — most of the tweets I ran into during scrolling expressed rather supportive sentiment than the negative, thus not many negative sentiment patterns were identified. Some of the patterns are listed in Table 5.2.

Pattern	Sentiment rating	Example ("Covington vs Usman 2" dataset)
"* beats"	+1	"@Troydan **colby beats** juiceman"
"* winning"	+1	"@LaguineeMansa Na I got **Colby winning** tonight"
"house on *"	+1	"Bet the **house on Colby** [**emojis**]"
"I got *"	+1	"**I got Colby** , Thug , And Justin tonight [**emojis**]"
"W for *"	+1	"easy **W for kamaru** tonight #UFC268"
"team *"	+1	"@espnmma Let's go, **Team Kamaru**!"
"go *"	+1	"Fight Day. **Go Usman**, Go Gaethje"
"bet *"	+1	"@jimipapifn @RumbleJunction I **bet Usman** will win against Covington. It's gonna be one hell of a fight. DC Rini#1110 REMOVED_URL"
"bet on *"	+1	"@Troydan **bet on colby**"
"scared of *"	+1	"@ufc @USMAN84kg @ColbyCovMMA Colby looks like he's a bit **scared of usman**"
"* by KO"	+1	"Chandler R1 + **Colby by Ko** or Dec + Canelo round 1-6 +11000 odds REMOVED_URL"
"* by murder"	+1	"[**...**] I saw a vid of colby and by his body language and the way he was rubbing his arm and looking the interviewer for affirmation. I got **usman by murder**"
"* by stoppage"	+1	"Predictions for #UFC268 tonight [**emojis**] I think my Chandler pick is more out of hope than actually what I think. **Usman by stoppage**, Rose by SD REMOVED_URL"
"* all day"	+1	"@KeyanKielty **Usman all day**, it'll be a murder imo"
"hope * gets"	−1	"I **hope usman gets** absolutely mopped this weekend."
"beat *"	−1	"I want Covington to **beat Usman**. Usman is a flip flopper when it comes to his faith and he's also very cringe. Plus Colby has grown on me over the last year"
"beats *"	−1	"If Colby Covington **beats Kamaru Usman** tonight, I'll show dong on twitter dot com."
"smash *"	−1	"Colby is gonny **smash usman**"
"smashes *"	−1	"[**...**] Chimaev could match up against Usman after he **smashes @ColbyCovMMA** ;"
"finish *"	−1	"[**...**] after Usman will **finish Colby** in 3 this time, people will finally realize that Usman is the best P4P fighter on earth."
"whoops *"	−1	"Man I hope Colby **whoops Usman**'s ass tonight, I don't even dislike Usman lol #UFC268"
"* is gonna get"	−1	"**Covington is gonna get** another broken jaw again [**emojis**]"
"out of *"	−1	"Not a big ufc guy but I hope to God Usman knocks the shit **out of Covington**"
"* to sleep"	−1	"Please @USMAN84kg put this man Colby **Covington to sleep** REMOVED_URL"
"sleeps *"	−1	"Im going to laugh my ass off if Colby goes out there and **sleeps usman**"
"over *"	−1	"@jeffwilton22 this weekend Canelo 7th round stoppage/ KO , Covington 3rd round KO **over Usman** ? [**emojis**]"

Table 5.2 Examples of opinion patterns and their corresponding sentiments

5.2.4 Query execution

Twitter provides its user with what they call "Advanced search", in which we can manipulate various parameters of our query, such as time window of desired tweets, keywords that should be present, keywords that should not be in these tweets. These queries can be given to *snscrape* solution too. Afterwards, it returns tweet istances, including various parameters such as id of tweet, location of its creator. In following paragraphs I will outline how these queries were constructed.

During query construction "*" in patterns of Table 5.2 would be replaced either by a first name (e.g., "Israel"), surname (e.g., "Adesanya"), full name ("Israel Adesanya"), twitter username (e.g., "@stylebender") or a nickname (e.g., "Izzy") during query construction. Thus, one fighter would have 3 to 5 queries constructed for him, depending on by how many names he can be referred to. Basically, query would be constructed as in Code listing 2, where:

exact_phrase — pattern as in Table 5.2 with "*" replaced by fighter's name, surname, full name, twitter username or nickname,

since_date — date since which we want to scrape tweets (date of the fight minus 1 day for first name, as after experiments I came to the conclusion that first name is not specific enough to scrape tweets within 7 days window before fight; date of the fight minus 7 days for all the other names),

until_date — date until which we want to scrape tweets (date of the fight),

filters — sequence "-free -live -watch -stream -streaming -PPV", which makes Twitter ignore all the tweets that containt these keywords inside it (keywords that I identified as appearing in spam tweets — if organisations want UFC viewers to join their platforms to view fights, they spam many tweets with keywords like "stream").

```
def construct_query(exact_phrase, since_date, until_date, filters):
    exact_phrase = '"' + exact_phrase + '"'
    query = exact_phrase + ' ' +
            filters + ' ' +
            f'until:{since_date}' + ' ' +
            f'since:{until_date}'
    return query
```

Code listing 2 Twitter query construction

Subsequently, constructed queries would be given to *snscrape* solution, which would extract the required tweets from Twitter platform. Expected lexicon-based sentiment for following tweets would be provided in form of label variable. This process is captured in Code listing 3. Once scraped, all the fighter's tweets (those scraped for name, surname, etc.) regarding that fight would be merged into one dataframe and saved in form of a .csv file. In total, it took me around 24 hours to scrape all the tweets. Once scraped, dataframes of both athletes that fought the fight were stored in the same folder.

```
def get_df(query, label):
    scraper = sntwitter.TwitterSearchScraper(query)
    tweets = []
    for i, tweet in enumerate(scraper.get_items()):
        data = [tweet.id,
                tweet.date.timestamp(),
                tweet.rawContent,
                tweet.user.location,
                label]
        tweets.append(data)
    df = pd.DataFrame(tweets, columns=['id',
                                       'time',
                                       'text',
                                       'location',
                                       'label'])
    return df
```

Code listing 3 Tweets dataframe construction

5.3 Data preprocessing

One of the current issues in SA is spam, as it makes SA analysis ineffective [13]. I personally ran into this problem with some spam tweets containing URLs. An instance of this issue is demonstrated in Table 5.3. The solution to this problem was to replace URLs within these tweets with a tag "REMOVED_URL" and cut off any text that follows it (as shown in Table 5.3, tweets themselves are identical, but URLs are different). After that, I got rid of duplicities, leaving only one unique instance for that kind of tweets. Due to this, I got rid of 23624 duplicated tweets (thus, the total number of collected tweets dropped from 153811 down to 130187).

Text
UFC 281 Gambling Preview: Will Israel Adesanya get his redemption, and going all-in on Zhang Weili https://t.co/PUSkqKMCY1 via @MMAFighting
UFC 281 Gambling Preview: Will Israel Adesanya get his redemption, and going all-in on Zhang Weili https://t.co/8a0wv4Otub via @MMAFighting
UFC 281 Gambling Preview: Will Israel Adesanya get his redemption, and going all-in on Zhang Weili https://t.co/aHRwf43Eh0
UFC 281 Gambling Preview: Will Israel Adesanya get his redemption, and going all-in on Zhang Weili https://t.co/r0bXyHmj8e https://t.co/kmniMpjZkO

Table 5.3 Examples of Twitter spam containing URL

Despite the attemps to assure relevance of retrieved tweets for patterns from Table 5.2 for first names by scraping tweets which were posted one day before the fight, I still happened to collect irrelevant tweets. I addressed this issue by creating a list of keywords relevant to UFC and applying it on tweets scraped for common first names (e.g., Daniel). In case that

these tweets did not contain the UFC keywords, they were removed from the dataset.

Another problem I faced with the fighter called Israel Adesanya is that his first name happens to be identical to the name of the state of Israel. This led to acquisition of political tweets that had to be filtered using keywords which I found to appear in these tweets.

89 of the scraped tweets — be it caused by *snscrape* or by Twitter itself — were written in Japanese. Those were removed from the dataset.

One more case I would like to point out happened with the fighter called Daniel Cormier. His nickname says "DC". When I added his surname into fight records (as in Code listing 1), I did not account for the fact that there may be tweets concerning either Washington DC (capital city of USA), or DC Universe (comics universe). I worked on removal of these tweets from the Daniel Cormier tweets dataset.

Lastly, for each dataset I would construct a wordcloud (plot of most frequents words) in order to ensure that there are no high levels of noise which would simply overshadow our UFC tweets. I observed a lot of noise in tweets scraped for the fight between Sean O'Malley and Petr Yan. Combination of the surname of Petr Yan and patterns like "KO *" led to scraping of a lot of irrelevant tweets written in Tagalog language which is spoken in Philippines. I found out that it is caused by the fact that phrases like "ko yan" are frequently used in that language. Effort was put into filtering these tweets out, while preserving the relevant tweets. The way these tweets looked before and after results can be demonstrated in Figure 5.1 and Figure 5.2 respectively. These figures show the most frequents word in tweets scraped for Petr Yan.

To conclude, it is a complex task to maximize both relevance of downloaded tweets as well as their cardinality. Chasing both at the same time leads to various complications and varying degrees of data noise, as discussed above.

Figure 5.1 Wordcloud of Petr Yan tweets before Tagalog tweets removal

Figure 5.2 Wordcloud of Petr Yan tweets after Tagalog tweets removal

Chapter 6

Statistics-based Approach

This chapter is dedicated to construction of statistics for UFC forecasting. First of all, statistics are gathered from various sources. Afterwards, the process of cleaning data is outlined. Subsequently, the way features are constructed is discussed.

6.1 Data sources

The primary source of statistical data is UFCStats[1]. Around 10 fights happen during each event. For each fight, UFCStats statistics include features that describe various aspects of the fight, such as name of the winner, time that fight lasted, method of winning. Most importantly, it gives us statistics of fighters' performance during a bout. These statistics and other UFCStats data will be discussed in following paragraphs.

