Introductory Investigations into the Sports Betting Market: NBA Case Study

Authors

We are a team of UC Berkeley undergraduates studying Data Science, Computer Science, Economics, and Business Administration. This project was conducted in Spring of 2019 within the Analytics Division of the Data Science Society @ Berkeley.

Role	Name	Contact
Project Manager	Chirasree Mandal	chirasreemandal@berkeley.edu
Research Analyst	Jhinuk Barman	jhinuk.barman@berkeley.edu
Research Analyst	Annie Cui	anniecui@berkeley.edu
Research Analyst	Dhruv Jhamb	dhruvjhamb@berkeley.edu
Research Analyst	Vinay Maruri	vmaruri1@berkeley.edu
Research Analyst	Lisa Zhou	lisazhou@berkeley.edu

Introduction

The market size of the United States sports betting market is estimated to somewhere between $67 billion to $150 billion [source]. The ambiguity of this figure is due in part to the prohibitions on sports betting in most states outside of Nevada and New Jersey. Many have voiced their opinions on this market's "untapped" potential, most notably NBA Commissioner Adam Silver. He argued that around $400 billion worth of illegal sports betting occurs in the United States through various channels, and that this market would be better regulated if sports betting were legalized. Although there is a large behavioral finance component to the

movement of the sports betting lines that is extremely hard to model, other components of sports betting line movement can be modeled like any other market. Using NBA betting data as a case study, we analyzed over 7 million NBA betting lines from the 2009 season to the present. Utilizing a variety of exploratory analysis and visualization techniques, we identified key features of the market that could be used to develop viable models of the NBA betting market. With these features, we developed three different linear models of the market and assessed their ability to model the market in real-world scenarios.

Glossary of Terms

Books: Sports books make money by accepting bets on a sports event.
Spread: Spread refers to a team winning or losing by a predetermined margin of points.
Lines: Lines refer to the odds set for a given game by a book.
Total: Refers to the expected total points for a game (sum of the favorite's expected score and the underdog's expected score).
Moneyline Probability: A system of odds analogous to American Odds for representing the implied probability of a win/loss outcome for a game.
Closing Line Value (CLV): Closing Line Value refers to the value of the bet relative to where the betting line closes. This is computed as the ratio of opening favorite and closing favorite probability minus 1.
Dog: In sports betting, "Dog" is the the team who is perceived to be as the underdog; the most likely to lose.
Favorite: The team who is perceived to be most likely to win.

Materials and Methods

Analysis was done using the pandas, sklearn, numpy, and matplotlib libraries in Python.

Quantifying Closing Line Value Movement

Dhruv Jhamb & Vinay Maruri

First, we examined closing line value movement. We calculated net line movement for Moneyline odds for sports books. Then, we plotted visualizations of net line movement for all the sports books together and then specifically 3 different books.

Formula for CLV:

$CLV = \frac{p_o}{p_c} - 1$

$p_o =$ opening favorite probability

$p_c =$ closing favorite probability

To calculate net line movement, we used the above equation.

	column_name	percent_missing
nba_game_id	nba_game_id	0.000000
timestamp_x	timestamp_x	0.000000
fave_nba_id_x	fave_nba_id_x	0.044276
dog_nba_id_x	dog_nba_id_x	0.044276
fave_ml_prob_x	fave_ml_prob_x	11.486562
dog_ml_prob_x	dog_ml_prob_x	12.073738
total_x	total_x	0.180415
total_over_prob_x	total_over_prob_x	0.201726
total_under_prob_x	total_under_prob_x	0.318830
ft_spread_x	ft_spread_x	0.136966
fave_ft_spread_prob_x	fave_ft_spread_prob_x	0.295243
dog_ft_spread_prob_x	dog_ft_spread_prob_x	0.150829
fh_spread_x	fh_spread_x	10.820557
fave_fh_spread_prob_x	fave_fh_spread_prob_x	10.820557
dog_fh_spread_prob_x	dog_fh_spread_prob_x	10.820557
book_name	book_name	0.000000
source_x	source_x	0.000000
sh_spread_x	sh_spread_x	23.676784
fave_sh_spread_prob_x	fave_sh_spread_prob_x	23.676784
dog_sh_spread_prob_x	dog_sh_spread_prob_x	23.676784
rn_x	rn_x	0.000000
timestamp_y	timestamp_y	0.000000
fave_nba_id_y	fave_nba_id_y	0.470486
dog_nba_id_y	dog_nba_id_y	0.470486
fave_ml_prob_y	fave_ml_prob_y	70.630211
dog_ml_prob_y	dog_ml_prob_y	70.809385
total_y	total_y	17.215360
total_over_prob_y	total_over_prob_y	17.222602
total_under_prob_y	total_under_prob_y	17.257774
ft_spread_y	ft_spread_y	1.918979
fave_ft_spread_prob_y	fave_ft_spread_prob_y	2.057807
dog_ft_spread_prob_y	dog_ft_spread_prob_y	1.922910
fh_spread_y	fh_spread_y	96.710322
fave_fh_spread_prob_y	fave_fh_spread_prob_y	96.710322
dog_fh_spread_prob_y	dog_fh_spread_prob_y	96.710322
source_y	source_y	0.000000
sh_spread_y	sh_spread_y	98.768957
fave_sh_spread_prob_y	fave_sh_spread_prob_y	98.768957
dog_sh_spread_prob_y	dog_sh_spread_prob_y	98.768957
rn_y	rn_y	0.000000
CLV_Movement	CLV_Movement	71.180560

Figure 1.0

We computed the proportion of NaNs in the data. We found that a large proportion, approximately 71.2%, of the computed closing line value movements were NaNs.
x corresponds to closing line data, y corresponds to opening line data.

In order to deal with NaN values, we employed 3 approaches. The first approach was to replace the NaN values with 0's. The second approach was to use sci-kit learn's pre_processing Imputer method to replace NaNs in the columns of our data frames with the mean of those corresponding columns. The third approach was to re-process our database of betting lines to check for the most recent opening lines and closing lines that were not NaN values.

We found that using the re-processed databases created the best, most stable results. We see that there is a significant amount of scatter in the data, and that

most CLV movements are between -1% against the favorite and 1% towards the favorite, with outliers above 2% movement towards the favorite.

Below are the plotted results of the line movement for all the sports books together using a scatter plot (right) and a kde plot (left).

Plots with the re-processed databases:

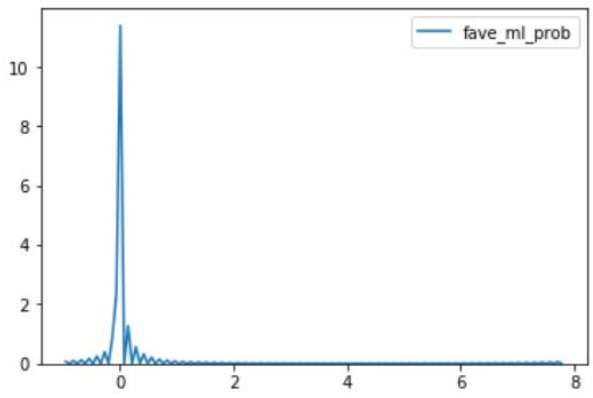

Figure 1.10

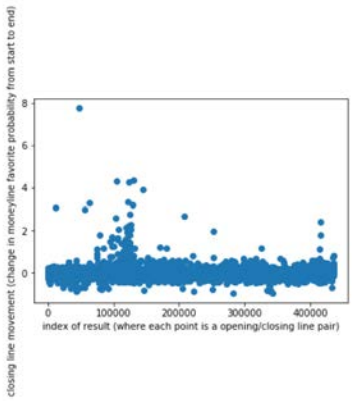

Figure 1.11

Now we can compute net line movement for individual sports books. We chose three books at random: 5dimes, MGM Mirage, and sportsbook.ag.

