Sprint 2: Sports Betting With Data Science

By Justin Tunley

Problem Statement, Opportunity and Impact

Problem Statement

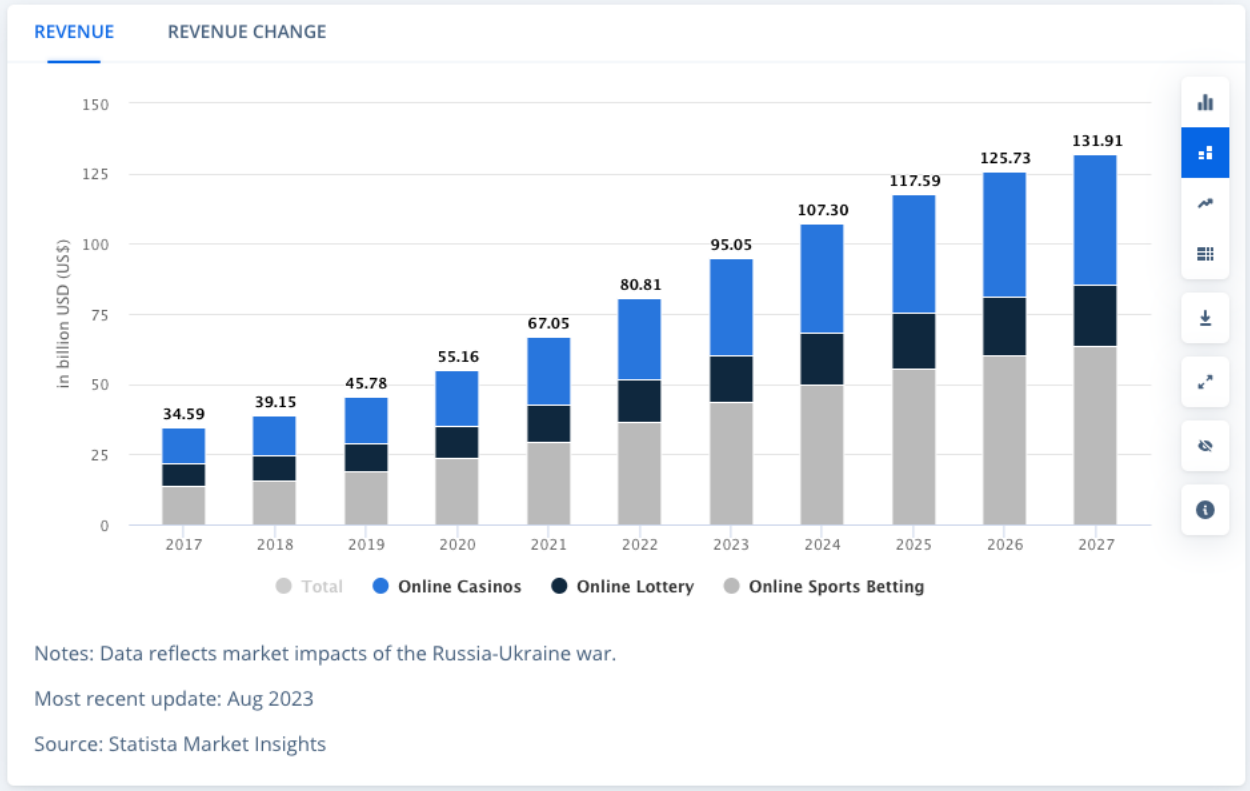
- Can we use team analytics to make predictions on future NFL games?

Opportunity

- Online Sportsbook: Create accurate lines that can help you make money
- Household Bettors: Beat the system by finding advantageous bets
  - Sportsbooks must adjust for user betting trends. You do not, giving you an opportunity to beat the system.