Table 6.1 describes overall fight statistics. In following description I will outline some of those that are not obvious:

KD — knockdown is scored when fighter knocks his opponent to the ground with a strike,

TD — takedown happens when fighter wrestles his opponent to the ground,

SUB. ATT — submission attempt is scored when one tries to force his opponent to tap and lose (for example, by choking his opponent, or doing an armbar),

REV. — reversal occurs when fighter which is dominated during wrestling gets into a dominant position,

CTRL. — control time is counted as time that fighter dominates his opponent in wrestling position.

FIGHTER	KD	SIG. STR.	SIG. STR. %	TOTAL STR.	TD	TD %	SUB. ATT	REV.	CTRL
Dustin Poirier	0	97 of 216	44%	122 of 243	3 of 3	100%	0	1	3:57
Jim Miller	0	71 of 135	52%	83 of 150	1 of 6	16%	3	0	1:22

Table 6.1 Totals statistics from UFCStats for Dustin Poirier vs Jim Miller bout

[1] http://www.ufcstats.com/statistics/events/completed

Table 6.2 illustrates a detailed description of significant strikes which were landed during the fight (e.g., "79 of 186" in `HEAD` column indicates that 186 times Dustin Poirier attempted to hit his opponent's head with a significant strike, but only 79 strikes landed).

FIGHTER	SIG. STR	SIG. STR. %	HEAD	BODY	LEG	DISTANCE	CLINCH	GROUND
Dustin Poirier	97 of 216	44%	79 of 186	10 of 14	8 of 16	80 of 185	13 of 21	4 of 10
Jim Miller	71 of 135	52%	36 of 98	20 of 20	15 of 17	56 of 118	13 of 15	2 of 2

Table 6.2 Significant strikes statistics from UFCStats for Dustin Poirier vs Jim Miller bout

UFCStats contain some characteristics of the fighters themselves. An example of this information is captured in Table 6.3.

FIGHTER	HEIGHT	WEIGHT	REACH	STANCE	DOB	RECORD
Jim Miller	5' 8"	155 lbs.	71"	Southpaw	Aug 30, 1983	35-17-0

Table 6.3 Characteristics of Jim Miller from UFCStats

Finally, UFCStats also provide current average statistics of fighters, as shown in Table 6.4. Quoted description of these variables by UFCStats is given below:

SLpM — Significant Strikes Landed per Minute,

Str. Acc. — Significant Striking Accuracy,

SApM — Significant Strikes Absorbed per Minute,

Str. Def — Significant Strike Defence (the % of opponents strikes that did not land),

TD Avg. — Average Takedowns Landed per 15 minutes,

TD Acc. — Takedown Accuracy,

TD Def. — Takedown Defense (the % of opponents TD attempts that did not land),

Sub. Avg. — Average Submissions Attempted per 15 minutes.

FIGHTER	SLpM	Str. Acc.	SApM	Str. Def	TD Avg.	TD Acc.	TD Def.	Sub. Avg.
Jim Miller	2.85	41%	3.08	58%	1.56	43%	48%	1.8

Table 6.4 Average statistics of Jim Miller from UFCStats

Information about fighters that is lacking on UFCStats website is country of origin of the fighters. However, this information is present on a different UFC page[2].

6.2 Data acquisition

In order to scrape the data discussed in the previous section, code was written in *Jupyter Notebook* [59] and these tools were used: *Python* and its library *requests*, *pandas* [58], *BeautifulSoup4* [60].

[2] https://www.ufc.com/athletes/all

6.2.1 Fights

Firstly, I gathered information about events' locations and dates. Subsequently, for each event I extracted URLs of the fights which were held during the event.

Fight statistics and other attributes provided on UFCStats website were obtained using fight URLs from the previous step. Total fight statistics were scraped (see Table 6.1) along with the detailed descriptions of significant strikes landed during the fight (see Table 6.2). Furthermore, I scraped fighters' names, winner's name, amount of rounds that fight lasted, time at which last round of the fight was finished, time format of the fight, name of the referee. In total, 7060 fight records were gathered.

6.2.2 Fighters

Characteristics of fighters provided on UFCStats were obtained (illustrated in 6.3). Among these were attributes describing height in feet and inches, weight in pounds, reach in inches, stance of the fighter, date of birth.

In contrast to "fight-centric" approach of McQuaide [42] (discussed in Chapter 4), I wanted to see whether country of origin attribute could provide improvement to accuracy of ML models. However, names of the fighters' countries of origin are mpt available on UFCStats website. Nevertheless, another UFC page[3] offered this information. Names of the homelands of fighters were acquired from that website.

Once data about fighters was scraped, I merged the dataset with the dataset containing fight statistics.

6.3 Data preprocessing

Tools which were utilized during preprocessing are *Python*, *pandas* [58] and *numpy* [61]. All the code was run in *Jupyter Notebook* [59].

As illustrated in Table 6.1, on UFCStats for each attribute information about fighters is provided for both fighters in one cell. For the sake of preparing them for a ML model, they needed to be separated. Similarly to Martinez-Ríos [44] (Chapter 4), I split fighters' columns into red and blue[4]. Thus, features of fighters that fought in the red corner received prefix `Red_`, while attributes of fighters in the blue corner were given prefix `Blue_`.

Similarly to Turgut [45] (as described in Chapter 4), I dealt with missing values in form of "`--`" and "`---`". In the same manner as Turgut, I replaced these values with "NaN" value.

Attributes describing weights of fighters were obtained in pounds. I decided to convert the weights into kilograms. Heights which were initially provided in feet and inches were converted to centimeters.

Some fighters did not have `Reach` provided. I imputed these values using linear regression and columns `Height` and `Weight`, as linear regression with `Height` alone showed slightly higher Root Mean Square Error (RMSE).

21 fights of the acquired fights had missing pretty much all of the statistics. I decided to drop them, and size of the dataset became 7039.

[3] https://www.ufc.com/athletes/all

[4] There are two corners in UFC. The first one is red, the second is blue.

During inspection of `Time_format` feature I observed value "No Time Limit". As occasional UFC viewer, I am used to time formats such as 3 or 5 rounds. This was new to me, and I decided do dig deeper into UFC rules and found an article [62] which claims that in 2001 UFC adopted new rules. Further research confirmed that these rules were adopted precisely on November 17, 2000 [63], [64]. To differentiate between the fights that happened prior to the adaption of the unified rules, I created feature `After_regulation` and marked those fights that happened after rules were adopted with "True" value. For fights which happened after adoption of new rules, there were only 4 unique values in `Time_format`. Prior to adaption of new rules, up to 18 time formats had been used.

35 fighters had missing date of birth. I decided to simply drop fight records with these fighters during modeling phase.

Two fighters in the dataset share the same name "Bruno Silva". I decided to change their names to "Bruno Silva (Jul_DOB)" and "Bruno Silva (Mar_DOB)", depending on whether the month of birth.

Attributes which were given in form of "23 of 53" (e.g., `Red_Head`, `Red_Body`) were separated into two columns, as demonstrated in Table 6.5. Thus, new `Red_Ground` would have value "23", and `Red_Ground att` would have value 50, where the former indicates 23 landed strikes on the ground by red fighter, while the latter indicates 50 attempts to do so.

Red_Ground (before)	Red_Ground	Red_Ground att
23 of 50	23	50

Table 6.5 Example of splitting landed and attempted actions

6.4 Feature engineering

In following subsections I will outline some of the features that were constructed.

6.4.1 Total time

Firstly, I created feature called `Total_time` using attributes `Rounds` (number of rounds that fight lasted), `Time_format` (format of the fight, indicating number of rounds and length of each round) and `Last_round_time` (time when last round ended). This is summarized in Table 6.6.

Time_format	Rounds	Last_round_time	Total_time
3 Rnd (5-5-5)	2	4:11	9:11

Table 6.6 Total time calculation

Finally, feature `Total_time_sec` was constructed by simply multiplying number of minutes in `Total_time` by 60 and adding number of seconds from `Total_time`. Later, `Total_time_sec` will be utilized in Section 6.4.7 for finding average performance statistics of athletes.

6.4.2 Age

As McQuaide (see Figure 4.1) found age to be one of the most important variables for her model, I construct this feature as well. For each fight record, `Age` was computed using difference between the date when fight took place and date of birth of the fighter. Table 6.7 captures this process.

Date	Red_DOB	Blue_DOB	Red_Age	Blue_Age
March 25, 2023	Oct 17, 1981	Nov 11, 1989	41.0	33.0

Table 6.7 Age calculation

6.4.3 Fight abroad

In Figure 6.1 we can see 20 most represented nations in UFC along with the absolute numbers (e.g., historically there have been 1352 fighters from USA). If athlete fights abroad, it can potentially have a negative effect on his performance and, thus, on fight outcome. Historically, UFC fights have been carried out in 26 countries. I decided to test whether an attribute telling if he fights abroad could improve accuracy of modeling and created feature called `Fights_abroad`. So, in case that fight takes place in USA and fighter in the red corner is from, say, Poland, then `Red_Fights_abroad` would have "True" value. If athlete fights in his homeland, it would be "False".