5dimes:

Plots with the re-processed databases:

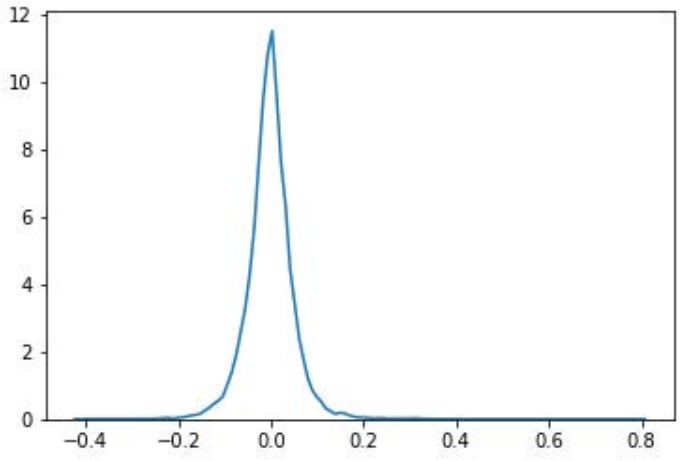

Figure 1.20

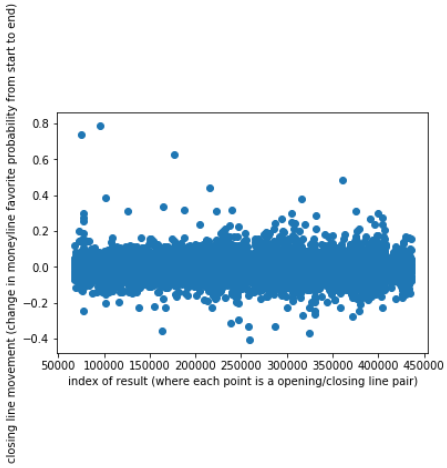

Figure 1.21

MGM Mirage:

Plots with the re-processed databases:

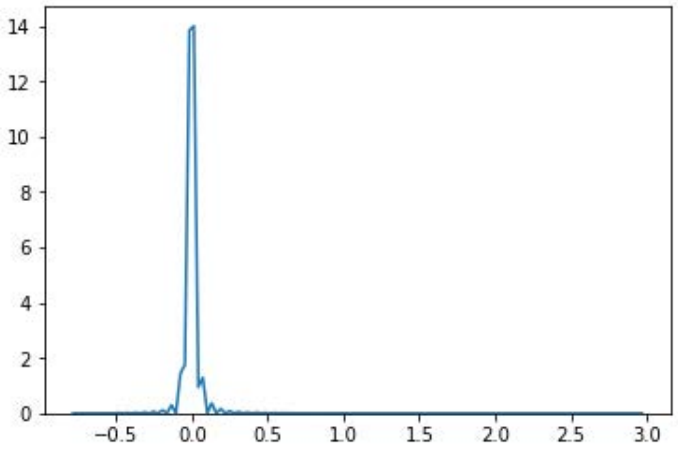

Figure 1.22

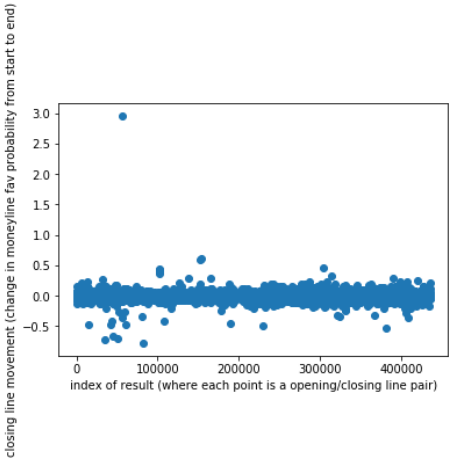

Figure 1.23

sportsbook.ag:

Plots with the re-processed databases:

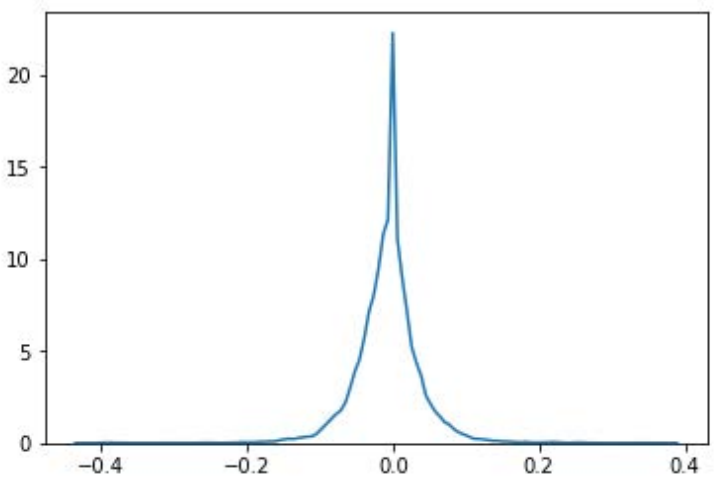

Figure 1.24

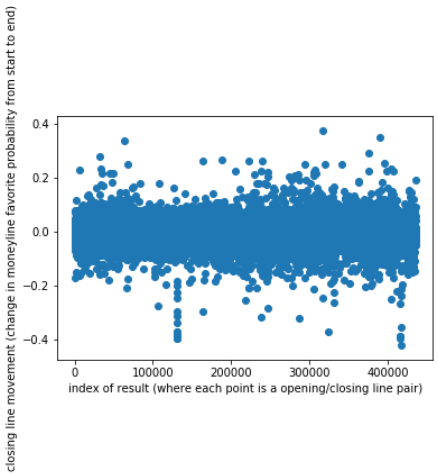

Figure 1.25

It appears that sportsbook.ag and 5dimes behave very similarly, while MGM Mirage is markedly different. 5dimes is based in Costa Rica, sportsbook.ag is based in Argentina, and MGM mirage is a casino in Las Vegas. For some reason, there are 2 sets of book trends: books that are based in Las Vegas or behave like Las Vegas books, and foreign books that are distinctly not modeling Las Vegas book behavior (either less or more aggressive).

Conclusion:

By examining the CLV movement for all the books, we see that the net line movement is clustered around 0. Even when picking specific books, this trend continues.

It is expected for the net line movement to be centered around 0 because although the books will move the lines to maintain a good betting split so they profit, the lines will not move drastically because the opening lines are set based on a variety of factors and are essentially an educated guess of a starting point that odds makers believe will get equal action from bettors on both sides. However, something that is more difficult is predicting which way the line will shift because while the data was clustered around 0, it was about equally spread between positive and negative closing line value.

One problem we faced with the data is dealing with NaNs. The best solution we had was to re-process the data to reduce the number of NaNs that we saw in opening betting line data. Doing so clusters the data around zero and exhibits a smoother, tighter distribution of CLV movements. We notice that the distribution of closing line value movement spreads out and becomes more varied to the point where the sportsbook.ag, mirage-mgm and 5dimes books CLV movements exhibit quasi-normal characteristics.

Further inquiry:

- What does net line movement mean in terms of market knowledge and trends?
- What sorts of factors affect CLV?

Seeking a Ranking of Sportsbooks

Annie Cui and Lisa Zhou

We wanted to find a way to rank the accuracy of each sports book. Our goal was to perform quantitative analysis as measures of accuracy. We used two metrics to rank each book: the mean-squared error on point spreads and the brier score of money line probabilities.

What is point spread?
Point spread if the the number of points by which an oddsmaker expects a favorite to defeat an underdog. For example, the "favorite" team (labeled with a "-" sign) would be at the disadvantage as they would need to win the game by a set number of points while the "underdog" team (labeled with a "+" sign) would be given an advantage to not lose the game by a set number of points.