Impact

- MPM – Make People Money



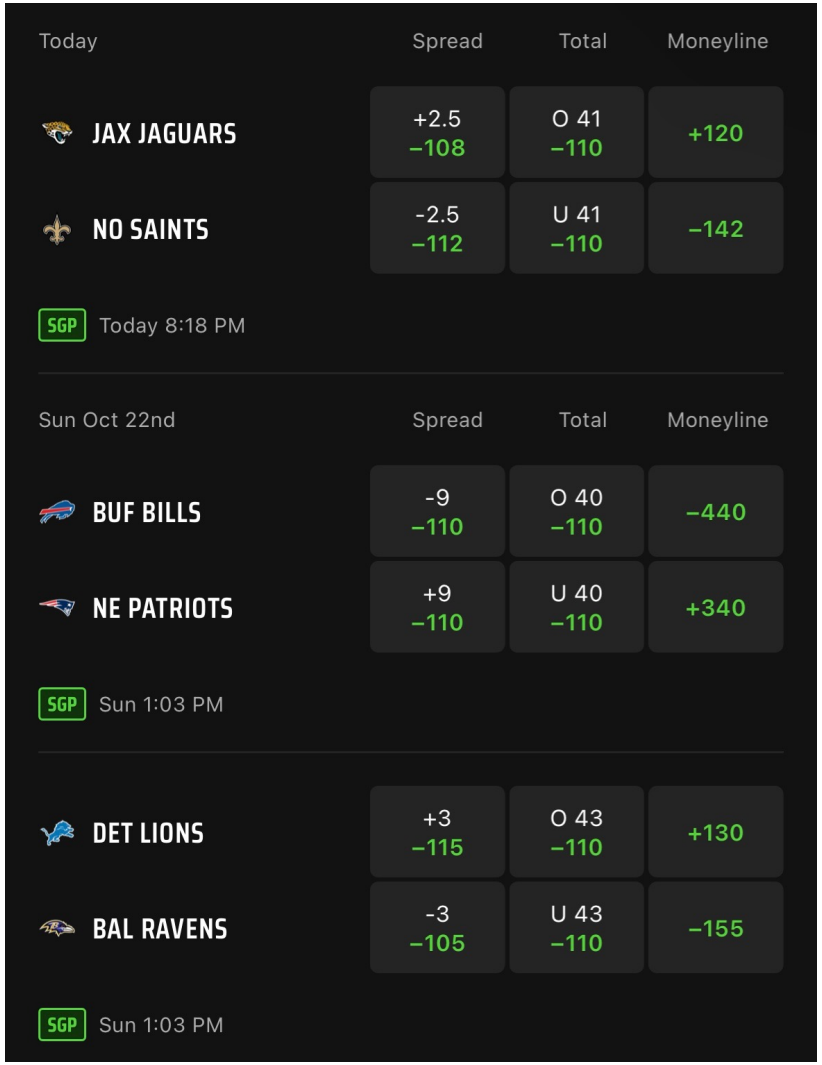
(Re)-Intro to Sports Betting

Key Terms

- **Moneyline**: Who will win?
  - Compare expected points between home and away teams.
- **Spread**: How much should the favored team win by?
  - The difference in expected points between teams.
- **Point Total** (O/U)
  - The sum of expected points for both teams.

Solving with Data Science

- 2 Regression models:
  1. Home Offense vs Away Defense --> Home Team Expected Points
  2. Away Offense vs Home Defense --> Away Team Expected Points



Datasets and Preprocessing (Set 1) – Off/Def

Scraping from ESPN (Season-Long Data)

- Create a for-loop cycling through ESPN hyperlink.
  - 32 teams x 19 seasons x 2 rows per season (playoffs not included)
- Off and Def are two separate rows. Split DF and rejoin along season index.

Feature Engineering

- Create a column for number of games.
- 85 Features ---> 70 Features (1 of 4 options)
  1. Drop columns that are redundant (i.e., OFF_Games and DEF_Games)
  2. Divide columns with 'yearly totals' by # of games (season points → PPG)
  3. Split columns into two new columns for more information (Kicks-Yds) → (Kicks), (K_Yds)
  4. Leave ratios as is.