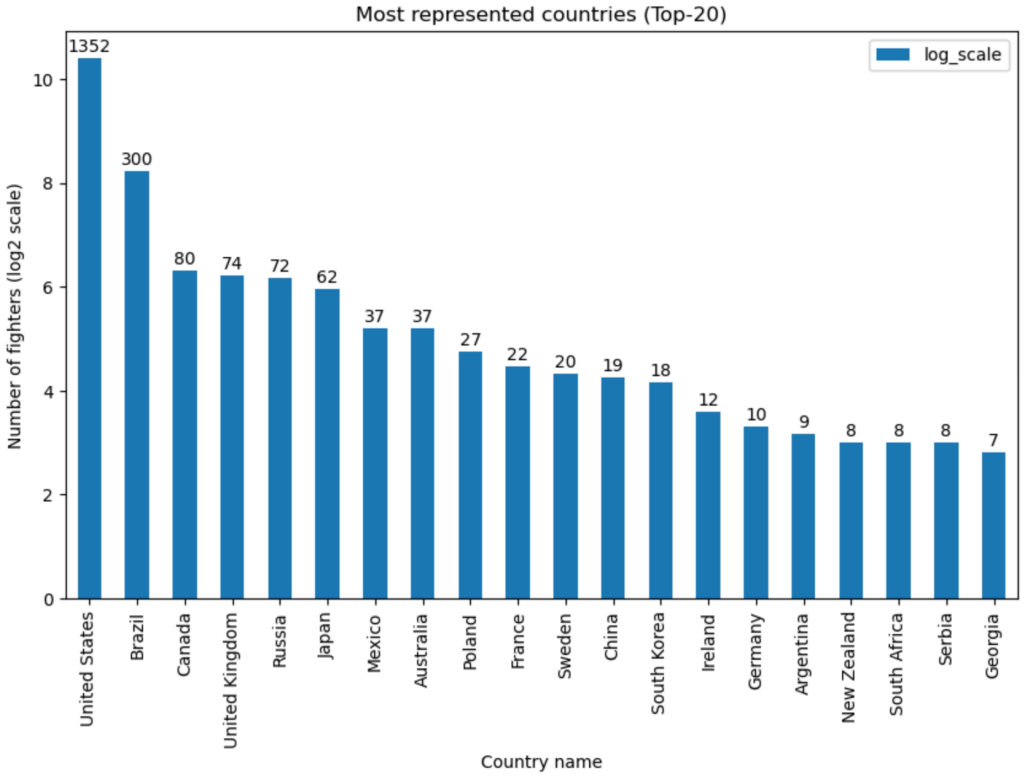

Figure 6.1 Most represented countries in UFC

6.4.4 Total record

Total historical record of the fighter up to the date of the fight in terms of wins, losses and draws was calculated by iterating the dataset backwards and tracking change in fighter's these attributes depending on whether he loses, wins or scores a draw. For blue fighter, these values were saved in columns `Blue_Wins`, `Blue_Losses`, `Blue_Draws`.

6.4.5 Streak record

Current streak record of the fighter up to the date of the fight was calculated in terms of losses and draws by iterating the dataset in reversed order and placing current streaks to each fight datapoint. These values are stored in features `Red_Win_streak`, `Red_Lose_streak` for fighter which fights in red corner. Former says how many wins he has in a row, while latter tells the same in terms of losses. Once fighter wins, his win streak is upped by 1 and his loss streak becomes 0 (in case that it is already not 0). Once he loses, same thing happens but the other way out.

6.4.6 Elo

Proposed by Arpad Elo [65], Elo is a player rating system which is mostly known by its application in chess. Nowadays, it is used in other areas as well. For instance, it is utilized in multi-player video games [66]. Moreover, some works have demonstrated application of Elo in football [67] and tennis [68]. I decided to conduct an experiment with this feature too and observe whether it can improve quality of ML-based UFC forecasting.

I set initial rating of fighters to 1000. Thus, every beginner starts with rating $R = 1000$. For construction of `Red_Elo` and `Blue_Elo`, following formulas [69] were used:

$$E_{\mathrm{A}} = \frac{1}{1 + 10^{(R_{\mathrm{B}} - R_{\mathrm{A}})/400}}, \tag{6.1}$$

$$R'_{\mathrm{A}} = R_{\mathrm{A}} + K \cdot (S_{\mathrm{A}} - E_{\mathrm{A}}), \tag{6.2}$$

where:

- E_{A} is expected score of fighter A for the fight between him and fighter B (between 0 and 1),
- S_{A} is actual score of fighter A which depends on the outcome (0 for loss, 0.5 for draw, 1 for win),
- R_{A} and R_{B} are current ratings of fighter A and B respectively,
- R'_{A} is A's updated Elo, which is modified after the fight,
- K is so-called K-factor (set to 32 by default).

If "K-value is too low, the sensitivity will be minimal, and the system will not respond quickly enough to changes in a player's actual level of performance" [69]. Thus, I decided to create `Red_Weighted_Elo` and `Blue_Weighted_Elo`, for which K-factor is multiplied by 1.5

in case that winner of the fight uses either submission, or a knockout (dominant methods) to win the fight. Weighted K-factor is calculated as follows:

$$R'_{\mathrm{A}} = R_{\mathrm{A}} + 1.5 \cdot K \cdot (S_{\mathrm{A}} - E_{\mathrm{A}}). \tag{6.3}$$

Lastly — with intention to accelerate Elo growth of those fighters that use knockouts and submissions to finish their opponents with respect to the rarity of these finishing methods at the moment of the fight — I constructed `Red_Rarified_Elo` and `Blue_Rarified_Elo`, where I scale K-factor as follows:

$$R'_{\mathrm{A}} = R_{\mathrm{A}} + \alpha(i) \cdot K \cdot (S_{\mathrm{A}} - E_{\mathrm{A}}), \tag{6.4}$$

with α calculated the following way:

$$\alpha(i) := \max\left\{1.5,\ \frac{\sum \text{fights ended via } \mathcal{M} \setminus \mu \text{ prior to } i\text{-th fight}}{\sum \text{fights ended via } \mu \text{ prior to } i\text{-th fight}}\right\}, \tag{6.5}$$

where $\mu \in \mathcal{M}$, and elements of $\mathcal{M} = \{$ “KO”, “Submission”, “Other” $\}$ are knockout, submission, and other finishing methods respectively (e.g., if there are currently 100 wins in UFC, of which 80 by decision, 18 by knockout and 2 by submission, then if next fighter wins by submission he gets his K-factor multiplied by $\alpha(101) = \frac{18+80}{2}$, which is bigger than 1.5 and thus chosen as α value).

During modeling phase, we will experiment and see which Elo feature is the most effective at improving predictive accuracy.

6.4.7 Average statistics

All we know about fighters prior to the fight is how they performed in their previous fights. Simply averaging past statistics by summing absolute numbers and dividing them by the number of fights performed up to this fight is rather ineffective, as it assumes that all fights last same amount of time, which is not always true.

With that in mind, similarly to Turgut [45] (as discussed in Chapter 4) — and the way averaging is done on UFCStats (see Table 6.4) — for each fight I calculate average statistics of both fighters’ prior to the fight by measuring their performance per minute (for frequent events, such as total strikes) or 15 minutes (for rarer events, such as knockdowns).

Likewise Turgut, in case that some fight is the first one for an athlete, his average statistics will be simply set to 0, as we know nothing about him prior to his very first bout. We will mark these fights with `Debut` column having “True” in case that any of the two fighters is a novice fighting for the first time. Later, during modeling phase, we will simply drop the debuts. Nevertheless, calculation of performance averages for fights which are at least second for both fighters is outlined in following text.

6.4.7.1 Per minute

For events which occur a lot of times during the fight (e.g., total strikes, body strikes) per minute (pM) averaging is applied for each fighter before his i-th fight as follows:

$$\omega_{\text{pM}}(i) := \frac{\sum \omega_F \text{ landed prior to } i\text{-th fight}}{\sum \text{seconds spent in octagon prior to } i\text{-th fight}} \cdot 60, \tag{6.6}$$

where $\omega_F \in \mathcal{F}$, and $\mathcal{F} = \{$`Total str.`, `Sig. str.`, `Head`, `Body`, `Leg`, `Distance`, `Clinch`, `Ground`$\}$ are attributes describing actions which happen frequently in octagon during the fight.

In Equation (6.6) we first calculate average statistics per second (using column from 6.4.1) and then scale it to per minute. For the sake of clarity, an example can be seen in Table 6.8.

i	Total_time_sec	ω_F	$\omega_{\text{pM}}(i)$
1	90	30	0
2	110	70	$(30)/(90) \cdot 60 = 20$
3	63	27	$(30+70)/(90+110) \cdot 60 = 30$

Table 6.8 Averaging of frequent performance statistics

6.4.7.2 Per 15 minutes

Pre-fight averages of features describing events which happen infrequently throughout the fight were calculated per 15 minutes (p15M) for each fighter before his i-th fight as in Equation (6.7) that follows:

$$\omega_{\text{p15M}}(i) := \frac{\sum \omega_R \text{ landed prior to } i\text{-th fight}}{\sum \text{seconds spent in octagon prior to } i\text{-th fight}} \cdot 900, \tag{6.7}$$

where $\omega_R \in \mathcal{R}$, and $\mathcal{R} = \{$`KD`, `Td`, `Sub. att`, `Rev.`$\}$ are attributes describing actions which happen rarely in octagon during the fight.

In Equation (6.7) we first calculate average statistics per second (using column from Subsection 6.4.1) and then scale it to gain per 15 minutes averages. To make things more comprehensible, an instance of this problem can be seen in Table 6.9.

i	Total_time_sec	ω_R	$\omega_{\text{p15M}}(i)$
1	90	3	0
2	110	1	$(3)/(90) \cdot 900 = 30$
3	63	2	$(3+1)/(90+110) \cdot 900 = 18$

Table 6.9 Averaging of rare performance statistics

Unlike features in set $\mathcal{R}$ which are given in form of frequencies, column `Ctrl` describes time that fighter controlled his opponent while wrestling. Firstly, I converted `Ctrl` features for both fighters to `Ctrl_sec`, which is a column that indicates the amount of seconds fighters were controlling their opponents in fights. Then, I calculated average per 15 minutes as described in Equation (6.7), with $\sum \omega_R$ now standing for total number of seconds fighter

had been controlling his opponents prior to the fight we calculate this statistic for. This led to creation of features `Red_Ctrl_p15M` and `Blue_Ctrl_p15M`.

