Betting the NFL Over/Under

A Data Science Perspective

Eric H. Wilson

Business Problem

- ~38 million[1] people bet on NFL games, wagering > $12 billion[2]
- Profits are maximized by responding to public demand in setting Over/Under

Can we use data and machine learning to identify betting opportunities for profit?

[1] American Gaming Association, 2019
[2] Play USA, 2021

Stakeholders

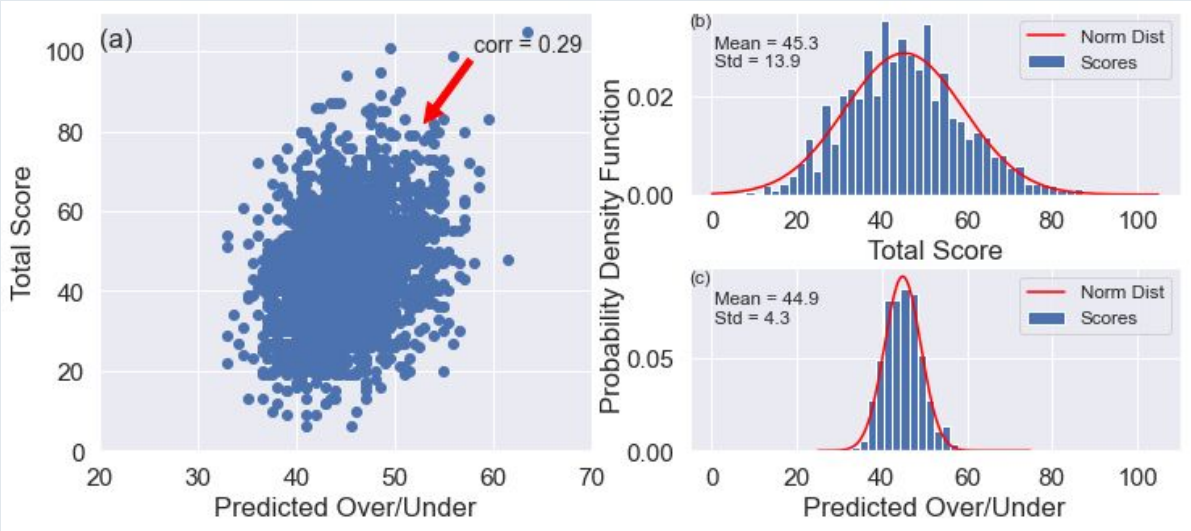
Coaches, General Managers, etc.

General Public

Where Do the Data Come From?

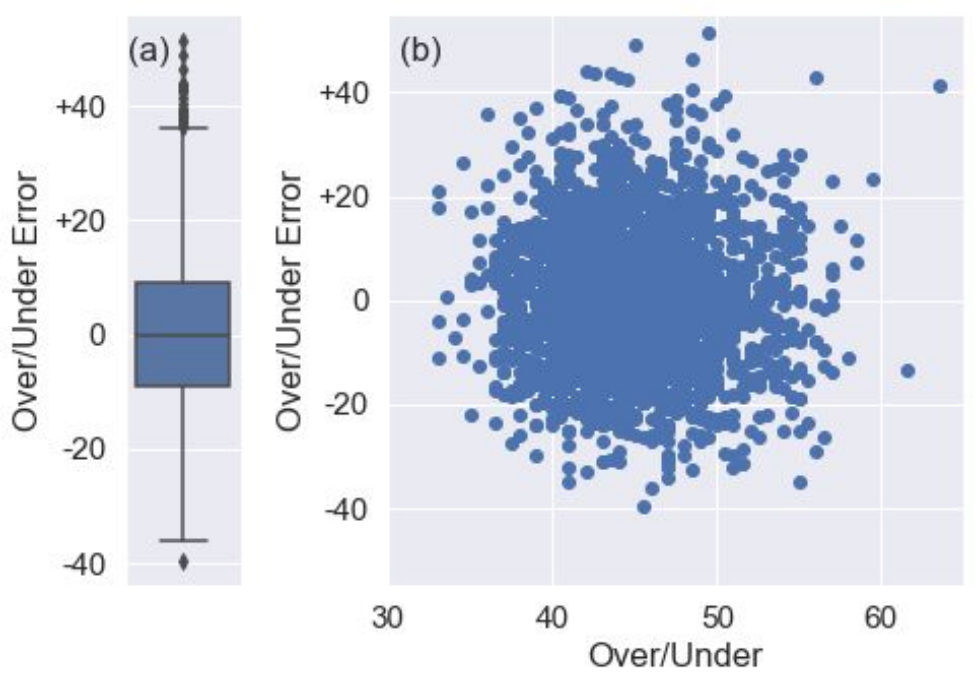
- Data from 2010-2019 comprising 2559 games compiled from http://pro-football-reference.com/
- 48 features generated from game conditions & averaging statistics over previous 5 contests,
  - Statistics include rushing and passing yards on both offense and defense, red-zone efficiency, and special teams stats
- Total score predictions for 1735 games are generated