1. **Quantitative Rankings Using the Brier Score and Mean Squared Error**
    a. **Mean-Squared Error**

To calculate the mean-squared error for each book, first we calculated the spread of each game by defining a function to consider whether it is the home or away team. Then with these new calculations, we compared them to the predicted spread values and squared the errors.

```
#Mean-Squared Error Calculations
# function to calculate spread
def spread(fave_id, home_team, home_score, away_score):
    if fave_id == home_team:
        return home_score - away_score
    else:
        return away_score - home_score
```

```
8
9 mse['mse'] = (mse['spread'] - mse['ft_spread']) ** 2
```

We calculated the mean squared error for each game and then grouped the result by book to determine the average mean squared error for each book. In the end, we were left with a ranking of the books based on their average mean-squared error on point spreads.

	mse
book_name	
grandsierra	199.813333
nine	241.876263
imperial	261.713327
elcortez	262.898605
wwts.com	263.850073
bet-jamaica	278.202863
bet365.com	278.740700
5dimes-reduced	279.752837
beted.com	285.447666
lucky's	286.222429

The top ten books based on mean-squared error ranking.

Additionally, we grouped the data by season to see how mean-squared error fluctuates by NBBA season.

	mse
season	
2007	270.217518
2008	328.422271
2009	307.113771
2010	398.093465
2011	278.388898
2012	295.658356
2013	2698.809587
2014	289.283365
2015	297.225748
2016	310.143099
2017	320.550031
2018	288.423872
2019	331.214528

The average mean-squared error by season

Looking at the average mean-squared error by season, we see that most of the seasons have a similar average error except for 2013 which has a score of 2698 likely due to a data error. Additionally, we don't see any trends in the seasons, there is no clear improvement in error as time increases.

To explore further, we also grouped by book and season.

		mse
book_name	**season**	
5dimes	**2011**	277.971374
	2012	295.105778
	2013	296.440213
	2014	288.465315
	2015	296.840961
	2016	309.356542
	2017	319.298568
	2018	288.861430
	2019	331.353427

Example of mean-squared error by season for the book 5dimes

b. **Brier Score**

Brier score is a way to verify the accuracy of a probability forecast. The score is between 0 and 1, with 0 indicating total accuracy and 1 meaning inaccurate. The formula to calculate brier score is $BS = \frac{1}{N}\sum_{t=1}^{N}(f_t - o_t)^2$.

For each game, we calculate its brier score using money line probability. Then we performed similar aggregation as with the mean-squared error by grouping by book and taking the average brier score.

```
# calculate brier score using formula
bs['bs'] = (bs['fave_ml_prob'] - bs['outcome']) ** 2
```

	bs
book_name	
bet365.com	0.183261
palmslv	0.190919
cal-neva	0.199989
bet-jamaica	0.200645
lvsc	0.201781
venetian	0.203642
intertops	0.205587
pinnaclesports	0.206946
5dimes-reduced	0.206948
gtbets.eu	0.207061

The top ten books based on brier score ranking.

Based on the calculations, the books had a very small range in brier score. One reason for this might be because books often change their money lines based on change from other books.

Then, we calculated the average brier score by season.

	bs
season	
2007	0.214871
2008	0.205631
2009	0.198278
2010	0.203322
2011	0.201087
2012	0.211221
2013	0.219015
2014	0.238455
2015	0.212516
2016	0.213830
2017	0.218944
2018	0.211807
2019	0.207513

Average brier score by NBA season

Similar to mean-squared error, we found the mean brier score for each book by season to investigate any trends in brier score.

		bs
book_name	**season**	
5dimes	**2011**	0.198319
	2012	0.208031
	2013	0.216406
	2014	0.231361
	2015	0.207472
	2016	0.208782
	2017	0.215363
	2018	0.208758
	2019	0.204742

Example of brier score by season for the book 5dimes

Comparing the Two Rankings

	mse
book_name	
grandsierra	199.813333
nine	241.876263
imperial	261.713327
elcortez	262.898605
wwts.com	263.850073
bet-jamaica	278.202863
bet365.com	278.740700
5dimes-reduced	279.752837
beted.com	285.447666
lucky's	286.222429
venetian	293.594246
cal-neva	293.702115
palmslv	296.292221
bet-phoenix	298.242007
vi-consensus	298.544030
wynn	298.548213
pinnaclesports	298.656945
bet-horizon	298.723270
southpoint	298.850284
sportbet.com	299.007098

The 20 rankings by average MSE

	bs
book_name	
bet365.com	0.183261
palmslv	0.190919
cal-neva	0.199989
bet-jamaica	0.200645
lvsc	0.201781
venetian	0.203642
intertops	0.205587
pinnaclesports	0.206946
5dimes-reduced	0.206948
gtbets.eu	0.207061
betus.com	0.207785
westgate-superbook	0.208300
sbg-global	0.208717
mirage-mgm	0.209879
bookmaker	0.210257
betdsi	0.210970
sportsbook.ag	0.211234
sportbet.com	0.211250
vi-consensus	0.211334
5dimes	0.211424

The 20 rankings by Brier Score

Comparing Quantitative Rankings

1. Qualitative Rankings from Online Reviews and Ratings: In order to gage the general public's sense and understanding for the world of sports betting, we look to social media and search engines.
    a. Environmental Factors
        i. seating, viewing, food/drinks, service

b. Tourists' Reviews
    i. reddit
    ii. ActionNetwork Relative Rankings
        1. Westgate SuperBook
        2. Wynn
        3. Caesars Palace
        4. Venetian
        5. Palazzo
        6. Bellagio
        7. South Point
        8. Red Rock
        9. Mirage
        10. Cosmopolitan

Conclusion
From our calculated BS values (Quantitative):
We can see that the lower the numerical value, the sharper the book. (the BS ranges between 0.1 to 0.3)
The Westgate, according to online sources, is a highly ranked sports-book. However, this book is only #12 based on our BS score calculations. According to online sources, the Wynn is ranked 2nd above Caesars Palace, however in out calculated BS Scores, the Wynn is #46 while the Caesar is #31. Both not in the top tier books. We can therefore conclude that quantitative observations and rankings do not correlate with qualitative rankings of sports books based on accuracy.

Further Questions to Investigate

- What other metrics could be used for ranking of books?
- Why are some of the top BS score sports-betting books not mentioned in popular reviews?
- How do the rankings of these books vary over time?

Quantifying the relationship between Spread, Total, and Moneyline Odds

Dhruv Jhamb and Jhinuk Barman

One approach to Exploratory Data Visualization consisted of randomly choosing 3 NBA teams based on their Team ID and making a scatterplot of Spread versus

Money Line Probability for each of those teams. We created scatterplots for when the teams were both favorite and underdog.

Point Spread vs. Moneyline Odds for three randomly chosen teams (as the favorite)

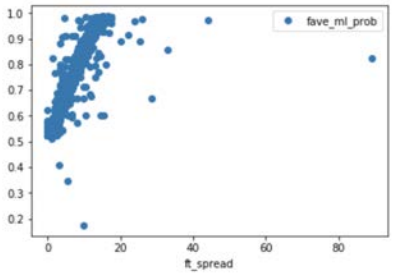

Figure 3.01: Team 1 ID - 1610612748

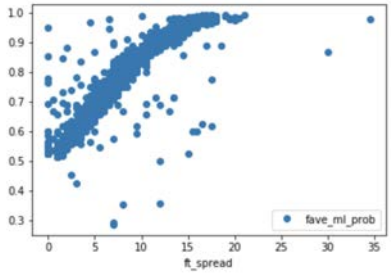

Figure 3.02: Team 2 ID - 1610612759

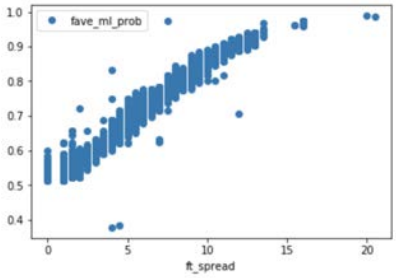

Figure 3.03: Team 3 ID - 1610612765

Point Spread vs. Moneyline Odds for three randomly chosen teams (as the underdog)