6.4.7.3 Accuracies

Average accuracies were calculated for columns which initially contained values like "97 of 216" for `SIG. STR.` in Table 6.1 (which then results in "44%" for `SIG. STR. %`) and next were split into landed and attempted strikes, as in Table 6.5. Computation of average accuracies for each fighter before his i-th fight was done as follows:

$$\omega_{\text{acc}}(i) := \frac{\sum \omega_A \text{ landed prior to } i\text{-th fight}}{\sum \omega_A \text{ attempted prior to } i\text{-th fight}}, \tag{6.8}$$

where $\omega_{\text{A}} \in \mathcal{A}$, and $\mathcal{A} = \{$`Total str.`, `Td`, `Sig. str.`, `Head`, `Body`, `Leg`, `Distance`, `Clinch`, `Ground`$\}$ are attributes describing actions which are decomposed in dataset in terms of landed (e.g., `Red_Total str.`) and attempted strikes (e.g., `Red_Total str. att`), and thus have accuracies calculated for them (e.g., `Red_Total str._Avg_Acc`).

6.5 Fighting stance

As shown in Figure 6.2, there have historically been 5 different fighting stances in UFC. For instance, "Orthodox" stance means that fighter has his right leg as his leading leg when he stands, as well as his right hand. "Southpaw" means the opposite, with right hand and right foot being in front. Some fighters did not have their stances provided, and so I replaced these values with "Unknown". We can expect that fighter which has "Orthodox" stance can be surprised by the athletes which have a different stance, as they are statistically rarer, and "Orthodox" fighters may have less experience fighting them.

Applying One Hot Encoding (OHE) to `Red_Stance` and `Blue_Stance` features which store these values would lead to creating of 12 columns. Hence, these columns were removed and 2 features `Red_Orthodox` and `Blue_Orthodox` were created, as "Orthodox" is the most frequent value. Their values are "True" if fighter has "Orthodox" stance, and "False" otherwise.

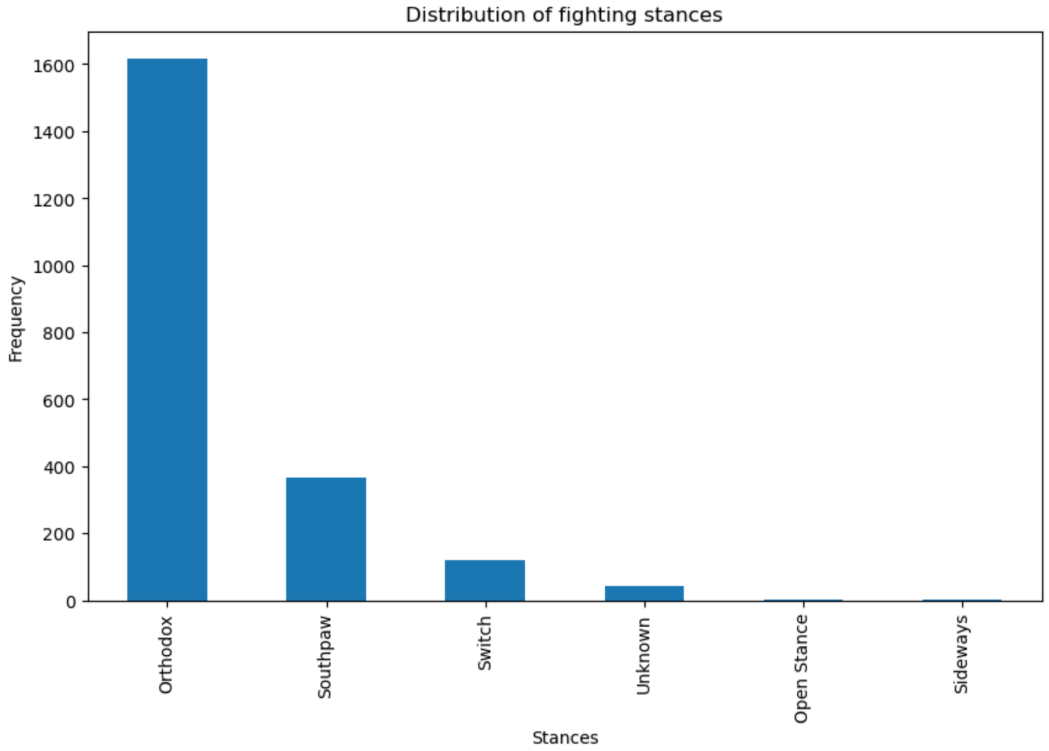

Figure 6.2 Frequencies of UFC fighters stances

Chapter 7

Experiments

7.1 Evaluation metrics

Performance of Machine Learning models for binary classification problems can be measured in various ways. Before we move further, the basic concepts for evaluating predictive efficacy of ML models need to be outlined:

TP — number of True Positives (predicted label is counted as TP if datapoint class is positive, and is predicted as such),

FP — number of False Positives (predicted label is counted as FP if datapoint class is negative, but is falsely predicted as positive),

TN — number of True Negatives (predicted label is counted as TN if datapoint class is negative, and is predicted as such),

FN — number of False Positives (predicted label is counted as TP if datapoint class is negative, but is falsely predicted as negative).

Commonly used measurements are built upon the fundamentals listed above. Metrics which are used in this work are discussed in following subsections.

7.1.1 Confusion matrix

For binary classification problems, confusion matrix can be defined as in Table 7.1, where:

- N stands for total number of datapoints, that is, $N = TP + FP + TN + FN$,
- $FN + TN$ stands for total number of negative class predictions,
- $TP + FP$ stands for total number of positive class predictions,
- $TN + FP$ stands for total number of negative class datapoints,
- $FN + TP$ stands for total number of positive class datapoints.

	Total	$FN + TN$	$TP + FP$	N
True value	Negative	TN	FP	$TN + FP$
	Positive	FN	TP	$FN + TP$
		Negative	Positive	Total
		Predicted value		

Table 7.1 Confusion matrix

For instance, in Figure 7.1 we can see a case with a dataset of 8 datapoints, for which 3 predictions are TN, 2 FP, 2 FN and 1 TP.

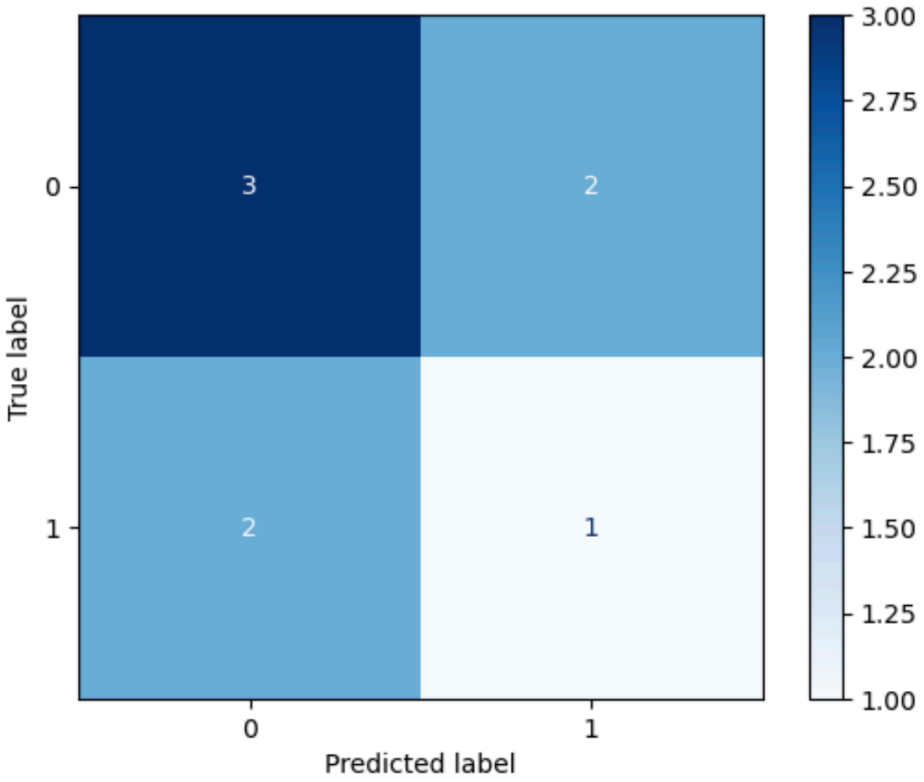

Figure 7.1 Confusion matrix

7.1.2 Accuracy

Accuracy is computed as follows:

$$\frac{TP + TN}{TP + FP + TN + FN}. \tag{7.1}$$

It describes an overall performance of a model with respect to correct predictions of each class.

7.1.3 Precision

Precision is computed as follows:

$$\frac{TP}{TP + FP}. \tag{7.2}$$

It stands for a ratio between relevant positive predictions and all positive predictions.

7.1.4 Recall

Recall, also referred to as sensitivity, is computed as follows:

$$\frac{TP}{TP + FN}. \tag{7.3}$$

It tells how well a model is at classifying datapoints with positive label as positive, without mispredicting them as negative. Intuitively, this metric can be thought of as percentage of positive classes a model preserves.

7.1.5 F1-score

F1-score is computed as follows:

$$2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \tag{7.4}$$

It takes into account both precision and recall in compact way.

7.2 Statistics-based approach modeling

After removing fight records with `Debut` being "True" (as discussed in Subsection 6.4.7) and fights which resulted in a draw, we end up with our target variable distribution as described in Table 7.2.

Red winners	Red winners %	Blue winners	Blue winners %
3105	62.14%	1892	37.86%

Table 7.2 Modeling dataset

Afterwards, in the manner of Martinez-Ríos [44], Turgut [45], dataset is split into 80% training data (see Table 7.3) and 20% testing data (see Table 7.4).

Red winners	Red winners %	Blue winners	Blue winners %
2513	62.87%	1484	37.13%

Table 7.3 Training dataset

Red winners	Red winners %	Blue winners	Blue winners %
592	59.20%	408	40.80%

Table 7.4 Testing dataset

7.2.1 Multiple baseline models

Target variable is represented as:

- 1 if red corner fighter won the fight,
- 0 if blue corner fighter won the fight.