❖ All columns were renamed, and defensive features begin with 'DEF_'

```
fun.columns
```

```
Index(['SeasonID', 'PPG', 'Tot_TDs_PG', '1st_Downs_PG', 'Rush_1st_Downs_PG',
       'Pass_1st_Downs_PG', 'OFF_1st_by_pen_PG', '3rd_Conv_Rate',
       '4th_Conv_Rate', 'Pass_Comp_Rate', 'Pass_Yds_PG',
       'Pass_Yds_Per_Attempt', 'Pass_Tds_PG', 'Off_Int_PG',
       'OFF_Sacks-Yards Lost', 'Rush_Att_PG', 'Yds_Per_Rush', 'Rush_Yds_PG',
       'Rush_Tds_PG', 'Off_Plays_PG', 'Tot_Yds_PG', 'OFF_Kickoffs: Total',
       'Avg_K_Return_Yds', 'OFF_Punt: Total', 'Avg_P_Return_Yds',
       'OFF_INT: Total', 'Avg_I_Return_Yds', 'Yds_Per_Punt',
       'OFF_Punt: Total Yards', 'OFF_FG: Good-Attempts', 'Touchback_Rate',
       'Total_Penalties-Yds', 'Avg_Pen_Yds_PG', 'Avg_TOP', 'OFF_Fumbles-Lost',
       'Games', 'Year', 'DEF_PPG_Against', 'DEF_Tot_Tds_PG_Against',
       'DEF_1st_Downs_PG_Against', 'DEF_Rush_1st_Downs_PG_Against',
       'DEF_Pass_1st_Downs_PG_Against', 'DEF_1st_by_pen_PG',
       'DEF_3rd_Conv_Rate', 'DEF_4th_Conv_Rate', 'DEF_Comp-Att',
       'DEF_Pass_Yds_Per_Attempt', 'DEF_Pass_Yds_PG', 'DEF_Pass_Tds_PG',
       'DEF_Int_PG', 'DEF_Sacks-Yards Lost', 'DEF_Rush_Att_PG',
       'DEF_Yds_Per_Rush', 'DEF_Rush_Yds_PG', 'DEF_Rush_Tds_PG',
       'DEF_Tot_Plays_PG', 'DEF_YPG_Against', 'DEF_Kickoffs: Total',
       'DEF_Avg_K_Return_Yds', 'DEF_Punt: Total', 'DEF_Avg_P_Return_Yds',
       'DEF_INT: Total', 'DEF_Avg_I_Return_Yds', 'DEF_Yds_Per_Punt_Against',
       'DEF_Punt: Total Yards', 'DEF_FG: Good-Attempts', 'DEF_Touchback_Rate',
       'DEF_Total_Penalties-Yds', 'DEF_Avg_Pen_Yds_PG', 'DEF_Avg_TOP',
       'DEF_Fumbles-Lost'],
      dtype='object')
```

Datasets and Preprocessing (Set 2) – Weather

Importing from Pro Football Reference (Ind. Game Data)

- Fairly clean on import
  - Dating back to 1960 so some teams don't exist anymore, but this won't matter for our objective.

Feature Engineering and Dummying

- Home Points + Away Points = Total Points
- 3 columns that must be dummied:
  - Temperature, Wind (mph), Humidity
- Create ranges for each and dummy to determine how each affects total points.

Final Goal

- Find coefficients for each data type and range, which can be added to my findings from DF1 to create even more accurate predictions.

```
max_wind = weather['wind_mph'].max()
min_wind = weather['wind_mph'].min()
max_temp = weather['temperature'].max()
min_temp = weather['temperature'].min()

print(f'Max wind recorded: {max_wind}')
print(f'Min wind recorded: {min_wind}')
print(f'Max temperature recorded: {max_temp}')
print(f'Min temperature recorded: {min_temp}')
```

```
Max wind recorded: 32.0
Min wind recorded: 1.0
Max temperature recorded: 96
Min temperature recorded: -7
```

```
# lets make cuttoffs for temperature first

# freezing:   -inf --> 14.99°
# cold:        15° --> 39.99°
# chilly:      40° --> 59.99°
# warm:        60° --> 79.99°
# hot:         80° --> inf

temp_range = weather['temperature']

def temp_dum(temp_range):
    for x in temp_range:
        if x < 14.99:
            return freezing
        elif x > 15 & x < 39.99:
            return cold
        elif x > 40 & x < 59.99:
            return chilly
        elif x > 60 & x < 79.99:
            return warm
        else:
            return hot
```

Exploratory Data Analysis

Interesting Takeaways

- Do rushing or passing metrics have a higher impact on expected points?
- Which types of turnovers are more detrimental to a team?
- Does special teams still matter in the NFL?