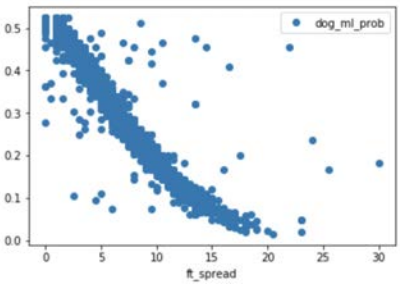

Figure 3.10: Team 1 ID - 1610612748

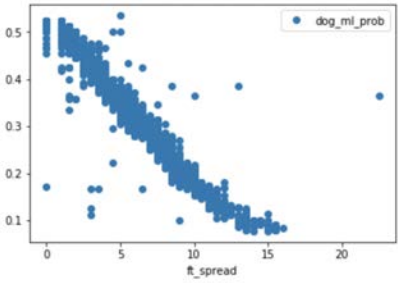

Figure 3.11: Team 2 ID - 1610612759

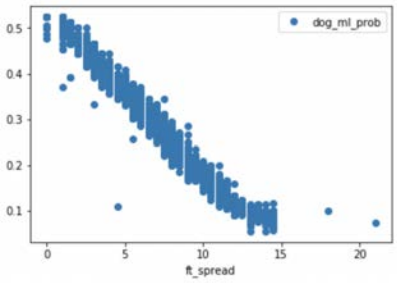

Figure 3.12: Team 3 ID - 1610612765

We can compute the value of a marginal point in terms of win probability (moneyline odds) by averaging the slopes of the graphs for the three teams.

```
#Linear Regression Calculation for Team 1
a2 = team_1_fav['fave_ml_prob'].values
a1 = team_1_fav['ft_spread'].values
mask = ~np.isnan(a1) & ~np.isnan(a2)
slope_1, intercept, r_value, p_value, std_err = linregress(a1[mask], a2[mask])
```

Using linear regression we can predict a linear model for each scatterplot:

$$y = \theta_0 + \theta_1 x$$

We used the linregress function from the SciPy library in Python to estimate the parameters for the slope (θ_1) and intercept (θ_0) of the scatterplots. Linregress calculates linear least-squares regression for 2 explanatory variables. We averaged the slopes for each of the 3 teams (when they were favorite) resulting in an average slope of 0.03261. We also found the slope of the teams when they were the underdog, which resulted in an average slope of -0.0268. The slope for underdog teams is negative because a marginal point is worth less in terms of win probability than favorite teams.

Conclusion:
We found that a marginal point is worth about 3.3% in terms of win probability from of average slope estimated to 0.033. The slope represents the increase in win probability for an increase in each marginal point.

Further inquiry:
Can we measure how sportsbooks move the ML in large/emotional sporting events as a response to the phenomenon in which individuals are biased to bet on the underdog?

Quantifying the value of a marginal point for all teams in the database

Chirasree Mandal

Using the methods outlined in the previous section, we generated a table of slopes for each team in the database when they're the dog and the favorite. The results were as follows.

	fave_nba_id	ml_vs_spread_fav	ml_vs_spread_dog
count	3.000000e+01	30.000000	30.000000
mean	1.610613e+09	0.031831	-0.031584
std	8.803408e+00	0.008492	0.008441
min	1.610613e+09	0.000086	-0.035693
25%	1.610613e+09	0.033165	-0.034406
50%	1.610613e+09	0.034091	-0.033839
75%	1.610613e+09	0.034780	-0.032911
max	1.610613e+09	0.035425	-0.000068

Figure 3.13: Table generated by using pd.DataFrame.describe() on a table with slopes for each team.

Analysis and Discussion: The average slope is ~3.2% for both the favorite and dog positions for the league. This is consistent with the previous findings.

Questions for Further Inquiry:

- The slopes and spread of slopes were very similar for when a team is in both the favorite and dog position. Is this to be expected? What can cause minor deviations?
- How does this value change with respect to line movement for a game?

Point Spread vs. Moneyline Odds vs Total for three randomly chosen teams (as the favorite)
Jhinuk Barman

Another form of Exploratory Data Analysis was generating a 3D scatterplot for spread and total versus Money Line Probability for 3 randomly chosen teams, where the team was a favorite.

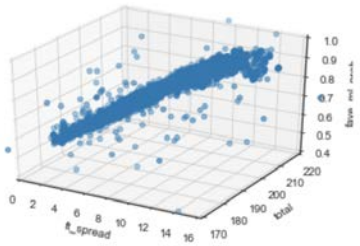

Figure 3.14 Team 1 ID - 1610612748

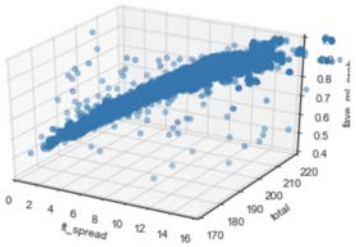

Figure 3.15 Team 2 ID - 1610612759

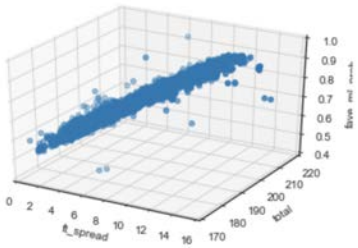

Figure 3.16 Team 3 ID - 1610612765

Analysis and Discussion: Using the 3D scatterplots with Money Line on the z-axis, we decided to calculate a the gradient of the surface, so we can find a general trend for each plot. Our approach was similar to our spread vs Money Line model but with the added explanatory variable of total. We attempted using the gradient function from the NumPy Library, but that resulted in multiple matrices and did not help us reach our result of finding a gradient per plot.

Conclusion: There seems to be a positive correlation between total and spread versus Money Line probability based on the trends of the scatterplot data. However, we could not arrive to a definite conclusion because we were not able to calculate the gradients for the plots.

Further inquiry:

- How can we calculate gradient from our visualization and what can we infer from the results?
- How can we interpret the scatterplot and approximate gradient calculation from the data?
- What are the other forms of representing the 3-dimensional data visually?

Comparing Models for predicting Moneyline Probabilities

Vinay Maruri and Dhruv Jhamb

Predicting ML (y) from point spread (x_1) and total points (x_2)

Our initial linear regression model involved a simple equation with just the features.

$$y = b_1 x_1 + b_2 x_2$$

However, intuitively we should not expect the model to go through the origin, as having the total points in a game be 0 is an impossible situation. Therefore, it is necessary to add a constant, making our model have a non-zero y-intercept. Here, c is the constant.

$$y = b_1 x_1 + b_2 x_2 + c$$

After finding that a standard linear regression model resulted in a low r^2 value implying a poor fit, we investigated possible methods to improve our model using these features. The solution we arrived at involved modifying the linear regression equation by first generating a degree 2 polynomial term from the point spread feature. Then, we noticed that the effect of the point spread feature on ML probability depended on the values of total points feature. Thus, we added an interaction term between point spread and total points to capture this effect and improve the fit of the model.

$$y = b_1 x_1 + b_2 x_1^2 + b_3 x_2 + b_4 x_1 x_2 + c$$

We will use three different metrics for evaluating models:

- Robustness
- Multi-Collinearity
- Goodness of Fit (r^2)

Multi-collinearity is the idea that one or more independent variables in our multiple regression model can be linearly predicted using other independent variables in the model. In other words, one or more independent variables in the model are linearly dependent on other variables. For detecting multicollinearity in the model, we are using the t-values of the individual variables in the model, as well as the f-statistic. If the probability of the t-values and the overall f-statistic are less than 0.05, we can confirm that multicollinearity is not present in the model. We do not expect point spread to be related to the total number of points in a game. This was confirmed when we ran our team-specific, book-specific, and general models. If we had detected multicollinearity in the model, then our model would experience high

variance in ML probability predictions for small changes in our input data, indicating a classic sign of overfitting.

