Camdin and Will

Professor Smalley

Math 239

17 December 2021

UFC Fight Data

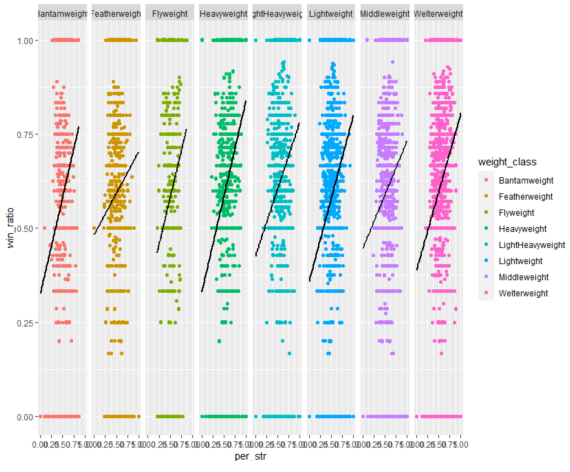
For our research we wanted to predict the results of UFC fights based on the fighters statistics. The dataset we choose is made up of UFC fights and fighters from 1993 to 2021. This dataset contains a combination of raw fighter data and Fight data. In the UFC dataset each observation is a fight which includes fighters' individual stats. There were 6012 observations with 155 variables that are from the fight and each fighter's career. The combination of these two allows us to compare fighters' individual stats such as stance, reach, age and compare that to their competitor and the overall fight outcome. We hope to be able to use this dataset to predict the fight outcomes relationship with the aforementioned variables. There were two main questions we wanted to try and answer, does a fighter's physical and infight stats help predict a fighter's career win rate and does the difference between two fighters in fight and physical stats help predict who wins? After looking at all our models we concluded that our models were unable to predict wins very well.

With our goals and questions in mind we decided to split up our dataset into two. We wanted to look at infight data and fighter career data. With the infight data we wanted to draw conclusions on the results of a fight from our explanatory variables. With the fighter career data we want to draw conclusions on a fighter's career success from our explanatory variables.. From the 155 variables we chose win rate as our response variable for the career dataset and winner as our response variable for infight. For our explanatory fights statistics we chose headshots landed, headshots attempted, body shots landed, body shots attempted, strokes attempted, strikes landed and the percentage for all of those. The explanatory biographical stats we used were height, reach, weight, stance and weight class. Our main scientific question was which fighter variables predict wins?

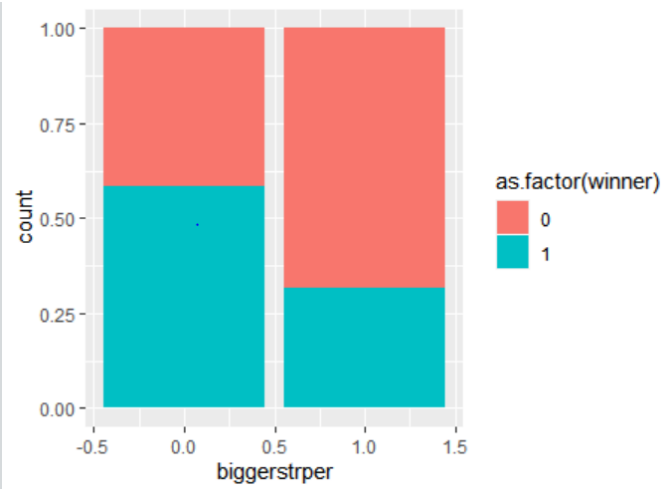
We ran some classification analysis on our infight predictions. We chose this because we want to find out if our explanatory variables could predict if a fighter would win or lose. So we

decided on a binomial logistic regression due to having a binomial response variable. The explanatory variables we were looking at were differences between the two fighters' biographical stats, such as difference between height, weight, and age. We also looked at differences between the two fighters' career statistics, such as difference in strike percentage, headshot percentage and bodyshot percentage. We ran the models to find a model and then tried to make some predictions. Of all the models and predictions we ran on all the different variables we found that difference in strike percentage. We fit a logistic regression model on if the difference in strike percentage has a relationship with if a fighter won or not. The model ended up being significant at an alpha level of .05. It also had an AIC number of 15904. To identify how well this model predicts who wins and who loses a fight we trained the model and created a confusion matrix. We got a 0.31628409847 error rate.

