



Degree Project in Technology

First cycle, 15 credits

# **Predicting UFC matches using regression models**

**SEBASTIAN APELGREN, CHRISTOFFER EKLUND**

# Abstract

This project applied statistical inference methods to historical data of mixed martial arts (MMA) matches from the Ultimate Fighting Championship (UFC). The goal of the project was to create a model to predict the outcome of Ultimate Fighting Championship matches with the best possible accuracy. The main methods used in the project were logistic regression and Bayesian regression. The data used for said model was taken from matches between early April 2000 and mid April 2024. The predictions made by these models were compared with the predictions of various betting sites as well as with the true outcomes of the matches. The logistic regression model and the Bayesian model predicted the true outcome of the matches 60% and 70% of the time respectively, with both having comparable predictions to those of the betting sites.

## Keywords

Bayesian regression, Logistic regression, MMA, Sports forecasting, UFC

# Sammanfattning

Detta projekt tillämpar statistiska inferensmodeller på data från mixed martial arts (MMA) matcher från Ultimate Fighting Championship (UFC). Målet med projektet var att skapa en model för att förutspå resultatet av Ultimate Fighting Championship matcher med så god noggrannhet som möjligt. De huvudsakliga metoderna som användes i projektet var logistisk regression och Bayesiansk regression. Datan som användes för att skapa modellen kom ifrån matcher mellan början på April 2000 och mitten på April 2024. Resultaten som förutspåddes av modellerna jämfördes med resultaten som olika spelsidor hade förutspått, samt med de sanna resultaten från matcherna. Den logistiska regressionsmodellen och den Bayesianska modellen lyckades förutspå det sanna resultatet 60% respektive 70% av matcherna vilket var jämförbart med spelsidornas träffsäkerhet.

## Nyckelord

Bayesiansk regression, Logistisk regression, MMA, sportprognoser, UFC

# Acknowledgements

We would like to express our heartfelt gratitude to our supervisor, Liam Solus. His infectious enthusiasm and remarkable patience throughout the project despite our endless stream of questions were invaluable. We are truly grateful for his mentorship and support.

# Acronyms

**MLE:** Maximum Likelihood Estimate. 7, 8, 16

**MMA:** Mixed Martial Arts. 1, 2, 14, 21, 22, 23

**UFC:** Ultimate Fighting Championship. 1, 2, 3, 21, 22, 23

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Data</b>	<b>3</b>
2.1	Data collection . . . . .	3
2.2	Data processing . . . . .	4
<b>3</b>	<b>Method</b>	<b>7</b>
3.1	Regression . . . . .	7
3.1.1	Logistic regression . . . . .	8
3.1.2	Newton's method . . . . .	9
3.2	Bayesian statistics . . . . .	10
3.2.1	Bayesian regression . . . . .	11
3.3	Models . . . . .	12
3.4	Implementation . . . . .	12
3.4.1	Logistic regression fighter-model . . . . .	12
3.4.2	Bayesian regression fighter-model . . . . .	13
3.4.3	Scikit-learn fighter-model . . . . .	14
3.5	Variable selection . . . . .	14
<b>4</b>	<b>Results</b>	<b>16</b>
<b>5</b>	<b>Analysis and Discussion</b>	<b>21</b>
5.1	Analysis . . . . .	21
5.2	Discussion . . . . .	22
5.3	Conclusion . . . . .	23
5.4	Future work . . . . .	23
<b>A</b>	<b>Code</b>	<b>25</b>

# Chapter 1

## Introduction

Sport forecasting is an important aspect of sports and can be used in many different ways. For instance, by analyzing historical data, weather conditions, and their physical condition, athletes can forecast their potential performance for an upcoming event and then optimize their game and recovery strategies accordingly [1, 2]. Other areas where sports forecasting can be useful are injury prevention, official ranking of players for tournament seeding purposes as well as for the setting of betting odds [3, 4, 5].

In this project, sports forecasting was used to predict the outcome of sports events, more precisely, Ultimate Fighting Championships (UFC) matches. UFC is the premier organization in the world of mixed martial arts (MMA) and showcases a wide array of fighters from various martial arts disciplines competing against one another under unified rules. MMA is an individual-reliant sport and the outcome of a match is heavily dependent on a fighter's performance, as well as the performance of their opponent. These performances are made up of a variety of different factors like the number of takedowns and strikes, as well as being in control. These factors are captured in metrics, like hit counts and amount of time spent in control, that are used to judge matches. An MMA match can generally end in two different ways. Either one fighter is rendered unable to continue fighting, either by getting knocked out or by getting forced to submit and in doing so conceding, or by getting called by a set of judges after the allotted time runs out. If a match is to be determined by the judges the aforementioned metrics play a fundamental role in determining the winner. Because of this, these metrics were deemed critical in developing the models to predict the outcome of matches for this project.

The main question that this thesis aims to answer is: *Can we construct a predictive model for the outcomes of UFC matches based on historical data?* with the subsequent questions: *If such a model can be constructed, how accurate can we make it?*

The aim of this project was to gain a deeper understanding of predictive models built around limited data sets for sporting events. More specifically, the performance of standard models for sports forecasting, *e.g.* logistic regression models, as well as Bayesian regression models, were explored.

Two papers that served as inspiration for this project were [4] and [6]. In [4] Clarke and Dyte simulated major tennis tournaments using a logistic regression model based on the official player rankings in combination with the match results, ending up with reasonably good results. In [6] Lam made one-match-ahead forecasts for matches in a major basketball league using a Bayesian regression model. They defined one-match-ahead forecasting based on the assumption that player ability changes smoothly across consecutive matches if occasional variation in performance is taken into account. Based on this assumption match  $g_{k+1}$  was predicted using the  $k$  previous matches. Lam also used a larger set of features to represent each player's performance, in contrast to Clarke and Dyte. This model ended up with impressively good results. Based on these papers it seems that there is some precedent of both logistic regression and Bayesian regression successfully being used in sports forecasting. Hence, for this project, some elements from these two articles were adopted. Lam's one-match-ahead forecasting philosophy was used both in the making of the predictions, in that the predicted performance of each athlete for a given match is made based on the most recent matches, as well as in the way training data was handled, in that each data point was made using a set of subsequent matches. Lam's approach of using a larger set of features to represent player performance also ended up being used.

Based on this two fighter-models were constructed, one using logistic regression and one using Bayesian regression. Both make use of one-match-ahead forecasting as well as a larger set of features to represent player performance. The key difference between them being that the logistic regression model only yields point estimates, while the Bayesian regression model yields a posterior distribution from which prediction can be drawn, granting further insight into the uncertainty of the model predictions.

In the end the predictions of these two models when compared to that of a collection of sports betting

sites [7], as well as the true outcomes over a set of 20 matches ended up being similar in accuracy with the Bayesian regression model doing the best.

The scope of this project was limited in that the only data considered for it was that of matches where both fighters were some of the most seasoned competitors of all time in the UFC (at the time of writing). More specifically, only fighters with  $\geq 20$  matches on record and a win rate between 15% and 85% at the time of the match in question were considered. The exact reasons for these delimitations will be discussed in Section 5.2. The project was also limited to using available data, meaning that only easily measurable data was used. Other more nebulous factors, like fighting style and current form, were not considered. Another thing that was not considered is how the rules used for the UFC might have changed during the period from which the data was collected.

The fact that the Bayesian model outperformed the other ones, despite it not being the typical kind of model used for sports forecasting, might suggest that further investigation into Bayesian predictive models for MMA could be a worthwhile endeavor.

The thesis is structured in the following way. In Chapter 2, the data as well as the data pre-processing strategy used in the project are described. In Chapter 3, the models used and some of the theory behind them are introduced. In Chapter 4, the results for both the logistic regression and the Bayesian regression fighter-models are presented. Finally, in Chapter 5, the results are analyzed and discussed and some suggestions for future work are provided.

# Chapter 2

## Data

In this chapter, the data that was used for the project will be discussed. In Section 2.1 the raw data that was collected will be described and an example of what it might look like will be given. Section 2.2 will describe the ways that the raw data was processed before being used in the fighter-models.

### 2.1 Data collection

The data used in this project was taken from the match results published to the official UFC website [8]. The file containing the data that was gathered can be found in Appendix 5. From the results for each match, the following data was gathered:

- The names of both fighters participating in the match
- Who won the match | Binary value
- The total time the match took, including how many rounds it went on for | Discrete minute and second value
- The total amount of knockdowns performed by each fighter (a fighter striking their opponent, causing them to fall constitutes a knockdown) | Discrete positive values
- The total amount of significant strikes attempted as well as successfully performed by each fighter (a hit using the fists, elbows, knees, or feet constitutes a significant strike) | Discrete positive values
- The total amount of takedowns attempted as well as successfully performed by each fighter (a fighter grappling their opponent and bringing them to the ground constitutes a takedown) | Discrete positive values
- The total amount of attempts to force a submission from the opponent by each fighter | Discrete positive value
- The total time each fighter was in control while grappling on the ground | Discrete minute and second value
- The total amount of reverses performed by each fighter (a fighter going from not being in control to being in control while grappling constitutes a reverse) | Discrete positive value
- The total amount of significant strikes towards the head, body, or legs attempted as well as successfully performed by each fighter | Discrete positive values
- The total amount of significant strikes while at range, while grappling standing up, or while grappling on the ground attempted as well as successfully performed by each fighter | Discrete positive values

This set of data was collected for every match from early April 2000 to mid April 2024. Following this the collected data was sorted to extract the data points pertaining to the athletes that the fighter-models were to be based on, this being some of the most seasoned competitors across the men's weight classes as of early 2024.

An example of one of the data points that ended up being used can be seen in Table 2.1,

Name	Charles Oliveira	Beneil Dariush
Winner	Charles Oliveira	-
Time	4:10	-
Knockdowns	1	0
Significant strikes	26/36	12/28
Takedowns	0/1	0/0
Submission attempts	0	0
Control	0:31	2:44
Reverses	0	0
Significant strikes to the head	23/32	8/24
Significant strikes to the body	3/3	3/3
Significant strikes to the legs	0/1	1/1
Significant strikes while at range	7/13	6/14
Significant strikes while grappling standing	3/3	0/0
Significant strikes while grappling on the ground	16/20	6/14

Table 2.1: Example of what a data point might look like.

where data of the form  $x/y$  should be read as  $x$  successful attempts out of  $y$  total attempts.