We would also have seen numerical issues in our regression model, as our algorithm solver would not been able to arrive at the optimal coefficient estimates for each of the independent variables and the constant term. This is because statistical packages solve regression problems using linear algebra. In our model, the computer would be trying to solve the problem:

$$y = \beta X + \epsilon$$

Where β is the matrix of estimated regression coefficients, X is the design matrix containing independent variable values for all of our data points, y is a vector of ML probability values, and ϵ is the constant vector. Notice that the optimal estimate of β is:

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

Thus, in order to solve for the optimal β, $X^T X$ must be invertible. When multicollinearity is detected in our model, the design matrix X will not be full rank, thus making $X^T X$ non-invertible. Hence the optimal β cannot be solved for and our regression problem is unsolvable.

To describe the goodness of fit for the model, we will be checking the r^2 and adjusted r^2 values. These statistics measure the amount of variance in the dependent variable that is captured by the independent variables. We notice that for the regressions we ran, r^2 and adjusted r^2 values are equal, meaning that the independent variables that we've selected are truly explaining variance in the dependent variable, and the model is doing much better than chance.

Finally, we need to check the robustness of our model. This is to test how regression coefficient estimates behave and whether our general model design holds if we change a few assumptions. We discuss this further when we introduce the book-specific model for the MGM Mirage.

Team specific models:

First, we will examine the model for the team corresponding to the ID: 1610612748.
Model: $y = 0.0253x_1 - 0.0005x_1^2 - 0.0008x_2 + 8.013 * 10^{-5}x_1x_2 + 0.6428$

OLS Regression Results

Dep. Variable:	fave_ml_prob	R-squared:	0.963
Model:	OLS	Adj. R-squared:	0.963
Method:	Least Squares	F-statistic:	1.151e+05
Date:	Thu, 02 May 2019	Prob (F-statistic):	0.00
Time:	22:05:45	Log-Likelihood:	42783.
No. Observations:	17555	AIC:	-8.556e+04
Df Residuals:	17550	BIC:	-8.552e+04
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.6428	0.007	98.651	0.000	0.630	0.656
ft_spread	0.0253	0.001	25.827	0.000	0.023	0.027
np.power(ft_spread, 2)	-0.0005	3.22e-06	-149.583	0.000	-0.000	-0.000
total	-0.0008	3.31e-05	-23.249	0.000	-0.001	-0.001
ft_spread:total	8.013e-05	5.01e-06	16.005	0.000	7.03e-05	8.99e-05

Omnibus:	19553.290	Durbin-Watson:	1.419
Prob(Omnibus):	0.000	Jarque-Bera (JB):	18796069.191
Skew:	-4.868	Prob(JB):	0.00
Kurtosis:	163.006	Cond. No.	5.89e+04

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 5.89e+04. This might indicate that there are
strong multicollinearity or other numerical problems.

We find that the team's linear regression model has a high r^2 value (0.914) indicating that the model has a good fit. The F-statistic ($9.335 * 10^4$) is large with a probability 0 of being seen. Combined with the individual variable t-statistic scores, this means that there is no multicollinearity between the features of the model. Now that we have seen that a linear regression model with good fit and non-multicollinear features can be created for a specific team, we will continue this process for other teams.

Second, we will examine the model for the team corresponding to the ID: 1610612741.
Model: $y = 0.0478x_1 - 0.0008x_1^2 - 9.11 * 10^{-5}x_2 - 1.367 * 10^{-5}x_1x_2 + 0.4955$

team: 1610612741

OLS Regression Results

Dep. Variable:	fave_ml_prob	R-squared:	0.976
Model:	OLS	Adj. R-squared:	0.976
Method:	Least Squares	F-statistic:	1.396e+05
Date:	Fri, 26 Apr 2019	Prob (F-statistic):	0.00
Time:	14:18:51	Log-Likelihood:	37286.
No. Observations:	13948	AIC:	-7.456e+04
Df Residuals:	13943	BIC:	-7.452e+04
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.4955	0.005	97.837	0.000	0.486	0.505
ft_spread	0.0478	0.001	56.073	0.000	0.046	0.049
np.power(ft_spread, 2)	-0.0008	1.27e-05	-59.716	0.000	-0.001	-0.001
total	-9.11e-05	2.59e-05	-3.523	0.000	-0.000	-4.04e-05
ft_spread:total	-1.367e-05	4.22e-06	-3.240	0.001	-2.19e-05	-5.4e-06

Omnibus:	12372.280	Durbin-Watson:	1.454
Prob(Omnibus):	0.000	Jarque-Bera (JB):	16774023.445
Skew:	3.036	Prob(JB):	0.00
Kurtosis:	172.782	Cond. No.	4.76e+04

We find again that this team's linear regression model has a high r^2 value indicating that the model has a good fit. The F-statistic is very large, and given the t-score probabilities for each of the individual variables is approximately zero, we can conclude that this team specific model has a good fit to the data and does not suffer from multicollinearity.

Third, we will examine the model for the team corresponding to the ID: 161012754.
Model: $y = 0.0486x_1 - 0.0008x_1^2 - 0.0001x_2 - 2.11 * 10^{-5}x_1x_2 + 0.5048$

team: 1610612754

OLS Regression Results

Dep. Variable:	fave_ml_prob	R-squared:	0.974
Model:	OLS	Adj. R-squared:	0.974
Method:	Least Squares	F-statistic:	1.575e+05
Date:	Fri, 26 Apr 2019	Prob (F-statistic):	0.00
Time:	14:18:51	Log-Likelihood:	43649.
No. Observations:	16581	AIC:	-8.729e+04
Df Residuals:	16576	BIC:	-8.725e+04
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.5048	0.004	118.271	0.000	0.496	0.513
ft_spread	0.0486	0.001	68.299	0.000	0.047	0.050
np.power(ft_spread, 2)	-0.0008	9.73e-06	-81.099	0.000	-0.001	-0.001
total	-0.0001	2.12e-05	-5.558	0.000	-0.000	-7.64e-05
ft_spread:total	-2.11e-05	3.5e-06	-6.033	0.000	-2.8e-05	-1.42e-05

Omnibus:	12818.057	Durbin-Watson:	1.470
Prob(Omnibus):	0.000	Jarque-Bera (JB):	12091697.270
Skew:	-2.403	Prob(JB):	0.00
Kurtosis:	135.208	Cond. No.	4.12e+04

We find again that this team's linear regression model has a high r^2 value indicating that the model has a good fit. The F-statistic is very large, and given the t-score probabilities for each of the individual variables is approximately zero, we can conclude that this team specific model has a good fit to the data and does not suffer from multicollinearity.

Fourth, we will examine the model for team corresponding to the ID: 1610612738.
Model: $y =$
$0.0443x_1 - 0.0003x_1^2 - 9.384 * 10^{-5}x_2 - 3.021 * 10^{-5}x_1x_2 + 0.5158$

team: 1610612738

OLS Regression Results

Dep. Variable:	fave_ml_prob	R-squared:	0.963
Model:	OLS	Adj. R-squared:	0.963
Method:	Least Squares	F-statistic:	1.077e+05
Date:	Fri, 26 Apr 2019	Prob (F-statistic):	0.00
Time:	14:18:51	Log-Likelihood:	40827.
No. Observations:	16484	AIC:	-8.164e+04
Df Residuals:	16479	BIC:	-8.161e+04
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.5158	0.006	88.743	0.000	0.504	0.527
ft_spread	0.0443	0.001	52.253	0.000	0.043	0.046
np.power(ft_spread, 2)	-0.0003	2.74e-06	-106.291	0.000	-0.000	-0.000
total	-9.384e-05	2.85e-05	-3.294	0.001	-0.000	-3.8e-05
ft_spread:total	-3.021e-05	4.04e-06	-7.478	0.000	-3.81e-05	-2.23e-05

Omnibus:	15833.578	Durbin-Watson:	1.398
Prob(Omnibus):	0.000	Jarque-Bera (JB):	30903755.126
Skew:	-3.459	Prob(JB):	0.00
Kurtosis:	215.006	Cond. No.	5.19e+04

We find again that this team's linear regression model has a high r^2 value indicating that the model has a good fit. The F-statistic is very large, and given the t-score probabilities for each of the individual variables is approximately zero, we can conclude that this team specific model has a good fit to the data and does not suffer from multicollinearity.