Elo variables (see Subsection 6.4.6) are removed at this point, in order to first determine the best baseline model trained upon the primary features. As demonstrated in Table 7.5, experiments are conducted with a range of baseline models using the *scikit-learn* [70] implementations and CatBoost model [71]. Similarly to Martinez-Ríos, accuracies (see Table 4.2) of the models are measured along with F1-scores (see Table 4.3) for both classes.

Model	Accuracy	F1-score (Red)	F1-score (Blue)
Decision Tree	51.9%	60%	39%
Random Forests	61.3%	**73%**	32%
KNN	55.5%	66%	35%
AdaBoost	60.9%	71%	**42%**
CatBoost	**62.2%**	73%	38%

Table 7.5 Baseline models performance

These two models proceed further: AdaBoost and CatBoost. The former has the best F1-score for the minority class, second best F1-score for majority class and third best accuracy among other baseline models. The latter has the best accuracy, best F1-score for the majority class, and third best F1-score for the minority class.

7.2.2 Adding Elo

7.2.2.1 AdaBoost

As demonstrated in Table 7.6, in comparison to the baseline model AdaBoost enriched with weighted Elo variables for both fighters achieves an improvement of 1.9% in terms of accuracy, and improvement of 1% in terms of F1-score for the majority class, and an improvement of 1% for the minority class of blue corner fighters. Twenty-five features which AdaBoost identified as the most important are provided in Figure 7.2. Confusion matrix (as discussed in 7.1.1) of the model is provided in Figure 7.3. In terms of accuracy, with this model we outperform solutions by Hitkul et al. [41] and Turgut [45], which were outlined in Chapter 4 with the results illustrated in Table 4.1 and Table 4.4 respectively. Although we outperformed Random Forests model of Martinez-Ríos in terms of F1-score for the minority class of blue fighters (see his F1-score results in Table 4.3), we have not managed to outscore the accuracy he achieves (as in Table 4.2).

Elo type	Accuracy	F1-score (Red)	F1-score (Blue)
No Elo	60.9%	71%	42%
Basic Elo	60.9%	71%	42%
Weighted Elo	**62.8%**	**72%**	**43%**
Rarified Elo	61.5%	71%	42%

Table 7.6 AdaBoost performance with Elo variables

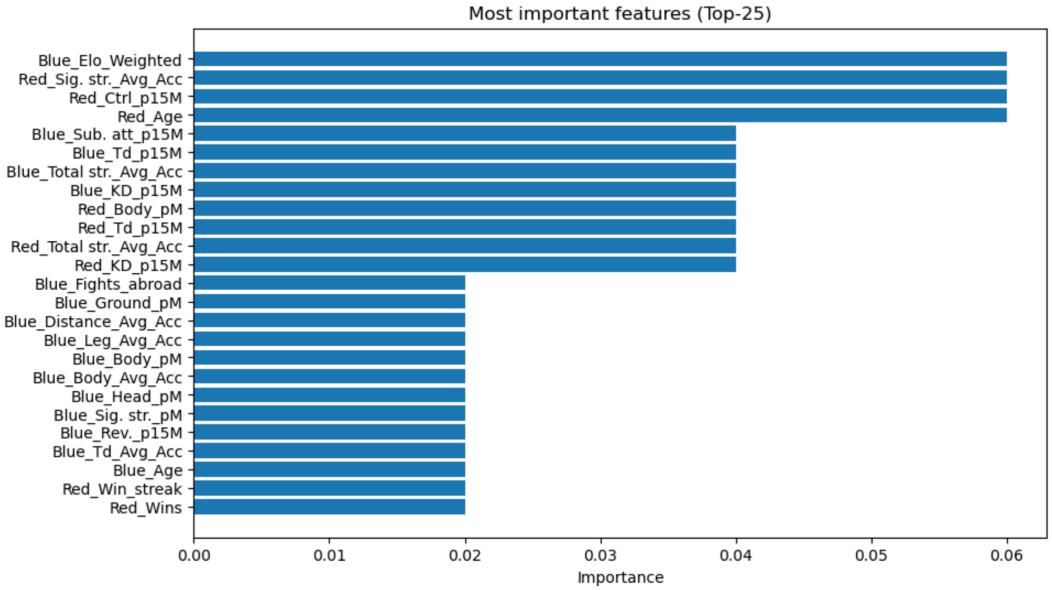

Figure 7.2 Feature importances per AdaBoost with weighted Elo

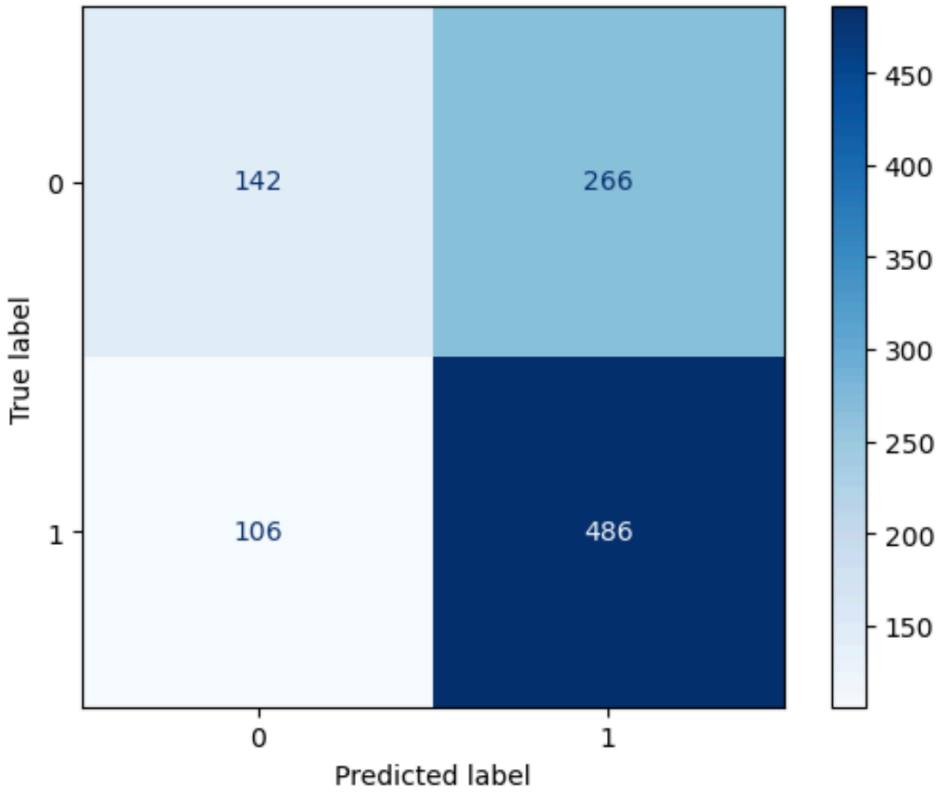

Figure 7.3 Confusion matrix of AdaBoost with weighted Elo

7.2.2.2 CatBoost

In Table 7.6 we can see how as opposed to the baseline model performance CatBoost which incorporates basic Elo variable achieves gain of 0.3% in terms of accuracy and an improvement of 1% for the minority class of blue corner fighters. Twenty-five features which identified by CatBoost as the most important are provided in Figure 7.4. Confusion matrix (as discussed in 7.1.1) of the model is provided in Figure 7.5.

Elo type	Accuracy	F1-score (Red)	F1-score (Blue)
No Elo	62.2%	73%	38%
Basic Elo	**62.5%**	73%	**39%**
Weighted Elo	62.0%	73%	38%
Rarified Elo	62.5%	73%	38%