What Do the Data Look Like?



- Total Score, Over/Under are normally distributed
- Total Score exhibits much larger range

What Do the Data Look Like?

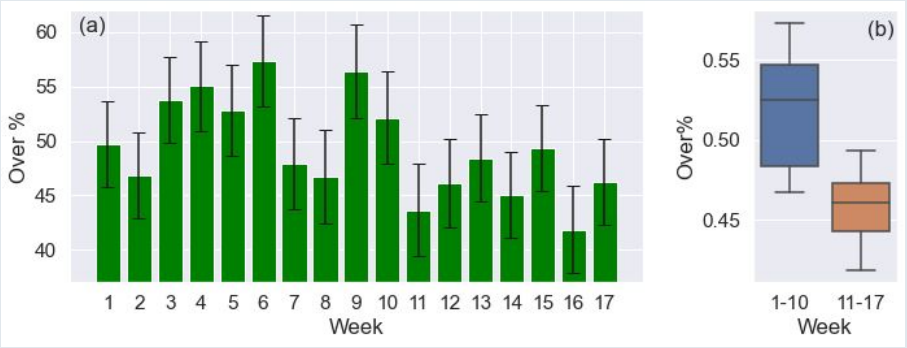


- Outliers are primarily distributed to the upside

Do Game Conditions Impact Total Score?

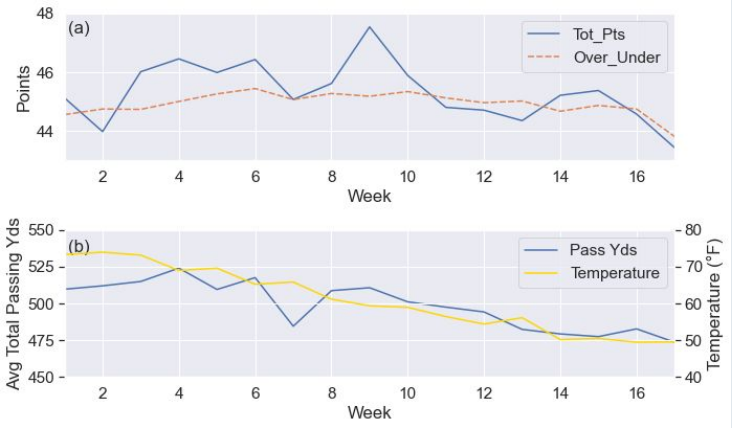
Week

- Over has statistically significantly higher winning percentage in first 10 weeks