For the career dataset linear regression was the best model to find relationships between our variables. The response variable for all regression was win ratio, which is the number of wins a fighter has over the number of fights they fought. Turning fighters' win loss records into a proportion allows us to draw conclusions on fighters' success without their number of fights distorting those results . The explanatory variables tested were proportion strike , proportion head, proportion body, weight, height, reach and age. The linear regressions between win ratio and all the explanatory variables resulted in the R squared ranging from .0025-.047. All linear regressions had significant P-values. Without any clear relationships in our linear regression we decided that we wanted to see if multiple linear regression would be effective. The response variable and explanatory variables used, were all the same as single linear regressions with the addition of the categorical variables weight class and stance. The regression of win ratio, proportion strike and weight class can be seen on the left. The P-value for proportion strike was significant, but weight class was not. The adjusted R squared the model was . 03568. The last model tested included all explanatory variables except for weight and height because they were not shown to be significant. The R squared for this model was .09745.



For the classification model that we fit the best ended up not being able to predict who wins and loses a fight. Even though we got a significant P-value at the alpha level of .05, our high AIC number of 15904 and high error rate of .316 suggests that strike percentage doesn't predict wins very well. This model was also very similar to many other models also created. Surprisingly every model had a better win ratio if they were on the low end of the explanatory variable. These models all were statically significant at an alpha level of .05. The other model just had higher AIC numbers and higher error rate. The results from the linear regression models were disappointing, there were relationships between our response and any of our explanatory variables. For the l regression models almost all of the p values were significant however all of the R squareds were very small. The multiple linear regressions were more successful, however still nowhere near significant. Model 3 had the best adjusted R^2 of .09745, but this took all significant variables and was still not large enough to be useful. All linear regressions ran had similar problems; None of the plots had a noticeable positive or negative relationship. The response variable win ratio resulted in a very strange distribution of points, which is caused by fighters who have fought a small number of fights >3.



After looking at all our models we concluded that our models were unable to predict wins very well. We believe that one of the reasons that our models were unsuccessful was to do with the complexity of analysing fighting. One big reason fighting is so complex is the way fights are determined if no one gets knocked out. If no one is knocked out or loses within the amount of rounds there is in a fight, then it goes to judges to decide a winner which will include some bias on who they think won the fight. Plus UFC fighting is after all a sport and sports are very variable there are upsets and unexpected endings. Some things we would do differently in future studies is separating weight class and gender of competition. We would look at if a female fight needs to fast where a male fight is slower or vice versa. We would look at if you go to small weight classes does what matters to win a fight change. We would also like to look at more complex variables suchs maybe a win rate with title fights weight more because they mean more

or variables like that. We would also either look into UFC more and learn more about it or give the dataset to someone with more knowledge. This would be because one of our setbacks during the time we were researching is that neither of us knew a lot about fighting or the UFC so we didn't know what somethings meant or what some variables were.