These variables are the typical things used by the judges to score a match if it goes to decision. They are therefore a good candidate for independent variables in a regression model. However, to achieve better model performance, functions of these statistics will be carefully crafted and used, as described in the following section.

## 2.2 Data processing

At first, the fighter-models were constructed using the unmodified data directly, with the data containing two values being represented as such. Wins and losses were represented as ones and zeros as this convention conformed with the prediction-model's expectations.

These early attempts were quickly discovered to be flawed in several ways. Firstly, the possibility of vast differences in match length presented an issue. Seeing as matches ending prematurely is a very real option, finishing due to knockout or submission, for example, the minimum length of a match is essentially zero. The shortest match in the data set went on for only five seconds. However, matches often do go on for the entirety of the allotted time with the winner being decided based on points given out by the judges. With the standard format of three rounds of five minutes each that results in a fifteen-minute match. However, some fights, referred to as title bouts, are extended to be able to go for five rounds instead of the usual three. The result of all this is that matches may vary in length from just a few seconds up to nearly half an hour.

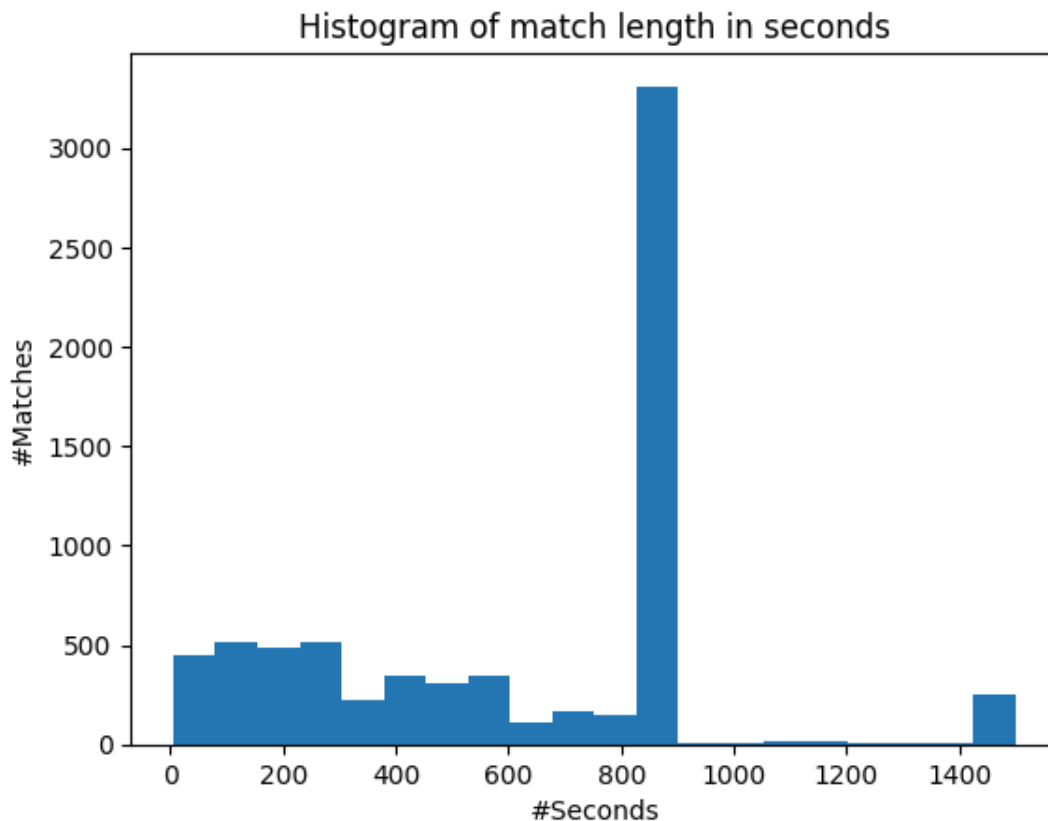


Figure 2.1: Histogram showing the spread of match length in seconds. The peak is located at 900 seconds which is the time for a full 3 round match.

Using the data without taking this into account naturally led to large differences in values between fights. It is not difficult to imagine that the amount of strikes performed by a fighter would vary substantially between a 60-second fight and a 600-second one. This was remedied by normalizing the numerical values down to per second. Another concern would then be that for longer fights, for example, the amount of significant strikes might decrease as the fighters grow increasingly tired towards the end of the match. However, this should not be a problem seeing as the referee is always making sure the fighters are fighting. The referee will give warnings to fighters who are not fighting. Also, continuing the example of less significant strikes, the fighters will then probably grapple instead, which will balance out the statistics.

Secondly, treating the variables defined by pairs of values, for example, successful takedowns and attempted takedowns, as tuples turned out to be challenging as doing so meant that their impact when it came to the number of variables used doubled in some sense. The benefit of having significantly more data than variables for predictive models is well established within statistics [9]. Generally speaking, more data makes models better, and more variables make them more difficult to get right. While this is greatly simplifying the concept, it gives the right idea. Based on this it can be understood that allowing most values to essentially represent two separate, dependent variables would not be advisable. Because of this, as well as the fact that having more variables led to issues with convergence for the logistic regression model, the decision was made to only consider the first value in each pair, *i.e.* the number of successful takedowns performed.

These changes seemed promising for the majority of the different data values. However, there were some where the resulting value made little sense. One example of this was the total amount of significant strikes. Seeing as this seemingly is one of the most important variables, with both significant strikes to the head,

body, and legs as well as significant strikes while at range, grappling standing and grappling on the ground just being different ways of dividing total significant strikes into finer categories, simply using the process described earlier was deemed insufficient. This was because just using the total amount of significant strikes successfully performed fails to take the success rate into account. For the less significant variables this loss of information was deemed acceptable in the pursuit of keeping the total number of variables used low but in the case of significant strikes the choice was made to add the success rate in the form of a decimal percentage as an extra variable.

Having remade the fighter-models after making these modifications the results were promising but the number of variables was still causing problems with convergence for the logistic regression model. This, in combination with the fact that both significant strikes per second and significant strike success rate had turned out to be some of the most used variables while not necessarily being completely independent, the choice was made to combine them into a single variable by multiplying them together. While it turned out that this would be the final reduction in the number of variables needed to finally get results somewhat reliably it also ran the risk of introducing a variable with a nonlinear relation to the output. The failure to extensively investigate whether or not the variables used were in fact linear in terms of the output will be discussed in Section 5.4.

Finally, the actual data points used in the fighter-models were running averages of each variable over a set number of matches. In short, the first data point would be made up of the averages of the first  $s$  values for each variable, with  $s = 5$  being used for the final results. The next data point would then use the averages of values number 2 through  $(s + 1)$  and so on. This was done, even though it meant losing out on  $s$  data points, in order to smooth out the data and avoid some of the fluctuations that binary outcomes would otherwise produce.

In summary, each individual match is represented by a list  $\mathbf{x}_i = (x_{i1}, \dots, x_{i4})$  with each element in the list corresponding to an independent variable. A total of 4 independent variables were used and  $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)^T$  denotes the matrix of all predictors, where each row correspond to a match and each column to an independent variable. For each fight, the modelled fighters' stats will be used together with the stats for the opponent, with the model not taking who the opponent is into account. This will better model the events in the match and it will give a better understanding to the models how to determine the winner. Why this works for the models in this project will be explained in the coming chapter. In the end, the variables used for the fighter-models were

- Significant strikes - fighter |  $x_{i1} = \frac{\text{fighters successful significant strike}}{\text{fighters total significant strikes}} \cdot \frac{\text{fighters successful significant strikes}}{\text{time in seconds}}$
- Significant strikes - opponent |  $x_{i2} = \frac{\text{opponents successful significant strike}}{\text{opponents total significant strikes}} \cdot \frac{\text{opponents successful significant strikes}}{\text{time in seconds}}$
- Control percentage - fighter |  $x_{i3} = \frac{\text{fighters time spent in control}}{\text{time in seconds}}$
- Control percentage - opponent |  $x_{i4} = \frac{\text{opponents time spent in control}}{\text{time in seconds}}$

with  $i$  denoting the  $i$ -th fight.

The data points in  $\mathbf{X}$  used in the fighter-models will go through this transformation

$$\hat{x}_{ij} = \frac{1}{s} \sum_{k=i}^{i+s-1} x_{kj} \quad (2.1)$$

$$\hat{x}_{ij} \in \hat{\mathbf{X}} \quad (2.2)$$

before being input into the models as the rolling mean is used. Further in this paper,  $\mathbf{X}$  will be used to denote  $\hat{\mathbf{X}}$ .

# Chapter 3

## Method

In this chapter, the methods that were used for the project will be described. Section 3.1-3.2 describes the logistic and the Bayesian methods of regression. Following this Section 3.3-3.4 explains the various models constructed as a part of this project and how they were implemented. Finally Section 3.5 discusses the choice of variables to base the fighter-models on.

### 3.1 Regression

Regression models are essential tools in statistics, enabling the prediction of a dependent variable based on the values of one or more independent variables. The simplest form of regression is simple linear regression with Gaussian errors, which is useful for understanding the relationship between two variables.

In simple linear regression, the model relates the continuous dependent variable  $y$  with the independent variables  $x$  through a linear equation:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i \quad (3.1)$$

where:

- $y_i$  is the  $i$ -th observed value of the dependent variable.
- $x_i$  is the  $i$ -th observed value of the independent variable.
- $\beta_0$  and  $\beta_1$  are the parameters representing the intercept and the slope of the regression line, respectively.
- $\epsilon_i$  represents the error term for the  $i$ -th observation, which is assumed to be Gaussian with mean zero and constant variance  $\sigma^2$  ( $\epsilon_i \sim N(0, \sigma^2)$ )

The main objective in simple linear regression is to estimate the parameters  $\beta_0$  and  $\beta_1$ . This is most commonly done by minimizing the sum of the least square errors of the regression line. In other words, to minimize the following expression

$$S(\beta_0, \beta_1) = \sum_{i=1}^n (y_i - (\beta_0 + \beta_1 x_i))^2. \quad (3.2)$$

The values of  $\beta_0$  and  $\beta_1$  that minimize Equation (3.2) are known as the least squares estimators.