Book Specific Models:

Similarly, we created specific models for each book.

First, we will examine the model for the sports book Stations.
Model: $0.0622x_1 - 2.33 * 10^{-5}x_1^2 + 0.0006x_2 - 0.0001x_1x_2 + 0.3901$

Book name: stations

OLS Regression Results

Dep. Variable:	fave_ml_prob	R-squared:	0.969
Model:	OLS	Adj. R-squared:	0.969
Method:	Least Squares	F-statistic:	1.007e+05
Date:	Thu, 02 May 2019	Prob (F-statistic):	0.00
Time:	22:18:10	Log-Likelihood:	33202.
No. Observations:	12907	AIC:	-6.639e+04
Df Residuals:	12902	BIC:	-6.636e+04
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.3901	0.006	68.688	0.000	0.379	0.401
ft_spread	0.0622	0.001	70.465	0.000	0.060	0.064
np.power(ft_spread, 2)	-2.33e-05	4.79e-08	-486.258	0.000	-2.34e-05	-2.32e-05
total	0.0006	2.78e-05	20.027	0.000	0.001	0.001
ft_spread:total	-0.0001	4.27e-06	-30.722	0.000	-0.000	-0.000

Omnibus:	3845.390	Durbin-Watson:	1.934
Prob(Omnibus):	0.000	Jarque-Bera (JB):	206140.985
Skew:	0.631	Prob(JB):	0.00
Kurtosis:	22.538	Cond. No.	7.67e+05

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 7.67e+05. This might indicate that there are
strong multicollinearity or other numerical problems.

We find that Station's linear regression model has a high r^2 value indicating that the model has a good fit. The F-statistic is very large, and given the t-score probabilities for each of the individual variables is approximately zero, we can conclude that this model has a good fit to the data and does not suffer from multicollinearity.

Second, we will examine the model for the sports book BetGrande.
Model: $0.0011x_1 - 0.0002x_1^2 - 0.0010x_2 + 0.0002x_1x_2 + 0.6973$

Book name: betgrande

OLS Regression Results

Dep. Variable:	fave_ml_prob	R-squared:	0.974
Model:	OLS	Adj. R-squared:	0.974
Method:	Least Squares	F-statistic:	9.293e+04
Date:	Thu, 02 May 2019	Prob (F-statistic):	0.00
Time:	22:24:53	Log-Likelihood:	28274.
No. Observations:	10109	AIC:	-5.654e+04
Df Residuals:	10104	BIC:	-5.650e+04
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.6973	0.003	247.305	0.000	0.692	0.703
ft_spread	0.0011	0.000	4.132	0.000	0.001	0.002
np.power(ft_spread, 2)	-0.0002	3.44e-07	-440.098	0.000	-0.000	-0.000
total	-0.0010	1.4e-05	-70.921	0.000	-0.001	-0.001
ft_spread:total	0.0002	1.3e-06	136.851	0.000	0.000	0.000

Omnibus:	4974.111	Durbin-Watson:	1.932
Prob(Omnibus):	0.000	Jarque-Bera (JB):	491382.882
Skew:	-1.426	Prob(JB):	0.00
Kurtosis:	37.036	Cond. No.	2.52e+04

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.52e+04. This might indicate that there are strong multicollinearity or other numerical problems.

We find that BetGrande's linear regression model has a high r^2 value indicating that the model has a good fit. The F-statistic is very large, and given the t-score probabilities for each of the individual variables is approximately zero, we can conclude that this model has a good fit to the data and does not suffer from multicollinearity.

Third, we will examine the model for the sports book Catalina.
Model: $0.0911x_1 - 3.423 * 10^{-6}x_1^2 + 0.0017x_2 - 0.0003x_1x_2 + 0.1780$

```
Book name: catalina
                            OLS Regression Results
==============================================================================
Dep. Variable:           fave_ml_prob   R-squared:                       0.936
Model:                            OLS   Adj. R-squared:                  0.936
Method:                 Least Squares   F-statistic:                 3.142e+04
Date:                Thu, 02 May 2019   Prob (F-statistic):               0.00
Time:                        22:25:19   Log-Likelihood:                 17984.
No. Observations:                8609   AIC:                         -3.596e+04
Df Residuals:                    8604   BIC:                         -3.592e+04
Df Model:                           4
Covariance Type:            nonrobust
================================================================================================
                                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
Intercept                     0.1780      0.011     15.601      0.000       0.156       0.200
ft_spread                     0.0911      0.002     57.446      0.000       0.088       0.094
np.power(ft_spread, 2) -3.423e-06   1.12e-08   -304.365      0.000   -3.45e-06    -3.4e-06
total                         0.0017   5.66e-05     29.906      0.000       0.002       0.002
ft_spread:total              -0.0003   7.79e-06    -37.496      0.000      -0.000      -0.000
==============================================================================
Omnibus:                    25352.258   Durbin-Watson:                   1.998
Prob(Omnibus):                  0.000   Jarque-Bera (JB):       3009518708.422
Skew:                         -41.084   Prob(JB):                         0.00
Kurtosis:                    2898.363   Cond. No.                     3.84e+07
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.84e+07. This might indicate that there are
strong multicollinearity or other numerical problems.
```

We find that Catalina's linear regression model has a high r^2 value indicating that the model has a good fit. The F-statistic is very large, and given the t-score probabilities for each of the individual variables is approximately zero, we can conclude that this model has a good fit to the data and does not suffer from multicollinearity.

Fourth, we will examine the model for the sports book Mirage-MGM.
Model:
$0.0410x_1 - 3.542 * 10^{-6}x_1^2 - 1.632 * 10^{-5}x_2 - 2.726 * 10^{-5}x_1x_2 + 0.5063$

```
Book name: mirage-mgm
                            OLS Regression Results
==============================================================================
Dep. Variable:           fave_ml_prob   R-squared:                       0.956
Model:                            OLS   Adj. R-squared:                  0.956
Method:                 Least Squares   F-statistic:                 7.053e+04
Date:                Thu, 02 May 2019   Prob (F-statistic):               0.00
Time:                        22:16:59   Log-Likelihood:                 31901.
No. Observations:               12967   AIC:                        -6.379e+04
Df Residuals:                   12962   BIC:                        -6.375e+04
Df Model:                           4
Covariance Type:            nonrobust
================================================================================================
                                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
Intercept                      0.5063      0.007     74.508      0.000       0.493       0.520
ft_spread                      0.0410      0.001     40.030      0.000       0.039       0.043
np.power(ft_spread, 2) -3.542e-06   6.69e-09   -529.052      0.000   -3.55e-06   -3.53e-06
total                  -1.632e-05   3.36e-05     -0.486      0.627   -8.21e-05    4.95e-05
ft_spread:total        -2.726e-05   5.04e-06     -5.408      0.000   -3.71e-05   -1.74e-05
==============================================================================
Omnibus:                    11035.855   Durbin-Watson:                   1.929
Prob(Omnibus):                  0.000   Jarque-Bera (JB):         17814734.281
Skew:                          -2.757   Prob(JB):                         0.00
Kurtosis:                     184.499   Cond. No.                     3.32e+07
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.32e+07. This might indicate that there are
strong multicollinearity or other numerical problems.
```

We find that Mirage-MGM's linear regression model has a high r^2 value indicating that the model has a good fit. The F-statistic is very large, but t-score probabilities for the total points variable is 0.627, which is much greater than the accepted value of 0.05, meaning that the features suffer from multicollinearity, making it not a robust model. The issue is that the feature for total points is linearly dependent on some combination of the three other variables (i.e. total points is a linear combination of the three other variables). This could thus be causing the F-statistic to be artificially high and imply false statistical significance for the variables jointly.