Appendix 1: Code

```
library(tidyverse)

UFCInfight = data %>%
  select(R_fighter, B_fighter, R_Reach_cms,B_Reach_cms,R_Height_cms,B_Height_cms, B_age, R_age, B_Stance, R_Stance, Winner,B_Weight_lbs,R_Weight_lbs
  mutate(reachdifference = R_Reach_cms- B_Reach_cms)%>%
  mutate(heightdifference = R_Height_cms - B_Height_cms)%>%
  mutate(weightdifference = R_Weight_lbs - B_Weight_lbs)%>%
  mutate(agedifference = R_age - B_age)


R_fighter = select(data,R_fighter, B_fighter, R_Reach_cms,B_Reach_cms,R_Height_cms,B_Height_cms, B_age, R_age, B_Stance, R_Stance, Winner,B_Weight_
R_fighter = R_fighter%>%
  mutate(reachdifference = R_Reach_cms- B_Reach_cms)%>%
  mutate(heightdifference = R_Height_cms - B_Height_cms)%>%
  mutate(weightdifference = R_Weight_lbs - B_Weight_lbs)%>%
  mutate(agedifference = R_age - B_age)%>%
  mutate(str_perR = R_avg_TOTAL_STR_landed/ R_avg_TOTAL_STR_att)%>%
  mutate(str_perB = B_avg_TOTAL_STR_landed/ B_avg_TOTAL_STR_att)%>%
  mutate(str_perdifference = str_perR - str_perB)%>%
  rename(fighter = R_fighter)%>%
  rename(winner = Rwinner)%>%
  rename(reach = R_Reach_cms)%>%
  rename(age = R_age)%>%
  rename(weight = R_Weight_lbs)
R_fighter = select(R_fighter, fighter, winner, reach, age, weight, reachdifference, heightdifference, weightdifference, agedifference, str_perdiffer
```

```
B_fighter = select(data,R_fighter, B_fighter, R_Reach_cms,B_Reach_cms,R_Height_cms,B_Height_cms, B_age, R_age, B_Stance, R_Stance, Winner,B_Weight_
B_fighter = B_fighter%>%
  mutate(reachdifference = B_Reach_cms- R_Reach_cms)%>%
  mutate(heightdifference = B_Height_cms - R_Height_cms)%>%
  mutate(weightdifference = B_Weight_lbs - R_Weight_lbs)%>%
  mutate(agedifference = B_age - R_age)%>%
  mutate(str_perR = R_avg_TOTAL_STR_landed/ R_avg_TOTAL_STR_att)%>%
  mutate(str_perB = B_avg_TOTAL_STR_landed/ B_avg_TOTAL_STR_att)%>%
  mutate(str_perdifference = str_perR - str_perB)%>%
  rename(fighter = B_fighter)%>%
  rename(winner = Bwinner)%>%
  rename(reach = B_Reach_cms)%>%
  rename(age = B_age)%>%
  rename(weight = B_Weight_lbs)
B_fighter = select(B_fighter, fighter, winner, reach, age, weight, reachdifference, heightdifference, weightdifference, agedifference, str_perdiffer

All_fighter = bind_rows(B_fighter, R_fighter)

library(leaps)

regfit.fwd<-regsubsets(winner~ biggerreach + biggerheight + biggerstrper + biggerweight + biggerweight + biggerage, data=All_fighter,
                       nvmax = 5, method="forward")
summary(regfit.fwd)
```

```
All_fighter$biggerstrper = 0
for(i in 1:6012){
  if(is.na(All_fighter$str_perdifference[i]) || All_fighter$str_perdifference[i] > 0){
    All_fighter$biggerstrper[i] = 1
  }
  if(is.na(All_fighter$str_perdifference[i])){
    All_fighter$str_perdifference[i] =  NA
  }
}

ggplot(All_fighter, aes(x = biggerstrper, fill=as.factor(winner)))+
  geom_bar(position = "fill")

set.seed(1)
train_indices = sample(1:nrow(All_fighter), size = floor (nrow(All_fighter)/2))
train_data = All_fighter%>%
  slice(train_indices)
test_data <- All_fighter%>%
  slice(-train_indices)

modB = glm(winner ~ biggerstrper , family = "binomial", data = All_fighter)
summary(modB)
test <- predict(modB, newdata = All_fighter, type = "response")
test_mat1<-data.frame(winner=as.factor(test_data$winner), predwinner= test>.5)%>%
  group_by(winner, predwinner)%>%
  summarise(n=n())

test_mat1
```

```
library(tidyverse)
data = UFC.FIght.carrer.