When the error terms  $\epsilon_i$  are normally distributed, the least square estimators have a direct connection to the maximum likelihood estimation (MLE). The idea behind MLE is to maximize the likelihood function  $L$  which is the probability of seeing the observations  $y_i$  given the parameter values  $\beta_0$  and  $\beta_1$  in the model seen in Equation (3.1). In this case, the likelihood function for the observations based on the normal distribution of errors is

$$L(\beta_0, \beta_1) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - (\beta_0 + \beta_1 x_i))^2}{2\sigma^2}\right).$$

Maximizing this likelihood function with respect to  $\beta_0$  and  $\beta_1$ , with a given  $\sigma$ , leads to the same estimators as the least square estimators.

When the output variable is discrete *e.g* only takes the values 0 or 1, a linear model will not generally work seeing as they provide continuous outputs over the entire real line. This is a relatively common issue given the abundance of situations with binary outcomes. One solution to this problem is to use a logistic regression model.

### 3.1.1 Logistic regression

For a logistic regression model, firstly let  $P(Y|\mathbf{X})$  denote the probability of a certain output  $Y$  given some input  $\mathbf{X}$ . A probability is preferred over simply outputting one of the two possible binary results as it is more nuanced. For example, a 51% chance and a 99% chance can both represent the same binary output, while conveying different levels of confidence in the predicted result. With this, the probability of a  $Y = 1$  outcome becomes  $P(Y = 1|\mathbf{X} = \mathbf{x})$  which will be referred to as  $p(\mathbf{x})$  going forward. Assuming that separate observations of  $\mathbf{x}$  and  $y$  are independent while  $p(\mathbf{x})$  stays the same for a given  $\mathbf{x}$  the likelihood function is

$$\prod_{i=1}^n P(Y = y_i|\mathbf{X} = \mathbf{x}_i) = \prod_{i=1}^n p(\mathbf{x}_i)^{y_i} (1 - p(\mathbf{x}_i))^{1-y_i} \quad (3.3)$$

which resembles a sequence of Bernoulli trials [10]. The likelihood of success for a sequence of Bernoulli trials is maximized by  $p = \hat{p} = \frac{1}{n} \sum_{i=1}^n y_i$ . This does however present a problem. In the current situation  $\hat{p}_i = 1$  when  $y_i = 1$  and  $\hat{p}_i = 0$  when  $y_i = 0$ . This estimation does not add anything new. This is solved by thinking of  $p(\mathbf{x}_i)$  as  $p(\mathbf{x}_i; \theta)$  where  $\theta$  is an unknown variable. This makes the likelihood a function of  $\theta$ , which can in turn be estimated by maximizing the likelihood. This is what lies at the core of regression models.

In solving this by fitting a regression model the first assumption would be that  $p(\mathbf{x}; \theta)$  is a linear function. However,  $p$  must be between 0 and 1 meaning that an unbounded linear function will not work. The solution here is to apply the logistic transformation to  $p$ , resulting in  $\log \frac{p}{1-p}$ , and making this a linear function of  $x$ . The result is

$$\log\left(\frac{p(\mathbf{x})}{1-p(\mathbf{x})}\right) = \beta_0 + \mathbf{x} \cdot \boldsymbol{\beta} \quad (3.4)$$

which is the formal logistic regression model [10]. Solving this for  $p$  gives

$$p(\mathbf{x}; \beta_0, \boldsymbol{\beta}) = \frac{e^{\beta_0 + \mathbf{x} \cdot \boldsymbol{\beta}}}{1 + e^{\beta_0 + \mathbf{x} \cdot \boldsymbol{\beta}}} = \frac{1}{1 + e^{-(\beta_0 + \mathbf{x} \cdot \boldsymbol{\beta})}} \quad (3.5)$$

Seeing as logistic regression represents probabilities we can fit it using the likelihood. Each data point in the training data is made up of a vector of features  $\mathbf{x}_i$  and a result  $y_i$ . The probability of that  $\mathbf{x}_i$  was then either  $p$  or  $1 - p$ , for  $y_i = 1$  and  $y_i = 0$  respectively. Based on this the likelihood is then

$$L(\beta_0, \boldsymbol{\beta}) = \prod_{i=1}^n p(\mathbf{x}_i)^{y_i} (1 - p(\mathbf{x}_i))^{1-y_i} \quad (3.6)$$

[10]. Rewriting this as the log-likelihood gives

$$\ell(\beta_0, \boldsymbol{\beta}) = \sum_{i=1}^n y_i \log(p(\mathbf{x}_i)) + (1 - y_i) \log(1 - p(\mathbf{x}_i)) \quad (3.7)$$

$$= \sum_{i=1}^n y_i \log\left(\frac{p(\mathbf{x}_i)}{1 - p(\mathbf{x}_i)}\right) + \sum_{i=1}^n \log(1 - p(\mathbf{x}_i)) \quad (3.8)$$

$$= \sum_{i=1}^n y_i (\beta_0 + \mathbf{x}_i \cdot \boldsymbol{\beta}) + \sum_{i=1}^n \log(1 - p(\mathbf{x}_i)) \quad (3.9)$$

$$= \sum_{i=1}^n y_i (\beta_0 + \mathbf{x}_i \cdot \boldsymbol{\beta}) + \sum_{i=1}^n -\log(1 + e^{\beta_0 + \mathbf{x}_i \cdot \boldsymbol{\beta}}). \quad (3.10)$$

Normally the MLE would be found by taking the derivative of the log-likelihood function with respect to the parameters and solving for said parameters at the point where the derivative is equal to zero. Attempting this, while denoting the component of  $\boldsymbol{\beta}$  being derived with respect to as  $\beta_j$ , results in the function

$$\frac{\partial \ell}{\partial \beta_j} = \sum_{i=1}^n y_i x_{ij} - \sum_{i=1}^n \frac{e^{\beta_0 + \mathbf{x}_i \cdot \boldsymbol{\beta}}}{1 + e^{\beta_0 + \mathbf{x}_i \cdot \boldsymbol{\beta}}} x_{ij} \quad (3.11)$$

$$= \sum_{i=1}^n (y_i - p(\mathbf{x}_i; \beta_0, \boldsymbol{\beta})) x_{ij} = 0 \quad (3.12)$$

which is unfortunately not exactly solvable [10]. An approximate solution can however be found numerically using for example Newton's method.

### 3.1.2 Newton's method

Newton's method is a numerical optimization method used to find minimums of functions. This is done through an iterative process by gradually refining an initial guess of where the minimum might be until the true minimum is reached. Starting with the simplest case, a single scalar variable, minimizing the function  $f(\beta)$  to find the global minimum  $\beta^*$ . Assuming that  $f(\beta)$  is smooth,  $\beta^*$  being the global minimizer means that  $f'(\beta^*) = 0$  and  $f''(\beta^*) > 0$ . For points near  $\beta^*$  the Taylor expansion

$$f(\beta) \approx f(\beta^*) + \frac{1}{2}(\beta - \beta^*)^2 f''(\beta^*) \quad (3.13)$$

is a good approximation of  $f(\beta)$  seeing as  $f(\beta)$  is continuous. Newton's method makes use of this approximation by minimizing this quadratic function instead of the often times more complicated  $f(\beta)$  [10]. It starts out with a somewhat close initial guess  $\beta^{(0)}$ , makes the same Taylor expansion,

$$f(\beta) \approx f(\beta^{(0)}) + (\beta - \beta^{(0)})f'(\beta^{(0)}) + \frac{1}{2}(\beta - \beta^{(0)})^2 f''(\beta^{(0)}) \quad (3.14)$$

which it then minimizes by setting the derivative to zero at some point  $\beta^{(1)}$  and solving for  $\beta^{(1)}$ ,

$$0 = f'(\beta^{(0)}) + (\beta^{(1)} - \beta^{(0)})f''(\beta^{(0)}) \quad (3.15)$$

$$\beta^{(1)} = \beta^{(0)} - \frac{f'(\beta^{(0)})}{f''(\beta^{(0)})} \quad (3.16)$$

with  $\beta^{(1)}$  as the updated guess. The process is then repeated and only stops once  $\beta^*$  is reached, seeing as it is a fixed point ( $f'(\beta^*) = 0$ ). In practice, the algorithm is implemented in such a way that it terminates when  $|\beta^{(n+1)} - \beta^{(n)}|$  is less than some set tolerance. This is done because finding the exact value of  $\beta^*$  numerically is infeasible when numerical uncertainty is taken into account.

Newton's method also generalizes to a multivariate version. This is done by replacing the update of  $\beta^{(n)}$  that occurs in every iteration with

$$\boldsymbol{\beta}^{(n+1)} = \boldsymbol{\beta}^{(n)} - H^{-1}(\boldsymbol{\beta}^{(n)})\nabla f(\boldsymbol{\beta}^{(n)}) \quad (3.17)$$

where  $\nabla f$  is the gradient of  $f$ ,  $\nabla_j f = \frac{\partial f}{\partial \beta_j}$  and  $H$  is the Hessian of  $f$ ,  $H_{jk} = \frac{\partial^2 f}{\partial \beta_k \partial \beta_j}$  [10].

In order to use the multivariate version of Newton's method to approximate the maximum likelihood  $\nabla \ell$  and  $H$  need to be calculated. The former was already calculated earlier. Calculating  $H_{jk}$  gives

$$H_{jk} = \frac{\partial^2 \ell}{\partial \beta_k \partial \beta_j} = \frac{\partial}{\partial \beta_k} \left( \sum_{i=1}^n \left( y_i - \frac{1}{1 + e^{-(\beta_0 + \mathbf{x} \cdot \boldsymbol{\beta})}} \right) x_{ij} \right) \quad (3.18)$$

$$= \frac{\partial}{\partial \beta_k} \sum_{i=1}^n - \frac{x_{ij}}{1 + e^{-(\beta_0 + \mathbf{x} \cdot \boldsymbol{\beta})}} \quad (3.19)$$

$$= - \sum_{i=1}^n \frac{\partial}{\partial \beta_k} \frac{x_{ij}}{1 + e^{-(\beta_0 + \mathbf{x} \cdot \boldsymbol{\beta})}} \quad (3.20)$$

$$= - \sum_{i=1}^n \frac{x_{ij} x_{ik}}{(1 + e^{-(\beta_0 + \mathbf{x} \cdot \boldsymbol{\beta})})^2 e^{\beta_0 + \mathbf{x} \cdot \boldsymbol{\beta}}} \quad (3.21)$$

$$= - \sum_{i=1}^n \frac{x_{ij} x_{ik}}{e^{-(\beta_0 + \mathbf{x} \cdot \boldsymbol{\beta})} + 2 + e^{\beta_0 + \mathbf{x} \cdot \boldsymbol{\beta}}}. \quad (3.22)$$

An important, but easily overlooked, detail of using Newton's method for logistic regression is that the log-likelihood is to be maximized while Newton's method finds minimums. This can be remedied by applying Newton's method in minimizing  $-\ell(\beta_0, \boldsymbol{\beta})$  instead of  $\ell(\beta_0, \boldsymbol{\beta})$ .