Table 7.7 CatBoost performance with Elo variables

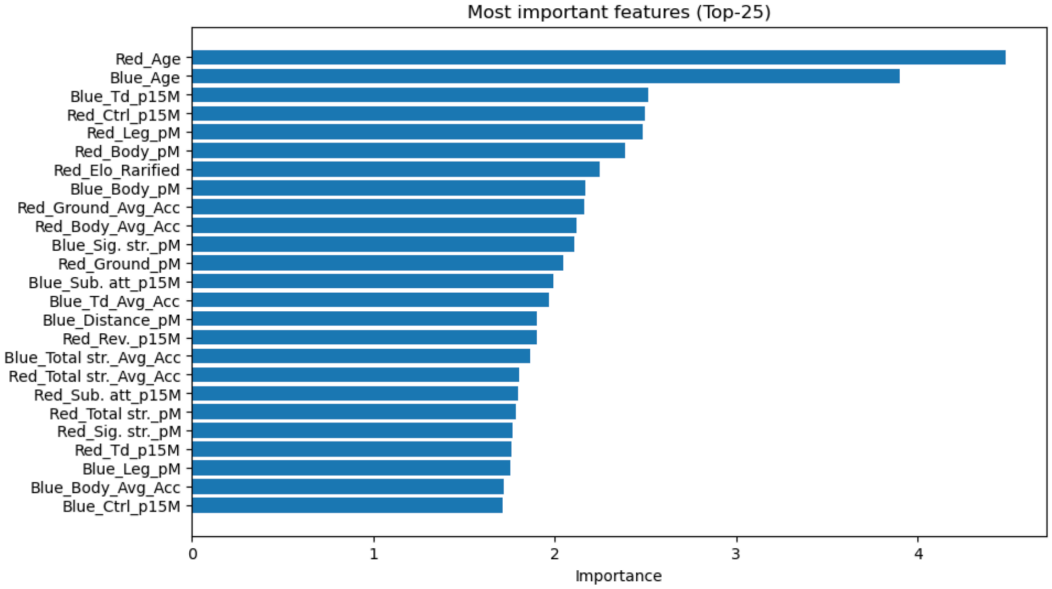

Figure 7.4 Feature importances per CatBoost with rarified Elo

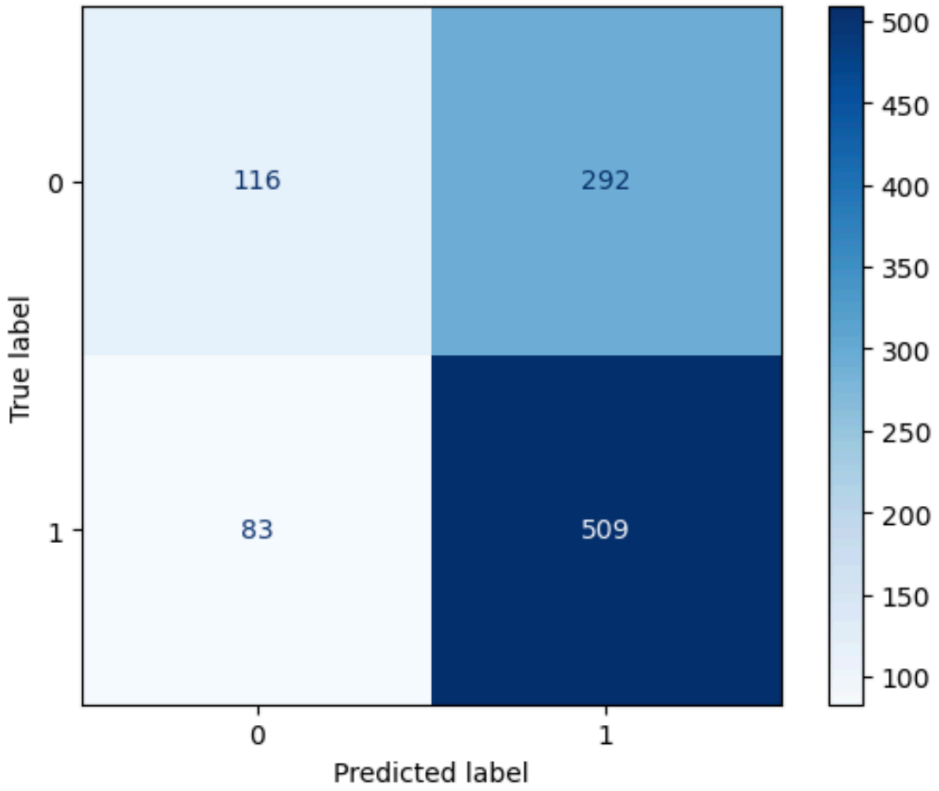

Figure 7.5 Confusion matrix of CatBoost with rarified Elo

7.3 Sentiment-based approach evaluation

7.3.1 Tweets distribution

In Figure 7.6 we can see how tweets we work with are distributed. These findings correspond to what was stated in Subsection 5.2.3 about the fact that only around 50 negative patterns were identified in tweets and labeled as such.

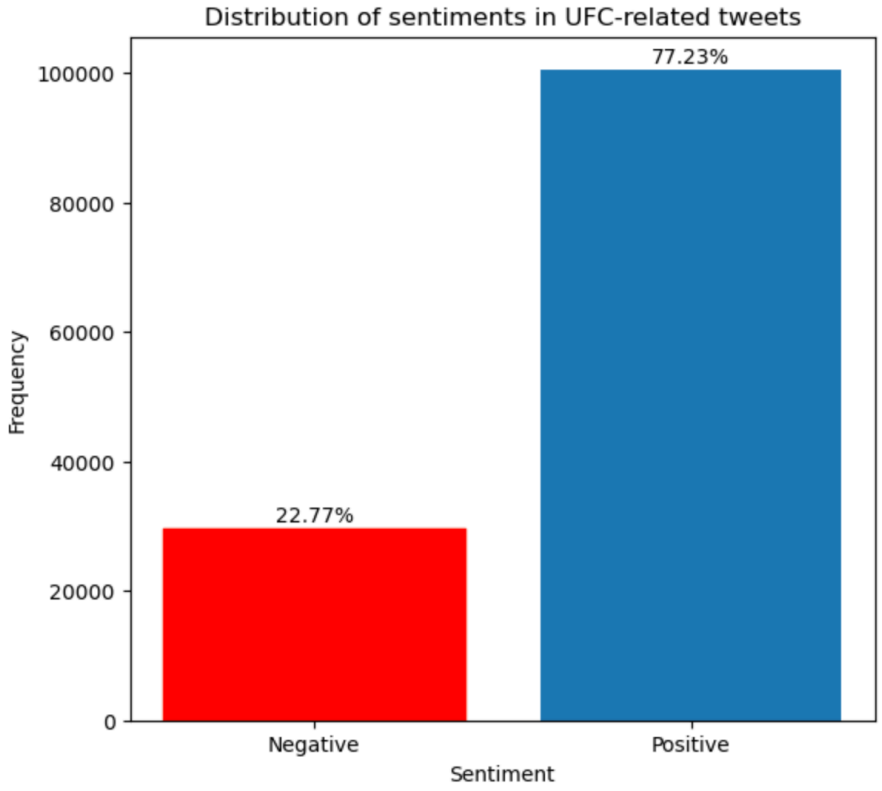

Figure 7.6 Distribution of sentiment in acquired UFC tweets

7.3.2 Winner evaluation metrics

As discussed in 5.2.2, total of 23 fights had tweets which were published prior to the fights scraped for them. In order to evaluate how good they serve as predictors of fight outcomes, I came up with the metrics which define who should be considered a winner based on his tweets in comparison to his opponent tweets. These metrics are discussed in following paragraphs.

Percentage of positive tweets:

$$\frac{positive\_tweets}{positive\_tweets + negative\_tweets} \cdot 100.0, \tag{7.5}$$

if fighter has higher percentage of positive tweets than opponent, then he is considered as predicted winner.

This metric is based upon count of total number of positive tweets:

$$positive\_tweets, \tag{7.6}$$

if fighter has higher total number of positive tweets than opponent, then he is considered as predicted winner.

The following metric utilizes total number of negative tweets:

$$negative\_tweets, \tag{7.7}$$

if fighter has lower amount of negative tweets than opponent, then he is considered as predicted winner, as there is less negative behavior towards him.

The last metric which follows uses positive minus negative tweets difference:

$$positive\_tweets - negative\_tweets, \tag{7.8}$$

if fighter has higher difference between amount of positive tweets and negative tweets than his opponent, then he is considered as predicted winner.

7.3.3 Results

Using metrics defined in Subsection 7.3.2, experiments were conducted. Results are demonstrated in Table 7.8. It is important to note that despite the imbalance between negative

Metric type	Formula	Accuracy
Positive tweets %	(7.5)	73.91%
Positive tweets count	(7.6)	39.13%
Negative tweets count	(7.7)	65.21%
Positive minus negative count	(7.8)	47.82%

Table 7.8 Predictive accuracy of Twitter sentiments

and positive sentiment distribution (as illustrated in Table 7.6), in Table 7.8 we can clearly see that tweets representing negative sentiments play a critical role in prediction, as positive tweets in stand-alone performance show the worst accuracy. The best performance is achieved with the metric described in Equation 7.5.

7.4 Combining the two

In following subsection, we will demonstrate some experiments with AdaBoost classifier with weighted Elo, ran on the 23 datapoints for which both sentiments and statistics were acquired. In one run we leave out column which states who the winner is per sentiments, while in seconds run we integrate this column. Experiments were run using k-folds (e.g., if k is 3, then we split our dataset into 3 equal parts, and during 3 runs model is trained on 2 of these parts, while the third part which is left-out serves as testing dataset). See Table 7.9.

With sentiment	k	Average accuracy
Yes	2	34.09%
No	2	52.27%
Yes	3	52.38%
No	3	42.85%

Table 7.9 AdaBoost with and without sentiment. Average accuracy

Since we have only 23 datapoints, which is a very small dataset for a Machine Learning model, it is hard to extrapolate the achieved results. As demonstrated in Table 7.8, by themselves tweets reach accuracy of 73.91%. Which suggests that future research may focus on getting Twitter sentiment data for more fights in order to see how significant the overall impact is for Machine Learning models.

Chapter 8

Conclusion

The primary objectives of this work were to study Sentiment Analysis theory and methods to forecast future fight outcomes based on the past statistics and propose approaches for UFC forecasting incorporating acquired knowledge. An extensive research was carried out for both of these aspects. Hence, this work can serve as a reference point for those who will conduct experiments in similar field.

After thorough exploration of online sources of opinionated data for UFC on a range of social media platforms, a lexicon-based approach using Twitter platform was proposed. Further, the proposed approach was evaluated and shown as effective on the studied sample of 23 fights, on which accuracy of 73.91% was achieved, with observation that negative sentiment represented in tweets plays the critical part for sentiment-based forecasting. Thus, the approach is potentially promising for the whole population of UFC fights.

Subsequently, Machine Learning models were built using past-statistics, and showed performance corresponding to the models discussed during the literature review. Experiments were carried out with features incorporating Elo. These variables improved performance of the ML models, with the best one scoring accuracy of 62.8%. This is suggestive of the fact that Elo can improve results of fight outcome predictions, and could be further explored in future works.

To conclude, this work demonstrates a novel approach for UFC forecasting, and serves as a proof that user opinions left on online social media platforms can be viewed predictors of UFC fight outcomes. Following works in this field may focus specifically on Twitter in similar manner. If future research shows that many past UFC fights can be covered in terms of sentiments, Machine Learning can be models fine-tuned with these sentiments.