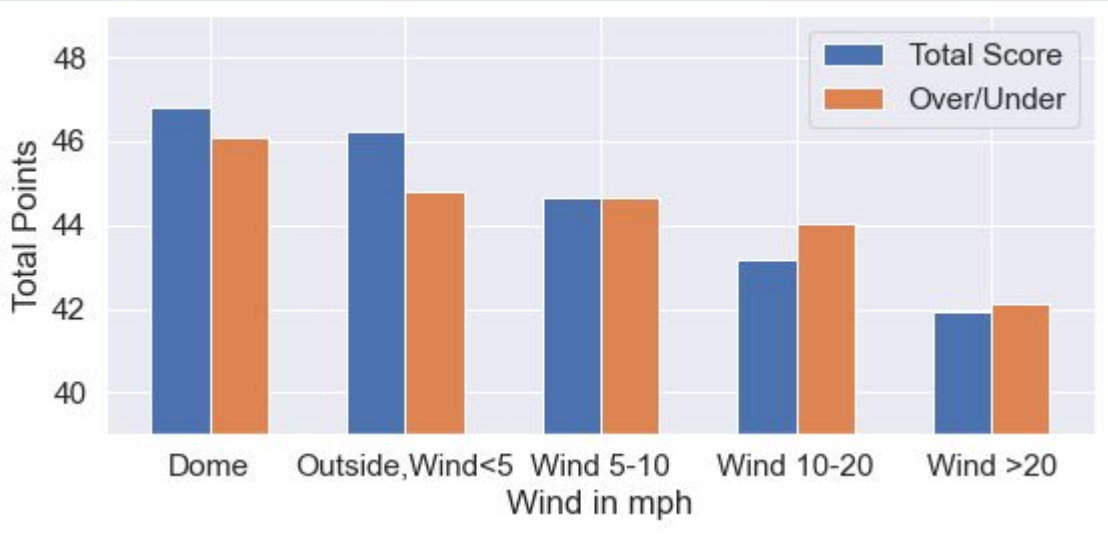
Temperature

- Temperature has significant effect on passing yards -> Total Points
- Not given sufficient consideration in Over/Under



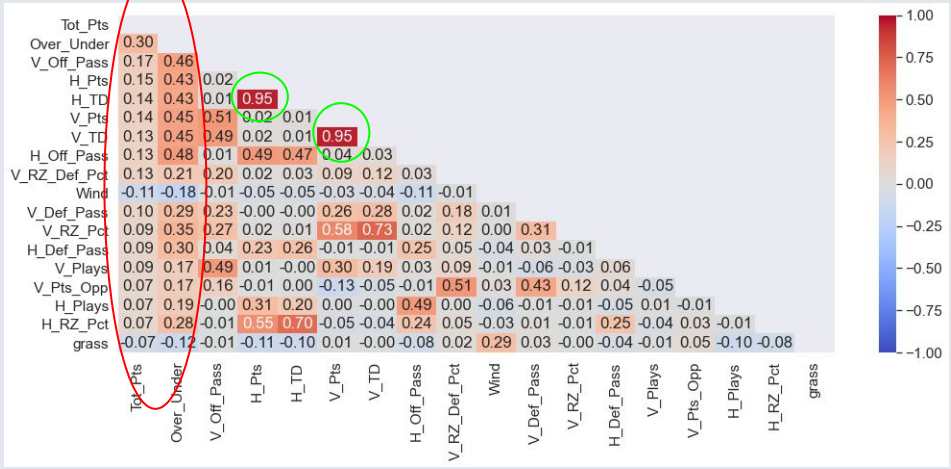
Do Game Conditions Impact Total Score?