R_fighter = select(data , R_fighter, R_wins , R_losses ,
                   R_draw, R_Reach_cms , R_age , R_Stance , R_Height_cms ,
                   R_Weight_lbs,weight_class , R_avg_TOTAL_STR_att , R_avg_TOTAL_STR_landed,
                   R_avg_HEAD_att , R_avg_HEAD_landed, R_avg_BODY_att,
                   R_avg_BODY_landed)
R_fighter = R_fighter %>%
  rename(fighter = R_fighter)%>%
  rename(wins = R_wins)%>%
  rename(losses = R_losses)%>%
  rename(draws = R_draw)%>%
  rename(stance = R_Stance)%>%
  rename(height = R_Height_cms)%>%
  rename(weight = R_Weight_lbs)%>%
  rename(reach = R_Reach_cms)%>%
  rename(avg_str_att = R_avg_TOTAL_STR_att)%>%
  rename(avg_str_landed = R_avg_TOTAL_STR_landed)%>%
  rename(avg_head_att = R_avg_HEAD_att)%>%
  rename(avg_head_landed = R_avg_HEAD_landed)%>%
  rename(avg_body_att = R_avg_BODY_att)%>%
  rename(avg_body_landed = R_avg_BODY_landed)%>%
  rename(age = R_age)

B_fighter = select(data , B_fighter, B_wins , B_losses ,
                   B_draw, B_Reach_cms , B_age , B_Stance , B_Height_cms ,
                   B_Weight_lbs, weight_class , B_avg_TOTAL_STR_att , B_avg_TOTAL_STR_landed,
                   B_avg_HEAD_att , B_avg_HEAD_landed, B_avg_BODY_att,
                   B_avg_BODY_landed)
B_fighter = B_fighter %>%
  rename(fighter = B_fighter)%>%
  rename(wins = B_wins)%>%
  rename(losses = B_losses)%>%
  rename(draws = B_draw)%>%
  rename(stance = B_Stance)%>%
  rename(height = B_Height_cms)%>%
  rename(weight = B_Weight_lbs)%>%
  rename(reach = B_Reach_cms)%>%
  rename(avg_str_att = B_avg_TOTAL_STR_att)%>%
  rename(avg_str_landed = B_avg_TOTAL_STR_landed)%>%
  rename(avg_head_att = B_avg_HEAD_att)%>%
  rename(avg_head_landed = B_avg_HEAD_landed)%>%
  rename(avg_body_att = B_avg_BODY_att)%>%
  rename(avg_body_landed = B_avg_BODY_landed)%>%
  rename(age = B_age)
All_fighter_career = bind_rows(B_fighter, R_fighter)


All_fighter_career = All_fighter_career%>%
  mutate(per_str = avg_str_landed/avg_str_att)%>%
  mutate(per_head = avg_head_landed/avg_head_att)%>%
  mutate(per_body = avg_body_landed/avg_body_att)%>%
  mutate(win_ratio = wins/(wins+losses+draws))%>%
  mutate(fullbody = height + weight + reach)
```

```
All_fighter_career = All_fighter_career%>%
  filter(win_ratio>0 & win_ratio<1)

ggplot(All_fighter_career, aes(y = win_ratio, x = stance, fill = stance))+
  geom_boxplot()

ggplot(All_fighter_career, aes(x=fullbody,y=win_ratio))+
  geom_point()

ggplot(All_fighter_career, aes(x=per_head, y = win_ratio, color = stance))+
  geom_point()
```

```r
ggplot(All_fighter_career, aes(x=fullbody,y=win_ratio))+
  geom_point()

ggplot(All_fighter_career, aes(x=per_head, y = win_ratio, color = stance))+
  geom_point()

ggplot(All_fighter_career, aes(x=per_body, y = win_ratio))+
  geom_point()+ geom_jitter()

ggplot(All_fighter_career, aes(x=per_str, y = win_ratio))+
  geom_point()+ geom_jitter()


ggplot(All_fighter_career, aes(x=reach, y = win_ratio))+
  geom_point() + geom_jitter()

ggplot(All_fighter_career, aes(x=weight, y = win_ratio))+ xlim(100,325)+
  geom_point() + geom_jitter()

ggplot(All_fighter_career, aes(x=height, y = win_ratio))+
  geom_point() + geom_jitter()

strlm = lm(win_ratio~per_str, data=All_fighter_career)
summary(strlm)

All_fighter_career%>%
  filter(win_ratio>0 & win_ratio<1)%>%
  ggplot(aes(x=per_str,y=win_ratio,color="purple"))+
  geom_point()+geom_smooth(method="lm", se=FALSE, color="green")