At this point, we decided to do a robustness check of our overall model design since MGM Mirage was the first specific model where we encountered multicollinearity issues.
To do this, we decided to remove the total variable from our MGM model, giving us:
Model: $0.0414x_1 - 3.542 * 10^{-6}x_1^2 - 2.947 * 10^{-5}x_1x_2 + 0.5030$

```
Book name: mirage-mgm
                            OLS Regression Results
==============================================================================
Dep. Variable:           fave_ml_prob   R-squared:                       0.956
Model:                            OLS   Adj. R-squared:                  0.956
Method:                 Least Squares   F-statistic:                 9.404e+04
Date:                Thu, 02 May 2019   Prob (F-statistic):               0.00
Time:                        23:26:39   Log-Likelihood:                 31900.
No. Observations:               12967   AIC:                        -6.379e+04
Df Residuals:                   12963   BIC:                        -6.376e+04
Df Model:                           3
Covariance Type:            nonrobust
================================================================================================
                                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
Intercept                    0.5030      0.000   1124.939      0.000       0.502       0.504
ft_spread                    0.0414      0.000     91.459      0.000       0.041       0.042
np.power(ft_spread, 2) -3.542e-06   6.67e-09   -530.774      0.000   -3.55e-06   -3.53e-06
ft_spread:total        -2.947e-05   2.19e-06    -13.440      0.000   -3.38e-05   -2.52e-05
==============================================================================
Omnibus:                    11030.021   Durbin-Watson:                   1.929
Prob(Omnibus):                  0.000   Jarque-Bera (JB):         17822784.529
Skew:                          -2.754   Prob(JB):                         0.00
Kurtosis:                     184.541   Cond. No.                     2.39e+06
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.39e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
```

The updated model has a large r^2 value, indicating a good fit. The F-statistic is very large and the given t-score probability for each variables are now 0, so the model does not suffer from multicollinearity. What we also noticed was that the regression coefficient estimates for the remaining variables and intercept term do not vary that much from the original four variable and intercept design for MGM Mirage.

General model:

We created a general linear regression model using data from all the sports books and teams.
Model $y = 0.0071x_1 - 2.38 * 10^{-6}x_1^2 - 0.0005x_2 + 8.367 * 10^{-5}x_1x_2 + 0.6635$

```
                            OLS Regression Results
==============================================================================
Dep. Variable:           fave_ml_prob   R-squared:                       0.693
Model:                            OLS   Adj. R-squared:                  0.693
Method:                 Least Squares   F-statistic:                 2.456e+05
Date:                Thu, 02 May 2019   Prob (F-statistic):               0.00
Time:                        21:59:54   Log-Likelihood:             5.9857e+05
No. Observations:              435739   AIC:                        -1.197e+06
Df Residuals:                  435734   BIC:                        -1.197e+06
Df Model:                           4
Covariance Type:            nonrobust
================================================================================================
                                   coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
Intercept                       0.6635      0.002    375.009      0.000       0.660       0.667
ft_spread                       0.0071      0.000     40.674      0.000       0.007       0.007
np.power(ft_spread, 2)      -2.38e-06   2.49e-09   -955.409      0.000   -2.38e-06   -2.37e-06
total                          -0.0005   8.74e-06    -55.190      0.000      -0.000      -0.000
ft_spread:total             8.367e-05   8.59e-07     97.459      0.000    8.2e-05    8.54e-05
==============================================================================
Omnibus:                  2367889.427   Durbin-Watson:                   1.388
Prob(Omnibus):                  0.000   Jarque-Bera (JB):   381492435538574.500
Skew:                        -290.948   Prob(JB):                         0.00
Kurtosis:                  144957.556   Cond. No.                     4.09e+06
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.09e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In this case, we find that the general model has a slightly worse fit, as the r^2 value is lower than the r^2 values for the team/book-specific models. However, this is expected as the model combines specific sources of data into a general source, so we can expect a lower quality of fit. The F-statistic is very large, and given the t-score probabilities for each of the individual variables is approximately zero, we can conclude that this model has a good fit to the data and does not suffer from multicollinearity.

Conclusion

Our regression model is a robust predictor of ML probability that does not suffer from multicollinearity. All results presented here are statistically significant, as indicated by the analysis on the F-statistic and t-score probabilities for each model. While the goodness of fit suffers slightly in the general case, for the team-specific and book-specific models, we have strong models that detect upwards of 95% of the variation in ML probability using our regression model's design.

Featurizing books to predict moneyline probabilities

Predicting ML using Ordinary Least Squares regression

Chirasree Mandal & Annie Cui

Our model for linear regression follows this least squares regression equation:
$y = ax_1 + bx_2 + cx_3 + dx_4 + ex_5 + c$

Following a similar theoretical model to that of predicting moneyline probability (y) where
x_1 is CLV movement
x_2 is Geographic Location
- binary value (1: Vegas, 0: Offshore)

x_3 is Accuracy
- Brier Score (unit-less)

x_4 is Fav/Dog
x_5 is Home
y is ML prob

The CLV movement is the calculation for the net line movement from the closing line values of each game. Refer here. We decided that this feature is an important factor because it's an important dynamic feature that measures the actions happening during the game.
The Geographic location accounts for whether or not the sports-book is onshore (located in Vegas) or offshore. This is an important factor to account for because onshore books tend to have stricter, more restrictive betting guidelines than offshore books, and therefore quantifiably distinct behavior.

Accuracy, measured by the brier score, calculates the relative winning bets made in each book. Refer here. This feature is an important factor for determining which book to choose if you want to maximize your chance winning bets.

The tutorial we referenced to develop our model was from an 'Example of Multiple Linear Regression in Python'.
The linear fits and statistical calculations we use in our model are from the statsmodel API.

Fitting the Model:

```
1                            OLS Regression Results
```

```
==============================================================================
Dep. Variable:                     ml   R-squared:                       0.988
Model:                            OLS   Adj. R-squared:                  0.987
Method:                 Least Squares   F-statistic:                     3503.
Date:                Fri, 26 Apr 2019   Prob (F-statistic):          1.71e-207
Time:                        15:53:49   Log-Likelihood:                 552.10
No. Observations:                 226   AIC:                            -1092.
Df Residuals:                     220   BIC:                            -1072.
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.2916      0.014     21.563      0.000       0.265       0.318
bs             0.2593      0.057      4.541      0.000       0.147       0.372
fav            0.3518      0.006     61.088      0.000       0.340       0.363
```

```
home            0.0054      0.003       1.777      0.077      -0.0
01        0.011
CLV            -0.7533      0.165      -4.553      0.000      -1.0
79       -0.427
vegas          -0.0003      0.003      -0.114      0.910      -0.0
06        0.005
==============================================================
==============
Omnibus:                      280.871   Durbin-Watson:
         1.388
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
      21595.969
Skew:                           5.152   Prob(JB):
           0.00
Kurtosis:                      49.768   Cond. No.
           157.
==============================================================
==============
```

Optimizing the Model:
After our first run, we wanted to test some factors to see if we could improve our model. We considered dropping the constant and the resulting stats are shown below. In order to make the decision to drop the constant, we must first consider if any data points exist at the origin. At the origin, CLV movement is 0, Geographic location is Offshore, brier score is 0 (perfectly accurate), the team is a dog, and the team is away. Therefore, it is acceptable for us to fit the model without adding a constant. This however proved a worse outcome than our first run.