## 3.2 Bayesian statistics

Bayesian statistics is a branch of statistics that provides a mathematical framework for updating beliefs in light of new information. It is based on Bayes Theorem which describes how to change predictions or hypotheses given new data. This approach is different from a frequentist approach, like logistic regression.

In a frequentist approach, parameters are seen as fixed values. This approach is reliant on consistency to make predictions and inferences. In the case of logistic regression, the parameters are predicted based on a fixed amount of data by maximizing the likelihood of observing the given data in the model. Then, while keeping the parameters constant, predictions are made on new data.

On the other hand, Bayesian models treat the parameters as random variables with their own distributions. This probabilistic approach makes it possible to incorporate prior beliefs and knowledge about the parameters which are then updated when new data comes in. To incorporate prior belief, a prior distribution that best fits the data is decided. After updating it, the outcome is a posterior distribution for the parameters, which provides a full spectrum of possible values weighted by their probability. This is done through Bayes Rule which gives a precise relation between the prior distribution together with the likelihood and the posterior distribution. Bayes Theorem can be mathematically stated as follows

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (3.23)$$

where  $P(B|A)$  is the likelihood,  $P(A)$  is the prior distribution,  $P(B)$  is the marginalization and  $P(A|B)$  is the posterior distribution.

This approach offers multiple advantages over frequentist statistics. Firstly, it provides a natural way to incorporate prior knowledge into the model. Expert opinions can easily be incorporated into the model and thus, if implemented in a field where data is expensive or rare, the model will perform better. Secondly, it is easy to update the posterior distribution when new data is available, making it great for dynamic environments where data is constantly flowing in. This is done by setting the posterior distribution as the prior distribution and then update with the new data. Lastly, the outcome is a posterior distribution, which allows for easy analysis of uncertainties.

There are different ways to choose a prior distribution based on the available knowledge and assumptions and they can vary a lot. However, there might not always be prior knowledge or assumptions to make. A non-informative prior is then chosen, which is an improper distribution that contains no prior knowledge about the data. Improper means that it is not a real distribution as it doesn't integrate to 1. The benefit of using a non-informative prior is the objectivity, no assumptions are made about the parameters which creates an objective analysis based on the new data only rather than also having a subjective belief beforehand. They are also very versatile and applicable in many different fields. When using an improper prior, it is important to verify that the posterior distribution is proper. It is also important to verify that the non-informative prior minimizes the influence on the posterior distribution. In some cases, especially when the sample size is small, the non-informative prior can influence the posterior distribution. Another prior distribution that can be used is a conjugate prior. A conjugate prior is a prior distribution that, when combined with a likelihood function (or data distribution) from a specific family, results in a posterior distribution that is of the same family as the prior. This relationship simplifies the computational process of Bayesian updating.

Following is a quick example of a conjugate prior distribution being updated with new available data. Let's assume the data  $X$  is distributed as a binomial distribution.

$$X \sim \text{Bin}(m, \theta) \quad (3.24)$$

A prior distribution for the parameter  $\theta$  is then a Beta distribution for given  $\alpha$  and  $\beta$ .

$$\Theta \sim \text{Beta}(\alpha, \beta) \quad (3.25)$$

When new data is available, the prior distribution is then updated to become the posterior distribution given by:

$$\Theta | \mathbf{X} \sim \text{Beta}(\alpha + n\bar{x}, \beta + nm - n\bar{x}) \quad \text{where,} \quad (3.26)$$

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (3.27)$$

and  $n$  is the amount of new data points.

### 3.2.1 Bayesian regression

One can take a Bayesian approach to linear regression. First and foremost, take a look at the outcome variable  $Y$

$$Y = \begin{cases} 1 & \text{if the fighter wins the match} \\ 0 & \text{if the fighter loses the match} \end{cases} \quad (3.28)$$

which models the outcome of a match. Of course, this variable is binary, either it is a win ( $Y = 1$ ) or a loss ( $Y = 0$ ). However, for simplicity,  $Y$  will be modelled as a continuous random variable and a trend analysis perspective will be taken when using the resulting predictions. Trend analysis is typically used in time series to predict trends over time. It is commonly used in climate models [11] or when predicting the stock market [12]. However, one can also examine shifts in a response variable following shifts in the posterior predictive distribution's mean towards zero or one as the independent variables changes value.

The variables  $\mathbf{x} = (x_1, \dots, x_m)$  are called explanatory variables and may be discrete or continuous. We often examine the distribution of  $y$  conditional on  $\mathbf{x}$  within a group of units or experimental subjects, labeled  $i = 1, \dots, n$ . For each subject  $i$ , measurements include  $y_i$  and  $x_{i1}, \dots, x_{im}$ . The index  $i$  denotes individual units, while  $j$  denotes the components of  $\mathbf{x}$ . The vector  $\mathbf{y}$  represents the outcomes for all  $n$  subjects, and  $\mathbf{X}$  denotes the  $n \times m$  matrix of predictors. In this model, the variable  $x_{i1}$  is fixed at 1, so that  $\beta_1 x_{i1}$  is constant for all values of  $i$ . Furthermore, it is assumed that the conditional variances are equal and independent of each other,  $\text{var}(y_i | \theta, \mathbf{X}) = \sigma^2$  for all  $i$ , and the observations  $y_i$  are conditionally independent given  $\theta$  and  $\mathbf{X}$ . The parameter vector is then  $\theta = (\beta_1, \dots, \beta_m, \sigma)$  [13]. A Bayesian model with a normally distributed data distribution will be used

$$\mathbf{Y} | \mathbf{X}, \theta \sim N(\mathbf{X}\boldsymbol{\beta}, \sigma^2 I_n). \quad (3.29)$$

When choosing a prior distribution, a non-informative prior distribution was chosen [13]. A convenient non-informative prior distribution is uniform on  $(\boldsymbol{\beta}, \log(\sigma))$

$$p(\boldsymbol{\beta}, \sigma^2 | \mathbf{X}) \propto \sigma^{-2}. \quad (3.30)$$

A non-informative prior is useful when there are few parameters and many data points and it takes less effort to implement than specifying prior knowledge in probabilistic form. In this project, fighters with a lot of data points will be prioritized over fighters with less data points to avoid errors in the distribution.

When determining the posterior distribution, first the posterior distribution for  $\boldsymbol{\beta}$ , conditional on  $\sigma$ , is determined, and then the marginal posterior distribution for  $\sigma^2$ . It will be done by factorizing the joint distribution for  $\boldsymbol{\beta}$  and  $\sigma^2$  as  $p(\boldsymbol{\beta}, \sigma^2 | \mathbf{y}, \mathbf{X}) = p(\boldsymbol{\beta} | \sigma^2, \mathbf{y}, \mathbf{X}) \cdot p(\sigma^2 | \boldsymbol{\beta}, \mathbf{X})$  [13].

For the conditional posterior distribution of  $\boldsymbol{\beta}$ , which is the posterior distribution given a specific variable, in this case given  $\sigma$ , it can be shown that

$$f(\boldsymbol{\beta} | \sigma^2, \mathbf{y}, \mathbf{X}) \propto f(\mathbf{y} | \boldsymbol{\beta}, \sigma^2, \mathbf{X}) \quad (3.31)$$

$$\propto \exp\left(-\frac{1}{2\sigma^2(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})}\right). \quad (3.32)$$

Thus,  $\boldsymbol{\beta} | \sigma^2, \mathbf{y}, \mathbf{X}$  is normal [13].

$$\boldsymbol{\beta} | \sigma^2, \mathbf{y}, \mathbf{X} \sim \mathcal{N}(\hat{\boldsymbol{\beta}}, V_\beta \sigma^2), \text{ where} \quad (3.33)$$

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

$$V_\beta = (\mathbf{X}^T \mathbf{X})^{-1}$$

Secondly, the marginal posterior distribution of  $\sigma^2$  can be written as [13]

$$p(\sigma^2 | \mathbf{y}, \mathbf{X}) = \frac{p(\boldsymbol{\beta}, \sigma^2 | \mathbf{y}, \mathbf{X})}{p(\boldsymbol{\beta} | \sigma^2, \mathbf{y}, \mathbf{X})} \quad (3.34)$$

$$\implies \sigma^2 | \mathbf{y}, \mathbf{X} \sim \text{Inv} - \chi^2(n - m, s^2), \text{ where} \quad (3.35)$$

$$s^2 = \frac{1}{n - m} (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})^T (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}). \quad (3.36)$$

Note that when generating a sample from the posterior predictive distribution for the Bayesian regression model, the first thing to do is to generate a sample  $\sigma^2$ s from its distribution, Equation (3.35), and then use those to generate a sample  $\beta$  from its distribution, Equation (3.33). These  $\beta$  and  $\sigma^2$  are then used in conjunction with  $\mathbf{X}$  to generate a new sample,  $\tilde{\mathbf{y}}$ , independent of  $\mathbf{y}$ , using Equation (3.29).

### 3.3 Models

For this project, a few models were constructed. Firstly there is the prediction-model which centers around predicting the outcome of a given match between two different fighters. Each fighter is in turn modelled using a fighter-models, based either on logistic regression or Bayesian regression. The fighter-models work by modelling a fighter based on their performance as well as on the performance of their opponents using  $\mathbf{x}_i$ , as specified in Section 2.2, as the independent variables with wins/losses as the dependent variable. This is done to build up a model of how the fighter does when facing an arbitrary opponent. The prediction-model then uses the fighter-models to simulate a match between two fighters by letting both modelled fighters face the  $s - 1$ , remember  $s$  represents the number of matches that the data points are averaged over with  $s = 5$  being used in this case, opponents that they met before the fight in question. Based on these matches the probability of each fighter winning their next match against an arbitrary opponent is calculated. The fighter with the greater probability is then predicted to win the simulated match. Note that the actual matchup between the two fighters in question never gets simulated.