Bibliography

1. DOMINGOS, Pedro. A Few Useful Things to Know About Machine Learning. *Commun. ACM* [online]. 2012, vol. 55, no. 10, pp. 78–87. Available from DOI: 10.1145/2347736.2347755.
2. MITCHELL, Tom. *Machine Learning*. New York, NY, USA: McGraw-Hill, 1997. ISBN 0070428077.
3. GÉRON, Aurélien. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*. Second Edition. Sebastopol, CA, USA: O'Reilly Media, Inc., 2019. ISBN 978-1-492-03264-9.
4. MUHAMMAD, Iqbal; YAN, Zhu. SUPERVISED MACHINE LEARNING APPROACHES: A SURVEY. *ICTACT Journal on Soft Computing* [online]. 2015, vol. 5, no. 3. Available from DOI: 10.21917/ijsc.2015.0133.
5. ALI, Jehad; KHAN, Rehanullah; AHMAD, Nasir; MAQSOOD, Imran. Random forests and decision trees. *International Journal of Computer Science Issues (IJCSI)* [online]. 2012, vol. 9, no. 5, p. 272. Available also from: https://www.uetpeshawar.edu.pk/TRP-G/Dr.Nasir-Ahmad-TRP/Journals/2012/Random%20Forests%20and%20Decision%20Trees.pdf.
6. BREIMAN, Leo. Random forests. *Machine Learning* [online]. 2001, vol. 45, pp. 5–32. Available also from: https://link.springer.com/article/10.1023/a:1010933404324.
7. NADKARNI, Prakash M; OHNO-MACHADO, Lucila; CHAPMAN, Wendy W. Natural language processing: an introduction. *Journal of the American Medical Informatics Association* [online]. 2011, vol. 18, no. 5, pp. 544–551. ISSN 1067-5027. Available from DOI: 10.1136/amiajnl-2011-000464.
8. KHURANA, Diksha; KOLI, Aditya; KHATTER, Kiran; SINGH, Sukhdev. Natural language processing: state of the art, current trends and challenges. *Multimedia Tools and Applications* [online]. 2022, vol. 82, no. 3, pp. 3713–3744. Available from DOI: 10.1007/s11042-022-13428-4.

9. BAID, Palak; GUPTA, Apoorva; CHAPLOT, Neelam. Sentiment analysis of movie reviews using machine learning techniques. *International Journal of Computer Applications* [online]. 2017, vol. 179, no. 7, pp. 45–49. Available from DOI: 10.5120/ijca2017916005.
10. YU, Liang-Chih; WU, Jheng-Long; CHANG, Pei-Chann; CHU, Hsuan-Shou. Using a contextual entropy model to expand emotion words and their intensity for the sentiment classification of stock market news. *Knowledge-Based Systems* [online]. 2013, vol. 41, pp. 89–97. ISSN 0950-7051. Available from DOI: https://doi.org/10.1016/j.knosys.2013.01.001.
11. BERMINGHAM, Adam; SMEATON, Alan F. On Using Twitter to Monitor Political Sentiment and Predict Election Results [online]. 2011. Available also from: https://doras.dcu.ie/16670/1/saaip2011.pdf.
12. MEDHAT, Walaa; HASSAN, Ahmed; KORASHY, Hoda. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal* [online]. 2014, vol. 5, no. 4, pp. 1093–1113. ISSN 2090-4479. Available from DOI: https://doi.org/10.1016/j.asej.2014.04.011.
13. LIU, Bing. Sentiment analysis: A multi-faceted problem. *IEEE Intelligent Systems* [online]. 2010, vol. 25, no. 3, pp. 76–80. Available also from: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=795d5908f0f409a7db7b62e3372dec316f5f3478.
14. LIU, Bing. Sentiment Analysis and Subjectivity. *Handbook of natural language processing* [online]. 2010, vol. 2, no. 2010, pp. 627–666. Available also from: https://www.cs.uic.edu/~liub/FBS/NLP-handbook-sentiment-analysis.pdf.
15. FELDMAN, Ronen. Techniques and Applications for Sentiment Analysis. *Commun. ACM*. 2013, vol. 56, no. 4, pp. 82–89. ISSN 0001-0782. Available from DOI: 10.1145/2436256.2436274.
16. JINDAL, Nitin; LIU, Bing. Identifying Comparative Sentences in Text Documents. In: *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Seattle, Washington, USA: Association for Computing Machinery, 2006, pp. 244–251. SIGIR '06. ISBN 1595933697. Available from DOI: 10.1145/1148170.1148215.
17. WILSON, Theresa; WIEBE, Janyce; HOFFMANN, Paul. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In: *Proceedings of human language technology conference and conference on empirical methods in natural language processing* [online]. 2005, pp. 347–354. Available also from: https://aclanthology.org/H05-1044.pdf.
18. BUYYA, Rajkumar; CALHEIROS, Rodrigo N; DASTJERDI, Amir Vahid. *Big data: principles and paradigms*. Cambridge, MA, USA: Morgan Kaufmann, 2016. ISBN 9780128053942.
19. HATZIVASSILOGLOU, Vasileios; MCKEOWN, Kathleen. Predicting the semantic orientation of adjectives. In: *35th annual meeting of the association for computational linguistics and 8th conference of the european chapter of the association for computational linguistics* [online]. 1997, pp. 174–181. Available also from: https://aclanthology.org/P97-1023.pdf.

20. VICENTE, Iñaki San; SARALEGI, Xabier. Polarity Lexicon Building: to what Extent Is the Manual Effort Worth? In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)* [online]. Portorož, Slovenia: European Language Resources Association (ELRA), 2016, pp. 938–942. Available also from: https://aclanthology.org/L16-1149.pdf.

21. KAMPS, Jaap; MARX, Maarten; MOKKEN, Robert J.; RIJKE, Maarten de. Using WordNet to Measure Semantic Orientations of Adjectives. In: *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)* [online]. Lisbon, Portugal: European Language Resources Association (ELRA), 2004. Available also from: http://www.lrec-conf.org/proceedings/lrec2004/pdf/734.pdf.

22. CHOMSKY, Noam. *Syntactic Structures*. The Hague, Netherlands: Mouton & Co., 1957. ISBN 9789027933850.

23. JIANQIANG, Zhao; XIAOLIN, Gui. Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis. *IEEE Access* [online]. 2017, vol. 5, pp. 2870–2879. Available from DOI: 10.1109/ACCESS.2017.2672677.

24. PRADHA, Saurav; HALGAMUGE, Malka N.; TRAN QUOC VINH, Nguyen. Effective Text Data Preprocessing Technique for Sentiment Analysis in Social Media Data. In: *2019 11th International Conference on Knowledge and Systems Engineering (KSE)* [online]. 2019, pp. 1–8. Available from DOI: 10.1109/KSE.2019.8919368.

25. MEJOVA, Yelena. Sentiment analysis: An overview. *University of Iowa, Computer Science Department* [online]. 2009. Available also from: https://www.researchgate.net/profile/Yelena-Mejova/publication/264840229_Sentiment_Analysis_An_Overview/links/590ad68e0f7e9b1d0823eff2/Sentiment-Analysis-An-Overview.pdf.

26. MEJOVA, Yelena; SRINIVASAN, Padmini. Exploring Feature Definition and Selection for Sentiment Classifiers. *Proceedings of the International AAAI Conference on Web and Social Media* [online]. 2021, vol. 5, no. 1, pp. 546–549. Available from DOI: 10.1609/icwsm.v5i1.14163.

27. DAS, Sanjiv Ranjan; CHEN, Mike Y. Yahoo! for Amazon: Sentiment parsing from small talk on the web. *For Amazon: Sentiment Parsing from Small Talk on the Web (August 5, 2001). EFA* [online]. 2001. Available from DOI: 10.2139/ssrn.276189.

28. PRINCETON UNIVERSITY. *WordNet* [online]. [N.d.]. Available also from: https://wordnet.princeton.edu. Accessed: May 3, 2023.

29. MILLER, George A.; BECKWITH, Richard; FELLBAUM, Christiane; GROSS, Derek; MILLER, Katherine J. Introduction to WordNet: An On-line Lexical Database. *International Journal of Lexicography* [online]. 1990, vol. 3, no. 4, pp. 235–244. ISSN 0950-3846. Available from DOI: 10.1093/ijl/3.4.235.

30. LAM, Khang Nhut; TAROUTI, Feras Al; KALITA, Jugal. Automatically constructing Wordnet Synsets. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* [online]. Association for Computational Linguistics, 2014, pp. 106–111. Available from DOI: 10.3115/v1/p14-2018.

31. SEBASTIANI, Fabrizio; ESULI, Andrea. Sentiwordnet: A publicly available lexical resource for opinion mining. In: *Proceedings of the 5th international conference on language resources and evaluation* [online]. European Language Resources Association (ELRA) Genoa, Italy, 2006, pp. 417–422. Available also from: `https://www.researchgate.net/publication/200044289_SentiWordNet_A_Publicly_Available_Lexical_Resource_for_Opinion_Mining`.

32. BACCIANELLA, Stefano; ESULI, Andrea; SEBASTIANI, Fabrizio, et al. Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In: *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)* [online]. 2010, vol. 10, pp. 2200–2204. Available also from: `http://www.lrec-conf.org/proceedings/lrec2010/pdf/769_Paper.pdf`.

33. SWEENEY, Colm Joseph. *Sentiment Analysis on Twitter feeds to establish opinion towards entities in single entity and multi-entity texts* [online]. 2019. Available also from: `https://pureadmin.qub.ac.uk/ws/portalfiles/portal/198632863/final_thesis.pdf`. PhD thesis. Queen's University Belfast. Faculty of Engineering and Physical Sciences.

34. MIKOLOV, Tomas; CHEN, Kai; CORRADO, Greg; DEAN, Jeffrey. Efficient Estimation of Word Representations in Vector Space. In: *International Conference on Learning Representations (2013)* [online]. 2013. Available from DOI: `10.48550/arXiv.1301.3781`.

35. MA, Long; ZHANG, Yanqing. Using Word2Vec to process big text data. In: *2015 IEEE International Conference on Big Data (Big Data)* [online]. 2015, pp. 2895–2897. Available from DOI: `10.1109/BigData.2015.7364114`.

36. DEVLIN, Jacob; CHANG, Ming-Wei; LEE, Kenton; TOUTANOVA, Kristina. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)"* [online]. 2018, pp. 4171–4186. Available from DOI: `10.48550/arXiv.1810.04805`.

37. HOANG, Mickel; BIHORAC, Oskar Alija; ROUCES, Jacobo. Aspect-Based Sentiment Analysis using BERT. In: *Proceedings of the 22nd Nordic Conference on Computational Linguistics* [online]. Turku, Finland: Linköping University Electronic Press, 2019, pp. 187–196. Available also from: `https://aclanthology.org/W19-6120`.