```
                            OLS Regression Results

==============================================================
==============
Dep. Variable:                     ml   R-squared:
         0.996
```

```
Model:                            OLS   Adj. R-squared:
         0.995
Method:                 Least Squares   F-statistic:
        9842.
Date:                Fri, 26 Apr 2019   Prob (F-statistic):
     2.07e-257
Time:                        15:56:48   Log-Likelihood:
        423.76
No. Observations:                 226   AIC:
       -837.5
Df Residuals:                     221   BIC:
       -820.4
Df Model:                           5

Covariance Type:            nonrobust

==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
bs             1.4262      0.032     44.454      0.000       1.363       1.489
fav            0.3913      0.010     40.709      0.000       0.372       0.410
home           0.0051      0.005      0.949      0.343      -0.006       0.016
CLV            0.3566      0.277      1.288      0.199      -0.189       0.902
vegas          0.0077      0.005      1.524      0.129      -0.002       0.018
```

```
==============================================================================
Omnibus:                      107.165   Durbin-Watson:                   1.827
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             4610.382
Skew:                          -1.048   Prob(JB):                         0.00
Kurtosis:                      25.027   Cond. No.                         108.
==============================================================================
```

The OLS Linear Model

$$y = 1.4262x_1 + 0.3913x_2 + 0.0051x_3 + 0.3566x_4 + 0.0077x_5$$

x_1 is average CLV movement
x_2 is Geographic Location

- binary value (1: Vegas, 0: Offshore)

x_3 is Accuracy

- Brier Score (unit-less)

x_4 is Fav/Dog
x_5 is Home
y is closing ML probability

Further Explanations:

- the purpose of a constant is explained in the first model. Refer here.
- the meaning of F-stats : A value you get when you run a regression analysis on a data set.
  - Generally, if your f-stat is larger than your calculated F-value, you can reject the null hypothesis.
- the meaning for r-squared: The fraction of the variation in the independent variable that is accounted for by independent variables.

Prediction Using Stochastic Gradient Descent

Lisa Zhou and Vinay Maruri

Another approach that we thought about was implementing a machine learning technique. We decided to use gradient descent to minimize an empirical loss function to learn our model's weights. In this case, we tried minimizing the mean squared error function using L1, L2, and elastic net regularization, as well as minimizing the Huber loss function with L1, L2, and elastic net regularization.

L1 regularization adds an extra term to the loss function that is the sum of the absolute value of model weights. L2 regularization adds an extra term to the loss function that is the sum of squares of the model weights. Elastic net regularization adds both the sum of squares of model weights and the sum of the absolute value of the model weights to the loss function, with each term having different regularization parameters (λ).

Formally, we state the gradient descent problem as follows:

$w^{t+1} = w^t - \alpha \frac{\partial}{\partial w} L(X, w)$

$min_w L(X, w) = (y - X_i w)^2$ + regularization term (mean squared error)

$min_w L(X, w) = if(\mid X_i \mid \leq \sigma) 0.5 X_i^2, else \sigma(\mid X_i \mid + 0.5\sigma)$ + regularization term

where X_i is a randomly selected point, y are the true y values, w is vector of independent variable weights, X is the independent variable data matrix, and σ is a hyperparameter determining a cuttoff for what points are considered outliers and what points are not.

Regularization terms that are added to the loss function (λ is the regularization hyperparameter):

$\lambda \sum_{w \in W} \mid w \mid$ (L1 Regularization)

$\lambda \sum_{w \in W} w^2$ (L2 Regularization)

$\lambda_1 \sum_{w \in W} \mid w \mid + \lambda_2 \sum_{w \in W} w^2$ (Elastic Net Regularization)

Noting the large size of our database, we decided to use stochastic gradient descent instead of batch gradient descent to speed up run time in training our linear regression model, since stochastic computes the gradient of the loss function at one

point, whereas batch computes the gradient of the loss function at every possible point in our data.

Stochastic gradient descent works because as we compute the loss using our convex loss function, we take the gradient of the loss function at a randomly selected point and use that to update our model parameters, in this case the coefficients of model's independent variables. If we run this algorithm a sufficient number of iterations, we should converge to the optimal set of model variable weights that minimize the empirical loss function. We verified that the loss functions we were working with were convex, so we proceed with this model.

Using scikit learn's linear model SGDRegressor, we used the same features as in the OLS model. So, we will be using book name, average brier score, fav/dog, home/away, and Vegas/offshore to predict closing money line probability.

After getting the data into the correct format, we will split the data into a training and test set. We will use 80% of the data for the training set and reserve 20% of the data for the test set.

Initially, we started fitting our model using squared loss, but this resulted in a poor fit as indicated by a negative r^2 value. This was likely due to the fact that our model was placing too much emphasis on minimizing the loss with respect to outlier points, a known problem with minimizing mean square error functions. To remedy this, we switched to minimizing the Huber loss function, which places less weight on outlier points.

Using the Huber loss function, we also experimented with various types of regularization functions. We evaluated the models using the provided score function, which computes the coefficient R^2 of the model's prediction, defined as: $1 - (u/v)$, where u is the sum of squared differences between the true values and the model's predicted values and v is the sum of squared differences between the true values and the mean of true values. In essence, we are testing how good the model is against a model that predicts a constant, the mean of the true values. This is meant to test how good our model is above randomly guessing. The best possible value is 1, indicating "perfect" predictions, the worst possible value is -1, indicating arbitrarily worse predictions, and 0 indicates that the model is no better than

constantly predicting the mean of the true values. The score we refer to below is the coefficient R^2.

Using L1 regularization, the score on the training set was 0.9819197795369871 and the score on the test set was 0.9977314074063863. With L2 regularization, the score on the training set was 0.9798374551370834 and the score on the test set was 0.9965117814980099. With elastic net, the score on the training set was 0.982597071984506 and the score on the test set was 0.9974592529468738.

Thus, it seems like using Huber loss is a good fit for the model. There does not seem to be much difference in goodness of fit for the model between the three types of regularization methods.

Running Real-life Simulation with both Models

In order to use our constructed models to determine investment strategy for sports books, we will run our models with their respective features and observe the predicted closing favorite Moneyline odds. Given that we will invest everything into a single book, we will determine which book has the highest likelihood to result in profit by investing based on which book has the maximum Δ between the book's favorite Moneyline probability and the model's predicted Moneyline probability. This exploits the elasticity of books to make money by beating the closing line value.

To test our model's profit capability, we will set up control and experiment strategies. The control strategy will be picking the best priced book and placing the total value of the bet on that book. The best priced book in this case will be the book that offers the most favorable odds on a given game. The experiment strategy will be picking a book with the maximum Δ from the model. Then we will compare the expected value from both the control and experiment betting strategies.

Expected value is defined as:

$$p_f * a_w - (1 - p_f) * a_l$$

$p_f =$ probability of winning
$a_w =$ amount you could win per bet

a_l = amount you could lose per bet

Conclusion

From our analysis of the "untapped" market of sports betting, we have developed three different models to model the market to determine the most profitable investment. When examining the specific feature CLV movement, we found that the net line movement was clustered around 0. This was expected as it is rare for sports books to drastically shift lines since sportsbooks set lines based on a variety of factors. The feature Brier Score tells us the sharpness of a book, with a range of $[0.1, 0.3]$. In determining the specific value of a point in the feature point spread, we calculated the average slope for a linear model the maps point spread to ML odds. We found that a marginal point is worth about 3.3% in terms of win probability. The slope represents the increase in win probability for an increase in each marginal point.

The three models we have created are robust predictors of ML probability that do not suffer from multicollinearity. All results presented here are statistically significant, as indicated by the analysis on the F-statistic and t-score probabilities for each model. While the goodness of fit suffers slightly in the general case model, for the other models, we detect upwards of 95% of the variation in ML probability using the regression model's design.

Areas for further study

The next step in this analysis is to put our models to the test in the sports betting market. In order to understand their potential for profit, we need to look beyond their statistical significance and prediction accuracy because markets are both efficient and — to a certain extent — unpredictable. We define a model to "profitable" if it provides insight that turns a better profit than simply selecting the best price for a game.

Citations

Seabold, Skipper, and Josef Perktold. "Statsmodels: Econometric and statistical modeling with python." *Proceedings of the 9th Python in Science Conference.* 2010.