### 3.4 Implementation

The models were implemented using Python, making use of libraries like SciPy and Matplotlib.

There were three different fighter-models implemented as a part of this project, a logistic regression model using Newton's method for its numerical solver, a Bayesian regression model as well as a proprietary version of the logistic regression model, using the library scikit-learn, to be used as a point of comparison for the two other fighter-models.

#### 3.4.1 Logistic regression fighter-model

The logistic regression fighter-model starts by fetching every data point concerning one specific fighter using a chosen set of variables. It then passes this data set, along with a randomized vector of  $\beta$ s,  $\beta$ , representing the initial guess, to Newton's method. Newton's method will then run as previously described until it converges to some final  $\beta$  which it then passes back. This  $\beta$  is then used to calculate the prediction of the fighter in question's chance at winning. This process is then repeated for the rest of the fighters whose matches have been selected for simulation. Finally, the fighters are paired up based on the matches and whoever has the highest likelihood of winning based on their predictions will be chosen as the one predicted to win their respective match.

The pseudocode for the logistic regression fighter-model can be found in Algorithm 1. However, before the algorithm is presented some functions need to be defined.

- **DATAEXTRACTOR**( $\mathbf{d}$ , *fighter*): this function returns a list of all matches from the data set  $\mathbf{d}$  that *fighter* participated in.
- **DATAPROCESSOR**( $\mathbf{d}$ ,  $s$ ): this function processed the data points  $\mathbf{d}$  in the manner described towards the end of Section 2.2, averaging over  $s$  data points.
- **VARIABLESELECTOR**( $\mathbf{d}$ ,  $v$ ): this function extracts the parts of  $\mathbf{d}$  that contain the variables  $v$  and return them as a list.
- **NEWTON**( $\mathbf{x}$ ,  $\beta_0$ ): this function, as described in Section 3.1.1, runs Newton's method with  $\beta_0$  as its initial guess and returns the  $\beta$  that it converges on.
- **P**( $\mathbf{x}$ ,  $\beta_0$ ,  $\beta$ ): see Equation (3.5).

---

**Algorithm 1:** Logistic regression using Newton's method

---

**Input:** list of historical data of every match  $\mathbf{d}$ , a list of matches to be simulated  $\mathbf{m}$ , a list of fighters in said matches  $\mathbf{f}$ , number of data point to average over  $s$ , variables to use  $\mathbf{v}$

**Output:** predictions of who will win each match in  $\mathbf{m}$

**Function predictions():**

```
for fighter in  $\mathbf{f}$  do
    rawData  $\leftarrow$  DATAEXTRACTOR( $\mathbf{d}$ , fighter)
    processedData  $\leftarrow$  DATAPROCESSOR(rawData,  $s$ )
    finalData  $\leftarrow$  VARIABLESELECTOR(processedData,  $\mathbf{v}$ )
     $\beta_0 \leftarrow$  random integer  $\in [-0.1, 0.1]$ 
     $\beta \leftarrow$  newton(finalData,  $\beta_0$ )
    // uses the most recent data point for making the prediction
    fighterPrediction  $\leftarrow$  P(finalData[0 : end][end],  $\beta[0]$ ,  $\beta[1 : \text{end}]$ )
end
for match in  $\mathbf{m}$  do
    for fighter in match do
        | print(fighterPrediction)
    end
end
```

---

### 3.4.2 Bayesian regression fighter-model

The Bayesian regression fighter-model shares many similarities with the first one in the way it functions. The initial fetching and processing of data is implemented in the same way. The same goes for comparing the prediction made for each fighter to predict who will win the match in question. The main difference lies in how  $\beta$  is calculated as well as how the  $\beta$  is used to predict the likelihood of each fighter to win, with the second fighter-model using the Bayesian approach.

The pseudocode for the Bayesian regression fighter-model can be found in Algorithm 2. However, before the algorithm is presented some additional functions need to be defined.

- GENERATESIGMA( $\mathbf{X}$ ,  $\mathbf{y}$ ,  $\hat{\beta}$ ): this function returns a value  $\sigma^2$ , see Equation (3.35).
- GENERATEBETA( $\mathbf{X}$ ,  $\mathbf{y}$ ,  $\sigma^2$ ,  $\hat{\beta}$ ): this function returns a list  $\beta$ , see Equation (3.33).
- BAYES( $\mathbf{X}$ ,  $n$ ,  $\beta$ ,  $\sigma^2$ ): this function calculates the prediction based on  $\mathbf{X}$ ,  $\beta$  and  $\sigma^2$   $n$  times and returns the mean of these results. See Equation (3.29).

---

**Algorithm 2:** Bayesian regression

---

**Input:** list of historical data of every match  $\mathbf{d}$ , a list of matches to be simulated  $\mathbf{m}$ , a list of fighters in said matches  $\mathbf{f}$ , number of data point to average over  $s$ , variables to use  $\mathbf{v}$ , number of samples to use per prediction  $n$

**Output:** predictions of who will win each match in  $\mathbf{m}$

**Function predictions():**

```
for fighter in  $\mathbf{f}$  do
    rawData  $\leftarrow$  DATAEXTRACTOR( $\mathbf{d}$ , fighter)
    processedData  $\leftarrow$  DATAPROCESSOR(rawData,  $s$ )
    finalData  $\leftarrow$  VARIABLESELECTOR(processedData,  $\mathbf{v}$ )
    // saves the most recent data point for making the prediction
     $\mathbf{X} \leftarrow$  finalData[1 : end][0 : end-1]
     $\mathbf{y} \leftarrow$  finalData[0][0 : end-1]
     $\hat{\beta} \leftarrow (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$ 
     $\sigma^2 \leftarrow$  GENERATESIGMA( $\mathbf{X}$ ,  $\mathbf{y}$ ,  $\hat{\beta}$ )
     $\beta \leftarrow$  GENERATEBETA( $\mathbf{X}$ ,  $\mathbf{y}$ ,  $\sigma^2$ ,  $\hat{\beta}$ )
    // uses the most recent data point for making the prediction
    fighterPrediction  $\leftarrow$  BAYES(finalData[1 : end][end],  $n$ ,  $\beta$ ,  $\sigma^2$ )
end
for match in  $\mathbf{m}$  do
    for fighter in match do
        | print(fighterPrediction)
    end
end
```

---

### 3.4.3 Scikit-learn fighter-model

The final fighter-model simply implements scikit-learns logistic regression model and fits it to the data. The same implementation of fetching and processing of data as well as comparing of predictions is used.

The pseudocode for the logistic regression fighter-model using scikit-learns can be found in Algorithm 3. However, before the algorithm is presented one additional function need to be defined.

- `SCIKIT(d)`: this function creates a scikit-learn `LogisticRegression()`-model, uses the `.fit()` function to fit it to the data set *d* and uses the `.predict_proba()` function to predict the likelihood which it then returns.

---

**Algorithm 3:** Logistic regression using scikit-learn

---

**Input:** list of historical data of every match *d*, a list of matches to be simulated *m*, a list of fighters in said matches *f*, number of data point to average over *s*, variables to use *v*

**Output:** predictions of who will win each match in *m*

**Function predictions():**

```
for fighter in f do
    rawData ← DATAEXTRACTOR(d, fighter)
    processedData ← DATAPROCESSOR(rawData, s)
    finalData ← VARIABLESELECTOR(processedData, v)
    fighterPrediction ← SCIKIT(finalData)
end
for match in m do
    for fighter in match do
        print(fighterPrediction)
    end
end
```

---

All the code written for this project can be found in Appendix 5.

## 3.5 Variable selection

With the fighter-models built and implemented the last step was to select a set of variables to use in them. The goal was for these variables to be able to encapsulate the major fighting styles used in MMA without unfairly favoring some over others. Doing this involved striking a balance between supplying the fighter-models with enough information in order for them to be able to make relevant predictions while keeping the number of variables used low enough to get consistent results from them. It was quickly discovered that some variables were a lot less relevant than others when it came to impacting the accuracy of the predictions. Firstly, the amount of reverses preformed by a fighter turned out to have little to no noticeable impact on the result of the match. This in combination with the fact that they were rarely performed at all meant that their variable did not end up seeing any use. Secondly, the two types of refinements of significant strikes, be that significant strikes to the head, body, and legs and significant strikes while at range, clinching and grappling on the ground, turned out to be too much of a burden on the amount of variables used given the amount of new information they added. This was likely in part due to the fact that they were all in some sense dependent on each other as well as the more general significant strikes variable, greatly limiting the possible amount of unique information that any one of them could have added. Because of this these sets of variables also ended up mostly unused.

This left the number of knockdowns, significant strikes thrown and landed, successful and non-successful takedown attempts, submission attempts, and control time to consider. When the variables knockdowns and submission attempts were used in the model undesirable results were noticed. Submission specialists such as Charles Oliviera saw a significant decrease in his probability of winning when submission attempts were used. This might depend on the fact that he does not have many submission attempts compared to other fighters as, when he gets the chance, he only needs one attempt to end the match in a submission. This is not highlighted in the submission attempt variable as there is no such variable as "submission attempt that ended in a submission". Multiple knockdowns are pretty rare seeing as if a fighter get knocked down there is a high likelihood of their opponent seizing this opportunity and shortly ending the match in a knockout (if the match was not already stopped after the initial knockdown). This made it so that most matches did not have any knockdowns or perhaps only one. Following this, the amount of successful and non-successful takedown attempts and the control time are highly dependent on each other and should therefore not be used simultaneously. If a given fighter successfully performs a takedown, that fighter will be in control and

thus have a longer control time. To best model the grappling and the ground game, it was decided that the control time would be used over the amount of takedowns.

The final variables used were the percentage of time in control while grappling and the combination of significant strikes landed and significant strike success rate ( $=$  total significant strikes landed / total significant strikes), which, after normalizing per second, turned into significant strikes landed times significant strike success rate by second.

# Chapter 4

## Results

In this chapter, the results from the prediction-model are presented. Firstly the results are presented in their entirety in the form of a table. This is then followed by a set of plots summarizing some of the information contained within the results.