headlm = lm(win_ratio~per_str,data = All_fighter_career)
summary(headlm)


All_fighter_career%>%
  filter(win_ratio>0 & win_ratio<1)%>%
  ggplot(aes(x=per_body,y=win_ratio, color="purple"))+
  geom_point(alpha=.5)+geom_smooth(method="lm", se=FALSE, color="green")

reachlm = lm(win_ratio~reach, data=All_fighter_career)
summary(reachlm)

All_fighter_career%>%
  filter(win_ratio>0 & win_ratio<1)%>%
  ggplot(aes(x=reach,y=win_ratio,color="purple"))+
  geom_point()+geom_smooth(method="lm", se=FALSE, color="green")

weightlm = lm(win_ratio~weight, data=All_fighter_career)
summary(weightlm)

All_fighter_career%>%
  filter(win_ratio>0 & win_ratio<1)%>%
  ggplot(aes(x=weight,y=win_ratio,color="purple"))+
  geom_point()+geom_smooth(method="lm", se=FALSE, color="green")
str_mult = lm(win_ratio~per_str+stance,data = All_fighter_career)
summary(str_mult)

agelm = lm(win_ratio~age, data=All_fighter_career)
summary(agelm)

All_fighter_career%>%
  filter(win_ratio>0 & win_ratio<1)%>%
  ggplot(aes(x=age,y=win_ratio,color="purple"))+
  geom_point()+geom_smooth(method="lm", se=FALSE, color="green")
```

```
str_mult = lm(win_ratio~per_str+stance,data = All_fighter_career)
summary(str_mult)


ggplot(All_fighter_career,aes(x=per_str,y=win_ratio, color = stance, alpha = .2)) +
  geom_point()+ geom_jitter() +
  geom_abline(intercept = str_mult$coefficients[1], slope=str_mult$coefficients[2],
              color="red", lwd=1)+
  geom_abline(intercept = str_mult$coefficients[1]+str_mult$coefficients[3], slope=str_mult$coefficients[2]+str_m
              color="yellow4", lwd=1)+
  geom_abline(intercept = str_mult$coefficients[1]+str_mult$coefficients[4], slope=str_mult$coefficients[2],
              color="green", lwd=1)+
  geom_abline(intercept = str_mult$coefficients[1]+str_mult$coefficients[5], slope=str_mult$coefficients[2]+str_m
              color="blue", lwd=1)+
  geom_abline(intercept = str_mult$coefficients[1]+str_mult$coefficients[6], slope=str_mult$coefficients[2]+str_m
              color="purple", lwd=1)

All_fighter_career%>%
  filter(win_ratio>0 & win_ratio<1)%>%
  filter(stance %in% c("Orthodox", "Southpaw", "Switch"))%>%
ggplot(aes(x=per_str,y=win_ratio, color = stance)) +
  geom_point()+ geom_jitter()+
  #geom_smooth(method="lm")+
  facet_grid(.~stance)+
  geom_smooth(method="lm", se=FALSE,
              color="black")

All_fighter_career%>%
  filter(weight_class %in% c("Heavyweight","LightHeavyweight", "Middleweight","Welterweight","Lightweight","Feath
  ggplot(aes(x=per_str,y=win_ratio, color = weight_class)) +
  geom_point()+ geom_jitter()+
  facet_grid(.~weight_class)+
  geom_smooth(method="lm", se=FALSE,
```

```
all_fight = All_fighter_career%>%
  filter(weight_class %in% c("Heavyweight","LightHeavyweight", "Middleweight","Welterweight"

model2 =lm(win_ratio~per_str+weight_class, data= all_fight)
summary(model2)

all_fightersw = All_fighter_career%>%
  filter(win_ratio>0 & win_ratio<1)%>%
  filter(stance %in% c("Orthodox", "Southpaw", "Switch"))

model1 =lm(win_ratio~reach+stance, data=all_fightersw)
summary(model1)
All_fighter_career%>%
  filter(win_ratio>0 & win_ratio<1)%>%
  filter(stance %in% c("Orthodox", "Southpaw", "Switch"))%>%
  ggplot(aes(x=stance,y=reach, fill = stance)) +
  geom_violin()

All_fighter_career%>%
  ggplot(aes(y=win_ratio, x=per, color=weig))+
  geom_point()

job =lm(win_ratio~per_str+per_head+per_body+age,data=All_fighter_career)
summary(job)
```