38. NOZZA, Debora; BIANCHI, Federico; HOVY, Dirk. *What the [MASK]? Making Sense of Language-Specific BERT Models* [online]. 2020. Available from DOI: `10.48550/arXiv.2003.02912`.

39. LIU, Yinhan; OTT, Myle; GOYAL, Naman; DU, Jingfei; JOSHI, Mandar; CHEN, Danqi; LEVY, Omer; LEWIS, Mike; ZETTLEMOYER, Luke; STOYANOV, Veselin. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* [online]. 2019. Available from DOI: `10.48550/arXiv.1907.11692`.

40. HUTTO, C.; GILBERT, Eric. VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. *Proceedings of the International AAAI Conference on Web and Social Media* [online]. 2014, vol. 8, no. 1, pp. 216–225. Available from DOI: 10.1609/icwsm.v8i1.14550.

41. HITKUL; AGGARWAL, Karmanya; YADAV, Neha; DWIVEDY, Maheshwar. A Comparative Study of Machine Learning Algorithms for Prior Prediction of UFC Fights. In: YADAV, Neha; YADAV, Anupam; BANSAL, Jagdish Chand; DEEP, Kusum; KIM, Joong Hoon (eds.). *Harmony Search and Nature Inspired Optimization Algorithms* [online]. Singapore: Springer Singapore, 2018, vol. 741, pp. 67–76. ISBN 978-981-13-0761-4. Available from DOI: 10.1007/978-981-13-0761-4_7.

42. MCQUAIDE, McKinley. Applying Machine Learning Algorithms to Predict UFC Fight Outcomes [online]. 2019. Available also from: http://cs229.stanford.edu/proj2019aut/data/assignment_308832_raw/26647731.pdf.

43. JOHNSON, Jeremiah Douglas. *Predicting outcomes of mixed martial arts fights with novel fight variables* [online]. 2012. Available also from: https://getd.libs.uga.edu/pdfs/johnson_jeremiah_d_201208_ms.pdf. PhD thesis. University of Georgia.

44. MARTINEZ-RÍOS, Erick. Machine learning applied to a UFC match database [online]. 2021. Available from DOI: 10.6084/m9.figshare.13567184.v1.

45. TURGUT, Mehmetcan. *Machine Learning approach to predicting Mixed Martial Arts matches* [online]. 2021. Available also from: http://arno.uvt.nl/show.cgi?fid=156304. MA thesis. Tilburg University.

46. MCCABE, Alan; TREVATHAN, Jarrod. Artificial Intelligence in Sports Prediction. In: *Fifth International Conference on Information Technology: New Generations (itng 2008)* [online]. 2008, pp. 1194–1197. Available from DOI: 10.1109/ITNG.2008.203.

47. DEXOPLEX. *Does Reddit limit the depth of comment nesting at all?* [online]. 2014. [visited on 2023-05-07]. Available from: https://www.reddit.com/r/Enhancement/comments/1zoly3/does_reddit_limit_the_depth_of_comment_nesting_at/.

48. RMMA. *[Official] UFC 229 Press Conference: Khabib vs McGregor - Discussion Thread* [online]. 2018. [visited on 2023-05-07]. Available from: https://www.reddit.com/r/MMA/comments/9hebqk/official_ufc_229_press_conference_khabib_vs/.

49. DEAN, Brian. *How Many People Use YouTube in 2023? [New Data]* [online]. 2023. [visited on 2023-05-07]. Available from: https://backlinko.com/youtube-users.

50. DEAN, Brian. *Reddit User and Growth Stats (Updated March 2023)* [online]. 2023. [visited on 2023-05-07]. Available from: https://backlinko.com/youtube-users.

51. GOOGLE DEVELOPERS. *YouTube Data API - Quota and Compliance Audits* [online]. 2023. [visited on 2023-05-07]. Available from: https://developers.google.com/youtube/v3/guides/quota_and_compliance_audits.

52. DEAN, Brian. *Instagram Demographic Statistics: How Many People Use Instagram in 2023?* [Online]. 2023. [visited on 2023-05-07]. Available from: https://backlinko.com/instagram-users.

53. SACNILK. *List Of Most Followed UFC Fighters on Instagram* [online]. 2023. [visited on 2023-05-07]. Available from: https://wwe.sacnilk.com/news/List_Of_Most_Followed_UFC_on_Instagram.

54. DEAN, Brian. *How Many People Use Twitter in 2023? [New Twitter Stats]* [online]. 2023. [visited on 2023-05-07]. Available from: https://backlinko.com/instagram-users.

55. PANZARINO, Matthew. *Interesting fact: more Tweets posted are 28 characters than any other length [Updated]* [online]. 2012. [visited on 2023-05-07]. Available from: https://thenextweb.com/news/interesting-fact-most-tweets-posted-are-approximately-30-characters-long.

56. JUSTANOTHERARCHIVIST. *snscrape: A social networking service scraper in Python* [online]. 2023. Available also from: https://github.com/JustAnotherArchivist/snscrape.

57. JIANG, Long; YU, Mo; ZHOU, Ming; LIU, Xiaohua; ZHAO, Tiejun. Target-dependent twitter sentiment classification. In: *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies* [online]. 2011, pp. 151–160. Available also from: https://aclanthology.org/P11-1016.pdf.

58. MCKINNEY, Wes. Data Structures for Statistical Computing in Python. In: VAN DER WALT, Stéfan; MILLMAN, Jarrod (eds.). *Proceedings of the 9th Python in Science Conference*. 2010, pp. 56–61. Available from DOI: 10.25080/Majora-92bf1922-00a.

59. KLUYVER, Thomas; RAGAN-KELLEY, Benjamin; PÉREZ, Fernando; GRANGER, Brian; BUSSONNIER, Matthias; FREDERIC, Jonathan; KELLEY, Kyle; HAMRICK, Jessica; GROUT, Jason; CORLAY, Sylvain; IVANOV, Paul; AVILA, Damián; ABDALLA, Safia; WILLING, Carol. Jupyter Notebooks – a publishing format for reproducible computational workflows. In: LOIZIDES, F.; SCHMIDT, B. (eds.). *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. IOS Press, 2016, pp. 87–90.

60. RICHARDSON, Leonard. *Beautiful Soup* [online]. 2004. Available also from: https://www.crummy.com/software/BeautifulSoup/.

61. HARRIS, Charles R.; MILLMAN, K. Jarrod; WALT, Stéfan J. van der; GOMMERS, Ralf; VIRTANEN, Pauli; COURNAPEAU, David; WIESER, Eric; TAYLOR, Julian; BERG, Sebastian; SMITH, Nathaniel J.; KERN, Robert; PICUS, Matti; HOYER, Stephan; KERKWIJK, Marten H. van; BRETT, Matthew; HALDANE, Allan; RÍO, Jaime Fernández del; WIEBE, Mark; PETERSON, Pearu; GÉRARD-MARCHANT, Pierre; SHEPPARD, Kevin; REDDY, Tyler; WECKESSER, Warren; ABBASI, Hameer; GOHLKE, Christoph; OLIPHANT, Travis E. Array programming with NumPy. *Nature*. 2020, vol. 585, no. 7825, pp. 357–362. Available from DOI: 10.1038/s41586-020-2649-2.

62. HILL, Adam. *A Timeline of UFC Rules: From No-Holds-Barred to Highly Regulated* [online]. 2013. [visited on 2023-04-27]. Available from: https://bleacherreport.com/articles/1614213-a-timeline-of-ufc-rules-from-no-holds-barred-to-highly-regulated.

63. ULTIMATE FIGHTING CHAMPIONSHIP. *Unified Rules of Mixed Martial Arts* [online]. 2018. [visited on 2023-04-27]. Available from: https://www.ufc.com/unified-rules-mixed-martial-arts.

64. ULTIMATE FIGHTING CHAMPIONSHIP. *UFC HALL OF FAME FAQ* [online]. 2023. [visited on 2023-04-27]. Available from: `https://www.ufcespanol.com/news/ufc-hall-fame-faq`.

65. ELO, Arpad E. *The Rating of Chess Players, Past and Present*. Second Edition. New York, NY, USA: Arco Publishing Company, 1978. ISBN 0-668-04721-6.

66. REID, April M. *Elo Rating System For Video Games Explained* [online]. 2021. [visited on 2023-04-23]. Available from: `https://esportsheadlines.com/elo-rating-system-for-video-games-explained/`.

67. LASEK, Jan; SZLÁVIK, Zoltán; BHULAI, Sandjai. The predictive power of ranking systems in association football. *Int. J. of Applied Pattern Recognition* [online]. 2013, vol. 1, no. 1. Available from DOI: `10.1504/IJAPR.2013.052339`.

68. KOVALCHIK, Stephanie. Searching for the GOAT of tennis win prediction. *Journal of Quantitative Analysis in Sports* [online]. 2016, vol. 12, no. 3, pp. 127–138. Available from DOI: `10.1515/jqas-2015-0059`.

69. WIKIPEDIA CONTRIBUTORS. *Elo rating system — Wikipedia, The Free Encyclopedia* [online]. 2023. [visited on 2023-04-23]. Available from: `https://en.wikipedia.org/w/index.php?title=Elo_rating_system&oldid=1149422986`.

70. PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* [online]. 2011, vol. 12, pp. 2825–2830. Available also from: `https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html`.

71. DOROGUSH, Anna Veronika; ERSHOV, Vasily; GULIN, Andrey. CatBoost: gradient boosting with categorical features support [online]. 2018. Available from DOI: `10.48550/arXiv.1810.11363`.

Appendix A

Appendix

Attached medium contents

- `README.md` ..... brief description of medium contents
- `src`
  - `past-statistics` ..... source code for past-statistics approach
  - `sentiment-analysis` ..... source code for sentiment-analysis approach
  - `combination` ..... combination of previous methods
- `text`
  - `thesis.pdf` ..... text of the thesis in PDF format
  - `thesis.zip` ..... source code of the thesis text in LaTeX
  - `assignment.pdf` ..... assignment in PDF format