For 20 fights, an estimate of the parameters of a logistic regression model for predicting the probability of a fighter winning their next fight, as described in Chapter 3.1, were computed. In these models, the MLE was estimated for the model parameters using Newton’s method. A logistic regression model was also trained for each of the fighters using scikit-learn. In addition to this, a normal Bayesian regression model was also trained from which 1000 samples from the posterior predictive distribution were generated. The arithmetic mean was then computed from these samples to be compared with the other fighters’ arithmetic mean.

There were a total of 31 different fighters who were modelled in these 20 matches. However, there were 40 different models computed as, for each match, all the data for each fighter up until the match in question was used, meaning that the model of a given fighter could not be reused for multiple matches. This resulted in a new model for each fighter in each match, as different data sets were used every time. The data that was used in the models to compute the probability of a fighter winning their next match was the rolling average of the independent variables for the  $s - 1$  last matches. With  $s = 5$ , the last 4 matches were used. This was done so the predictions are based on the current form of the fighters as well as how they compare to others in the sport at the time of the match. For the logistic regression models, an exact  $y$  value is predicted for each fighter and match and the fighter with the highest  $y$  value was predicted to win the match. For the Bayesian regression models the fighter with the highest arithmetic mean of the 1000 samples from the posterior predictive distribution was predicted to win the match.

The betting odds being referred to in the results are a combination of the betting odds presented by a handful of prevalent betting sites that have been archived by [7]. This site serves as a searchable archive allowing the historical betting odds from each match to be found through the fighters involved or through the event which the match was a part of. The odds used for the results were the closing odds for each match, meaning the odds as they were when the match started and betting closed. The prediction being attributed to the odds is then the fighter for whom the payout is the smallest if they end up winning.

Table 4.1: Results from using the different fighter-models as well as the actual winner and the favorite according to the betting odds.

<b>Fighter 1 vs Fighter 2</b>	<b>Year</b>	<b>Winner</b>	<b>Logistic regression</b>	<b>Bayesian</b>	<b>Sklearn</b>	<b>Betting odds</b>
Bobby Green vs Jim Miller	2024	Bobby Green	Bobby Green	Bobby Green	Jim Miller	Bobby Green
Vicente Luque vs Rafael Dos Anjos	2023	Vicente Luque	Vicente Luque	Vicente Luque	Vicente Luque	Rafael Dos Anjos
Tony Ferguson vs Bobby Green	2023	Bobby Green	Bobby Green	Bobby Green	Tony Ferguson	Bobby Green
Charles Oliveira vs Beneil Dariush	2023	Charles Oliveira	Beneil Dariush	Beneil Dariush	Beneil Dariush	Beneil Dariush

Matt Brown vs Court McGee	2023	Matt Brown	Court McGee	Matt Brown	Matt Brown	Court McGee
Gilbert Burns vs Jorge Masvidal	2023	Gilbert Burns	Gilbert Burns	Gilbert Burns	Gilbert Burns	Gilbert Burns
Gilbert Burns vs Neil Magny	2023	Gilbert Burns	Neil Magny	Neil Magny	Gilbert Burns	Gilbert Burns
Drew Dober vs Bobby Green	2022	Drew Dober	Drew Dober	Drew Dober	Drew Dober	Drew Dober
Nate Diaz vs Tony Ferguson	2022	Nate Diaz	Nate Diaz	Nate Diaz	Tony Ferguson	Tony Ferguson
Jim Miller vs Donald Cerrone	2022	Jim Miller	Jim Miller	Jim Miller	Donald Cerrone	Jim Miller
Mauricio Rua vs Ovince Saint Preux	2022	Ovince Saint Preux	Ovince Saint Preux	Ovince Saint Preux	Ovince Saint Preux	Ovince Saint Preux
Cub Swanson vs Darren Elkins	2021	Cub Swanson	Cub Swanson	Cub Swanson	Darren Elkins	Cub Swanson
Michael Johnson vs Clay Guida	2021	Clay Guida	Michael Johnson	Michael Johnson	Clay Guida	Michael Johnson
Anthony Pettis vs Donald Cerrone	2020	Anthony Pettis	Donald Cerrone	Donald Cerrone	Donald Cerrone	Anthony Pettis
BJ Penn vs Clay Guida	2019	Clay Guida	BJ Penn	Clay Guida	Clay Guida	Clay Guida
Matt Brown vs Diego Sanchez	2017	Matt Brown	Diego Sanchez	Diego Sanchez	Diego Sanchez	Matt Brown
Tim Boetsch vs Johny Hendricks	2017	Tim Boetsch	Tim Boetsch	Tim Boetsch	Johny Hendricks	Johny Hendricks
BJ Penn vs Dennis Siver	2017	Dennis Siver	Dennis Siver	Dennis Siver	Dennis Siver	Dennis Siver
Thiago Alves vs Patrick Cote	2017	Thiago Alves	Patrick Cote	Patrick Cote	Thiago Alves	Patrick Cote
Matt Hughes vs Josh Koscheck	2011	Josh Koscheck	Josh Koscheck	Josh Koscheck	Josh Koscheck	Josh Koscheck

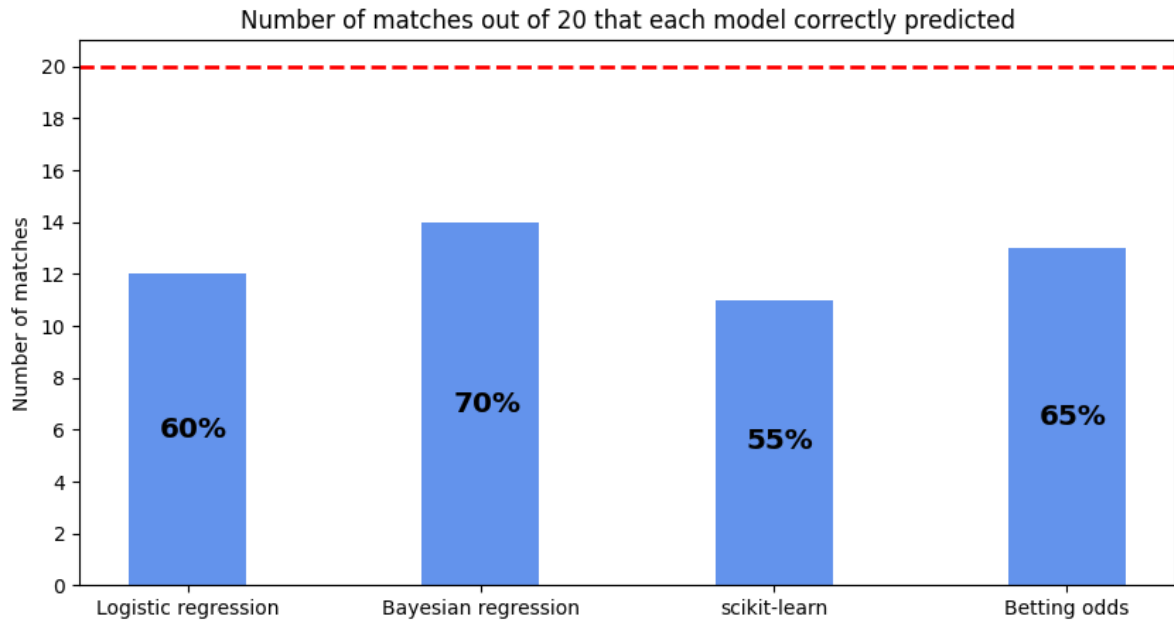


Figure 4.1: Percentage of matches that were predicted correctly divided by source.

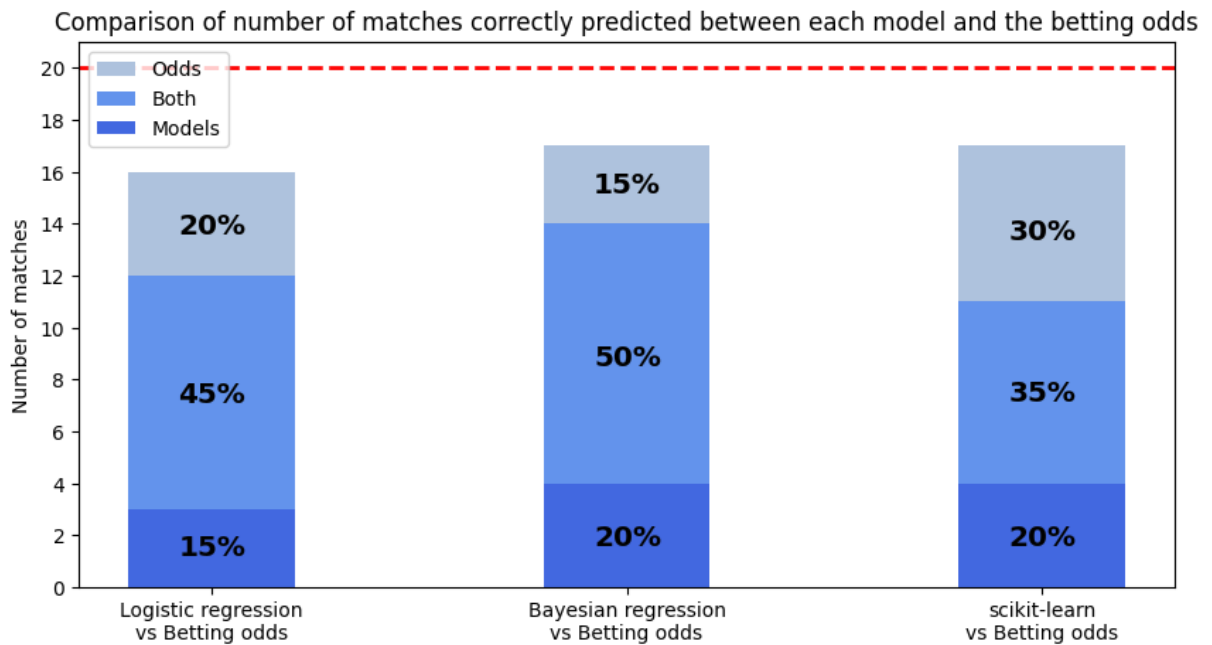


Figure 4.2: Comparison between each individual fighter-model and the betting odds.