Want to Learn More:

These are questions I want to pursue but need more to answer…

- Does weather affect **certain teams** more than others? Is there a way to quantify this?

	PPG	Tot_TI
PPG	1.000000	0.9
Tot_TDs_PG	0.972306	1.0
1st_Downs_PG	0.783250	0.7
Rush_1st_Downs_PG	0.432319	0.4
Pass_1st_Downs_PG	0.612523	0.6
OFF_1st_by_pen_PG	0.212739	0.2
3rd_Conv_Rate	0.676861	0.6
4th_Conv_Rate	0.304851	0.3
Pass_Comp_Rate	0.568423	0.5
Pass_Yds_PG	0.649649	0.6
Pass_Yds_Per_Attempt	0.736760	0.7
Pass_Tds_PG	0.801459	0.8
Off_Int_PG	-0.409568	-0.3
Rush_Att_PG	0.246655	0.2
Yds_Per_Rush	0.270292	0.2
Rush_Yds_PG	0.314305	0.3
Rush_Tds_PG	0.593662	0.5
Off_Plays_PG	0.391475	0.3
Tot_Yds_PG	0.809432	0.7
Avg_K_Return_Yds	-0.001822	-0.0
Avg_P_Return_Yds	-0.032804	-0.0
Avg_I_Return_Yds	0.060474	0.0
Yds_Per_Punt	0.053558	0.0
Touchback_Rate	0.154620	0.1

Preliminary Modeling

Logistical Regression

- Error: ‘Endog must be in the unit interval’
- Attempts to run with less features still create problems…
  - Let's try a linear model.

Linear Regression

- Adj R-squared = 0.998
  - Really high…. Like way too high
- Potential problems:
  - Overfitting – too many features
  - Too much correlation between features
- Probable solution:
  - Neural Network

```
In [461]: print(B_linregFit.summary())
```

OLS Regression Results

Dep. Variable:	PPG	R-squared (uncentered):	0.998
Model:	OLS	Adj. R-squared (uncentered):	0.998
Method:	Least Squares	F-statistic:	2.449e+04
Date:	Fri, 10 Nov 2023	Prob (F-statistic):	0.00
Time:	07:49:32	Log-Likelihood:	-830.32
No. Observations:	608	AIC:	1689.
Df Residuals:	594	BIC:	1750.
Df Model:	14		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Tot_TDs_PG	6.1020	0.133	45.990	0.000	5.841	6.363
1st_Downs_PG	-0.0282	0.092	-0.308	0.758	-0.208	0.152
Rush_1st_Downs_PG	-0.1339	0.123	-1.089	0.277	-0.375	0.108
Pass_1st_Downs_PG	-0.0500	0.117	-0.426	0.670	-0.280	0.180
3rd_Conv_Rate	-0.0037	0.013	-0.283	0.778	-0.030	0.022
4th_Conv_Rate	0.0034	0.003	1.146	0.252	-0.002	0.009
Pass_Comp_Rate	0.7479	1.394	0.537	0.592	-1.990	3.486
Pass_Yds_PG	0.0168	0.003	4.927	0.000	0.010	0.023
Off_Int_PG	-0.5592	0.156	-3.592	0.000	-0.865	-0.253
Rush_Att_PG	0.1054	0.058	1.826	0.068	-0.008	0.219
Rush_Yds_PG	0.0117	0.013	0.884	0.377	-0.014	0.038
Yds_Per_Rush	-0.1418	0.305	-0.464	0.642	-0.742	0.458
Rush_Tds_PG	0.1317	0.205	0.644	0.520	-0.270	0.534
Off_Plays_PG	0.0245	0.024	1.026	0.305	-0.022	0.071

Omnibus:	7.087	Durbin-Watson:	1.779
Prob(Omnibus):	0.029	Jarque-Bera (JB):	7.010
Skew:	0.260	Prob(JB):	0.0300
Kurtosis:	3.080	Cond. No.	9.89e+03

Notes:
[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[3] The condition number is large, 9.89e+03. This might indicate that there are strong multicollinearity or other numerical problems.

Next Steps

- ***<u>Gameplan</u>***
  - Step 1: Neural Network on my clean data.
  - Step 2: More data (tendencies, redzone conversion, players)
  - Step 3: Incorporate weather data (Meteostat API)