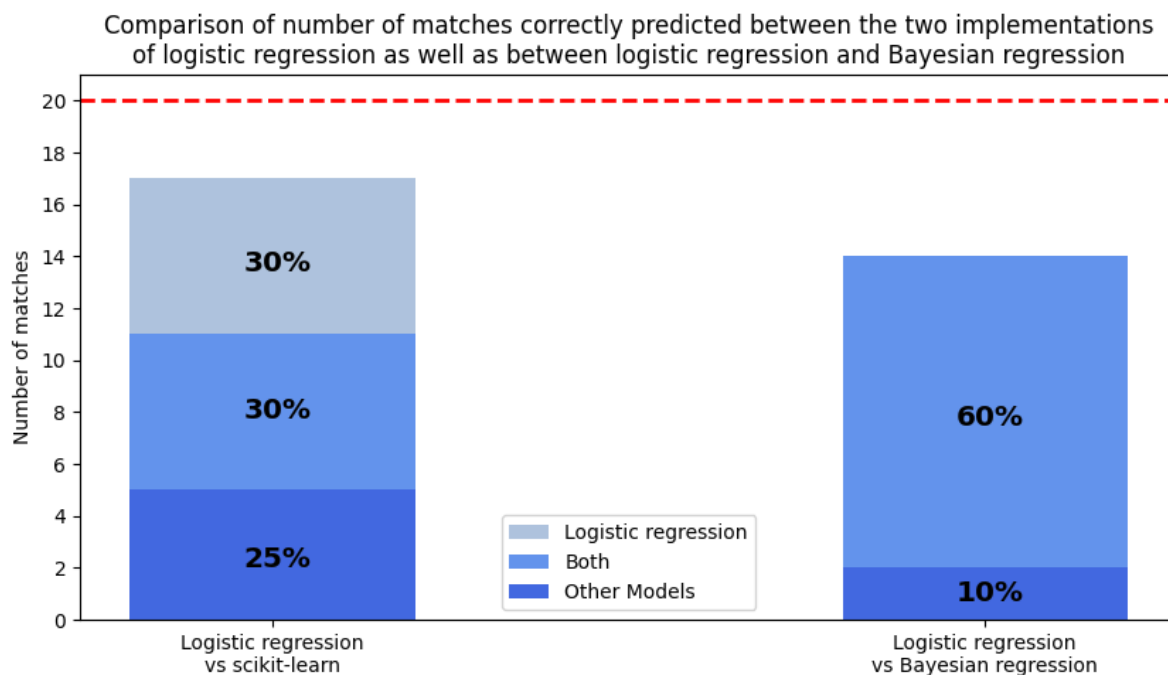


Figure 4.3: Comparison between the two implementations of logistic regression as well as between logistic regression and Bayesian regression.

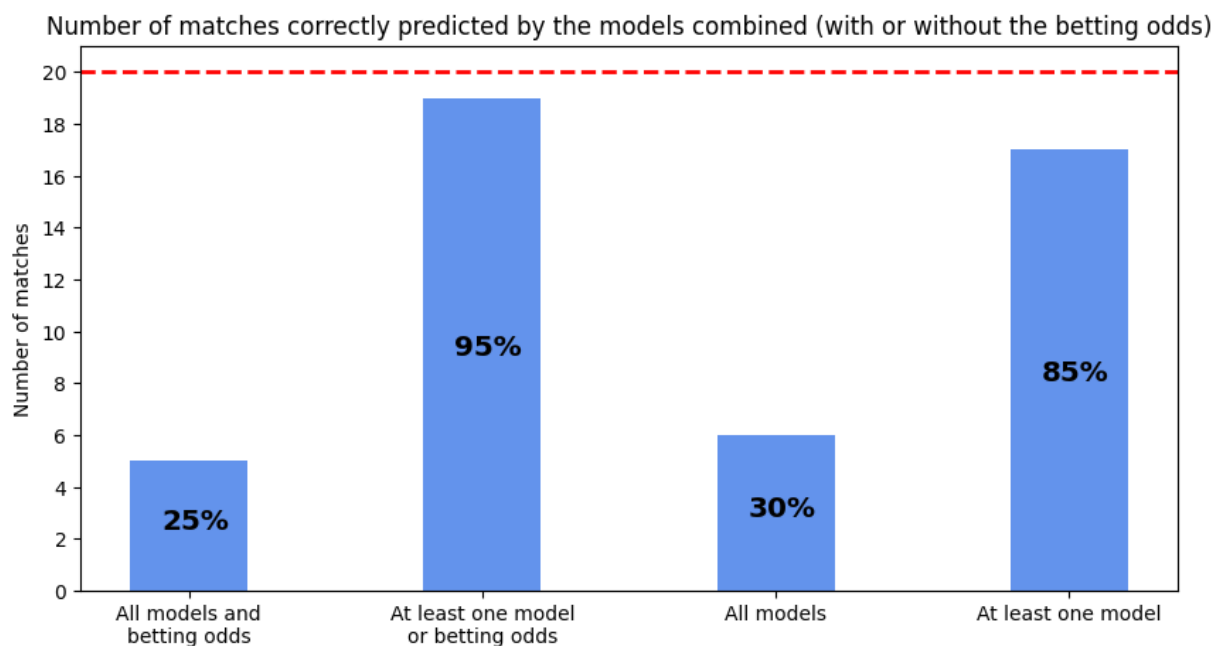


Figure 4.4: Percentage of matches correctly predicted using the different fighter-models and the odds as well as by just the fighter-models.

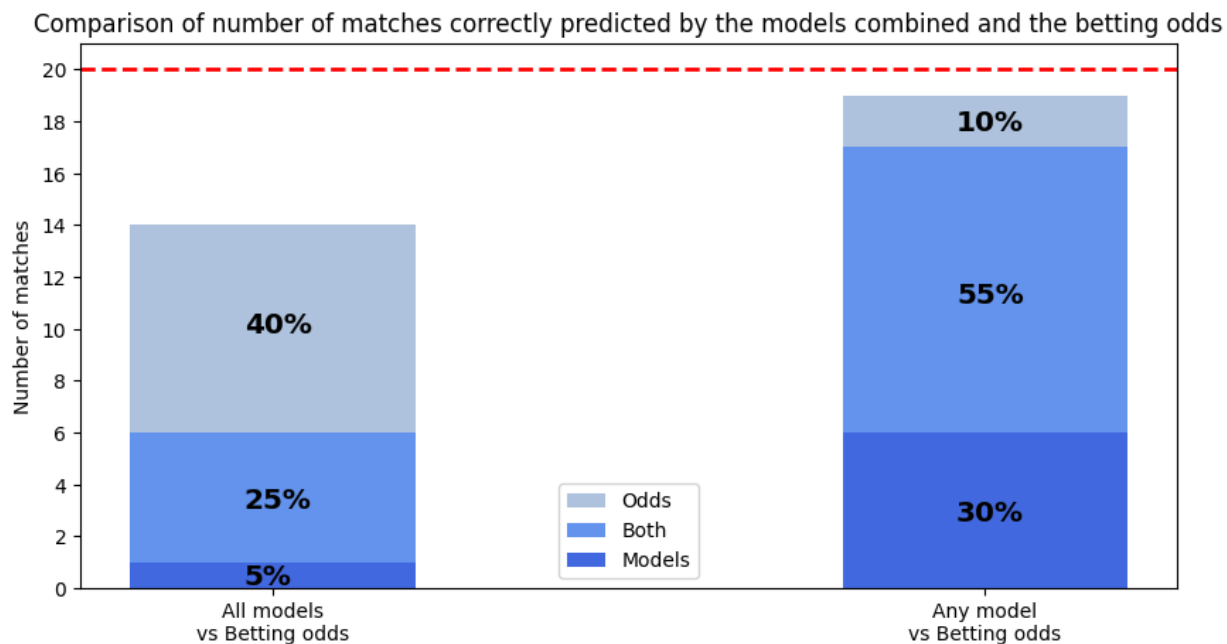


Figure 4.5: Comparison between the combined results of the fighter-models and the betting odds.

In Table 4.1, the predictions of all the different models are presented as well as the actual winner and the predicted winner of the betting odds. Then, in Figure 4.1, the results from Table 4.1 are summarized and each model's accuracy is displayed. Continuing, Figure 4.2 and Figure 4.3 show a more detailed comparison of the predictions between the results from the various models and betting odds. Finally, in Figure 4.4 and Figure 4.5, a more general analysis is conducted on the overall performance of all predictions combined and how all models and betting odds have performed.

# Chapter 5

## Analysis and Discussion

In this chapter, the results are analyzed and discussed. Conclusions are also drawn and potential future work is discussed.

### 5.1 Analysis

From the 20 matches that were simulated, the results were fairly good. As seen in Figure 4.1, the Bayesian regression fighter-model generated the best results with a 70% accuracy and the logistic regression fighter-model had a 60% accuracy. Compared to the betting odds, which had an accuracy of 65%, the Bayesian regression fighter-model was more accurate than them. The scikit-learn fighter-model was the least accurate with a 55% accuracy over the 20 matches.

There are a couple of conclusions that can be drawn from these results. For instance, the unpredictability of the sport is, for example, highlighted by the fight Charles Oliveira vs Beneil Dariush in 2023, where all the models and the betting odds incorrectly predicted that Beneil Dariush would win. Another example is the fight Michael Johnson vs Clay Guida in 2021, where the betting odds and all models, except the scikit-learn implementation, predicted the outcome of the match incorrectly. There will always be matches that end in an unexpected way. In MMA, it only takes one mistake to find oneself knocked out on the ground or forced to submit, so there will always be a certain unpredictability to the sport, which is what makes it enjoyable and interesting to watch.

Moreover, in 2023, the fight Vicente Luque vs Rafael Dos Anjos was correctly predicted by all models in this thesis but incorrectly predicted by the betting odds. This suggests that the betting odds are using other variables in their predictions which are not always giving the upper hand compared to the variables used in this thesis.

In Figure 4.2, when comparing the accuracy of the fighter-models implemented in this project and the betting odds, the logistic regression and the Bayesian regression, both predict correctly together with the betting odds about 50% of the time. The betting odds had 20% of matches correct for which the logistic regression failed to predict the correct outcome and 15% for the Bayesian fighter-model. When comparing the predictions of the Bayesian fighter-model and the logistic regression fighter-model, the Bayesian fighter-model successfully predicted all the matches the logistic regression fighter-model got right, plus a few more, see Figure 4.3. This suggests that the Bayesian fighter-model is better than the logistic regression fighter-model given that it managed to successfully predict an additional few matches that the logistic regression fighter-model failed to predict accurately.

While the accuracy of the UFC prediction-model seems high, there are still matches that no fighter-model managed to predict. In Figure 4.4, it can be observed that at least one of the fighter-models was correct 85% of the time, which isn't that much higher than the most accurate fighter-model (the Bayesian fighter-model). When also considering the betting odds, this number goes up to 95%, which again shows the unpredictability of the sport.

For Figure 4.5, it appears that variable selection is crucial, as the specific variables utilized by betting odds are unknown, whereas the fighter-models implemented in this project consistently use the same set of variables. Notably, there are 10% of matches that the fighter-models failed to predict, which were accurately forecast by the betting odds, suggesting the use of different variables. These variables could be anything from injuries or sickness to the amount of training before the fight. Matches in the UFC can be called off shortly before the match as one of the fighters could get a crucial injury right before the match and therefore not be able to fight. To satisfy the other fighter and the fans, a new fighter can take the match, however, sometimes on a very short notice (a couple of days before). This could then lead to a lack of preparation

and should be considered when predicting the outcome of the match. The betting odds will most likely take things like this into account while the fighter-models in this project could not.

One thing to be critical of when it comes to the results produced by this project is their reliability. Because of the limitations in the fighter-models used, mainly the limitation of only simulating matches between two seasoned fighters, the total amount of predictions that were able to be made was quite low because of the small number of matches that fulfilled the aforementioned criteria. Due to this fact, the results might not necessarily be indicative of how the prediction-model would perform on a larger set of samples. Now, in defense of the sample of matches that was used, the accuracy of the betting odds on these matches is in line with what it tends to be, even on larger samples, looking at the historical data [7]. This could be seen as an indication that the difficulty in predicting the outcomes of the matches used is fairly representative of the data at large. Whether this is the case or not, the reliability of the results is certainly one aspect that should be examined in potential future work.

## 5.2 Discussion

The delimitations used in this project, those being limiting the matches simulated to those consisting only of fighters with  $\geq 20$  matches on record and a win rate between 15% and 85% at the time of the match, were chosen to keep the number of computations needing to be performed down as well as to be able to fine-tune the fighter-model around the limited amount of fighters that ended up being considered. Additionally, they were chosen in part because many of the lesser-known fighters had less data available about them. This would have led to convergence issues for the logistic regression model when using Newton's method to numerically calculate the  $\beta$  and since a non-informative prior was used in the Bayesian regression, more data points is crucial to obtain good results. Originally the thought was to limit the prediction-model to some of the top competitors in a single weight class to have a more confined set of data. However, it became apparent that having this as a limitation would not be viable seeing as the amount of matches that would be feasible to simulate using the prediction-model that had been created would be no more than one or two if even that.

For the first of these limitations, the size of the data set,  $\geq 20$  ended up being used as the cutoff point. This value was found through trial and error by simply rerunning the entire simulation for ever more stringent constraints until the logistic regression fighter-model started converging consistently. When creating a fighter-model for a fighter with a small data set, the data points will not paint a full picture of the fighters fighting style and will not include all the different scenarios that could happen in their next match. Outliers will have a greater impact on the fighter-models and therefore have a higher probability of affecting the prediction-model and thus generate prediction based on inaccurate data. For instance, if a fighter with a small data set has gotten sick before one of their matches and still fought, the data from that match will not be representative of said fighter's potential. A proposed solution for something like this might be to remove any problematic data points. However, this type of information is not available as a part of the stats for a given match. All predictions in this project are made based on the available data and therefore had to assume that the fighter was going to fight their next match in good shape and without any unforeseen drawbacks.

For the latter of the limitations, the win rate of the fighters, the range (0.15, 0.85) ended up being used for the final results, *i.e.*, the matches predicted by the models only contained fighters with a win rate in that range at the time of the match in question. These values were found in a similar way to that of the previous ones, by rerunning the simulation for ever more stringent constraints until it started converging consistently. However, whereas the fact that the size of the data set would be a limiting factor was quite apparent, the win rate being one was more surprising. But in hindsight, the fact that a fighter-model based on attributing a fighter's chance of winning a given match to how they have performed previously might struggle with someone like Jon Jones, who currently has a record of 27-1 with his only loss being a disqualification due to testing positive for illicit substances, is understandable. It is understandable as there is no record of him losing and thus the models will not know what a loss looks like for him. What could happen is that the models will never predict a loss for him as it will seem like, no matter the values of the independent variables, he always wins. Figuring out how the variables in his data correspond to whether or not he will win is not possible if he always wins. Cases like these more often than not led to the numerical methods used, especially Newton's method, having incredibly slow convergence rates with some failing to converge at all. It was because of this that this limitation was added.

These two limitations ended up meaning that the data set that was chosen for this project, that of UFC matches, was quite a difficult one to use. This is because the data set has a high tendency to include these two traits, being a small number of matches per competitor and abnormally high win rates, at least compared to most other sports data. The first of these comes from the fact that MMA is a physically taxing sport. This means that no single fighter should compete too often, in the interest of their own health and general

well-being. Typically a fighter has no more than two to three matches in a year, meaning that one must have a relatively long career behind them to have  $\geq 20$  previous matches. The second of these comes from the nature of the UFC as a sports league. Where as a lot of leagues will contain teams/individuals that are doing poorly as a result of someone else doing well, the UFC tends to only hire the very best within the MMA landscape. Those who are unable to keep up a high win rate are swiftly demoted to a lower league and replaced with someone else. This leads to a lot of newer fighters, further contributing to the difficulty of finding fighters with a large number of previous matches, and artificially high win rates.

What could be done in order to avoid having so few data points would be to consider matches from other organizations where the fighters have fought before joining the UFC. This would enlarge the available data set for those fighters, potentially giving better results for them.

### 5.3 Conclusion

In conclusion, a predictive model for UFC matches based on historical data was successfully implemented using logistic regression as well as Bayesian regression. While doing this a deeper understanding of the inner workings of such methods was gained, especially when it comes to the challenges brought on by a smaller data set. In regards to the accuracy of the fighter-models, the logistic regression model had a 60% accuracy and the Bayesian regression model had a 70% accuracy over the 20 matches simulated. These models achieving a similar accuracy to that of real betting odds, 65% in this case, while utilizing basic statistical methods was well above what was expected at the onset of the project. However, some further work is still needed on the fighter-models to accommodate for fighters with smaller datasets.

### 5.4 Future work

For future work, one could try to predict more matches, maybe specializing in a couple of different weight classes or specific athletes. Other fighter-models could also be explored, such as a Bradley-Terry type model [14], or using a Markov Chain model to simulate matches [15]. As the Bayesian regression fighter-model outperformed the other fighter-models in this project, it could be interesting to try and use a proper prior distribution for the data and then use a Monte Carlo Markov Chain method to draw from the posterior distribution that is computed. Newton's method for numerical optimization could also be replaced with a more resilient and/or more accurate method for numerically approximating the  $\beta$  values as a potential way of improving the logistic regression fighter-model.

After applying the logit function, see Equation (3.4), on the data points when using logistic regression they are assumed to be linear in regards to the logit transform of the outcome. This was not always the case for the data points used in this project and it could be of interested to find appropriate transformations to linearize the data further. This would most likely improve the logistic regression fighter-models and hopefully give even better results.

It could also be interesting to attempt to incorporate some of the more nebulous variables, like fighting style and current form, into the fighter-models. This would require finding a source that can provide such information, which might be difficult to do, but doing so could certainly help improve the fighter-models.

# Bibliography

- [1] D. Percy. “Strategy selection and outcome prediction in sport using dynamic learning for stochastic processes”. In: *Journal of the Operational Research Society* (2015).
- [2] Yan Chu et al. “Machine learning to predict sports-related concussion recovery using clinical data”. In: *Annals of Physical and Rehabilitation Medicine* (2022).
- [3] Alessio Rossi, Luca Pappalardo, and Paolo Cintia. “A Narrative Review for a Machine Learning Application in Sports: An Example Based on Injury Forecasting in Soccer”. In: *Sports* (2022).
- [4] S.R. Clarke and D. Dyte. “Using official ratings to simulate major tennis tournaments”. In: *International Transactions in Operational Research* (2000).
- [5] Erik Štrumbelj and Petar Vračar. “Simulating a basketball match with a homogeneous Markov model and forecasting the outcome”. In: *International Journal of Forecasting* (2012).
- [6] Max W. Y. Lam. “ONE-MATCH-AHEAD FORECASTING IN TWO-TEAM SPORTS WITH STACKED BAYESIAN REGRESSIONS”. In: *Journal of Artificial Intelligence and Soft Computing Research* (2017).
- [7] *Best Fight Odds*. URL: <https://www.bestfightodds.com> (visited on 04/25/2024).
- [8] *UFC Events*. URL: <https://www.ufc.com/events> (visited on 04/25/2024).
- [9] G. Casella and R.L. Berger. *Statistical Inference*. Thomson Learning, 2002.
- [10] C Shalizi. “Chapter 12 Logistic Regression”. In: *Undergraduate Advanced Data Analysis Notes*. Carnegie Mellon University, 2012.
- [11] Sharad K. Jain and Vijay Kumar. “Trend Analysis of Rainfall and Temperature Data for India”. In: *Current Science* (2012).
- [12] Selene Yue Xu and C. U. Berkely. “Stock Price Forecasting Using Information from Yahoo Finance and Google Trend”. In: *UC Berkeley* (2014).
- [13] Gelman A. et al. *Bayesian Data Analysis (3rd ed.)* Chapman and Hall/CRC, 2013.
- [14] Ian McHale and Alex Morton. “A Bradley-Terry type model for forecasting tennis match results”. In: *International Journal of Forecasting* (2011).
- [15] Håvard Rue and Øyvind Salvesen. “Prediction and Retrospective Analysis of Soccer Matches in a League”. In: *Journal of the Royal Statistical Society: Series D (The Statistician)* (2000).

# Appendix A

## Code

All the code written for this project as well as the data set used can be found here:  
[https://github.com/JochenEklund/Bachelor\\_Thesis](https://github.com/JochenEklund/Bachelor_Thesis)