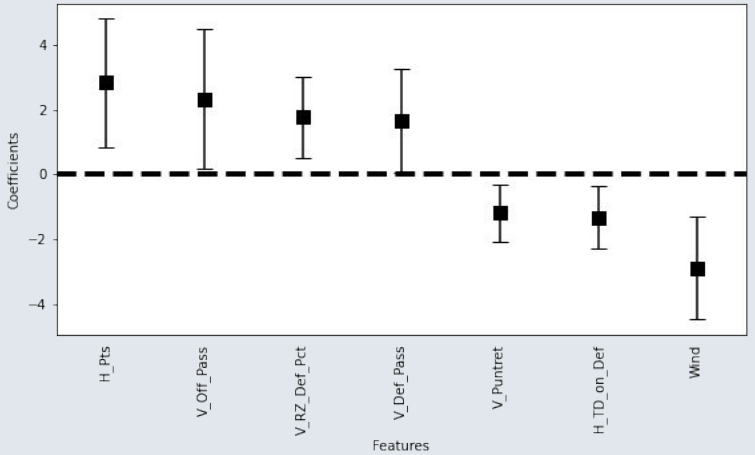
Wind has a significant effect on Total Points output



- Higher scoring in Dome games but not as much as assumed in Over/Under

Predicting Total Score – What Are the Important Features?



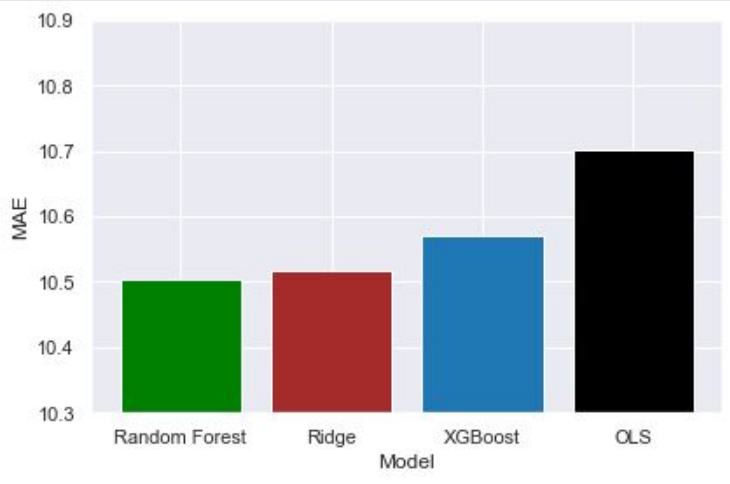


- No dominant feature
- V_RZ_Def_Pct has greater relative importance in Total Points than in Over/Under
- _TD high multicollinearity with _Pts

- Wind only significant feature with neg. correlation to Total Points
- Visitor features > home features
- Lack of offensive rushing features

Which Model to Choose?

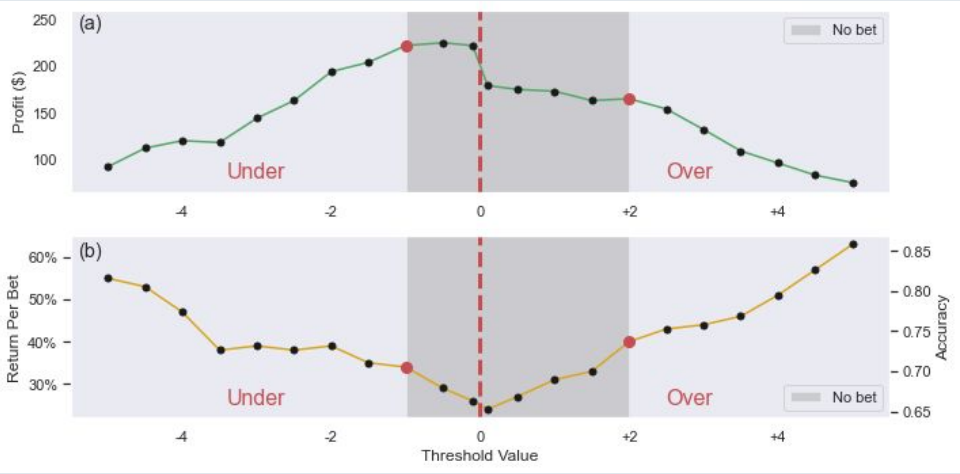
- Recursive feature elimination, hyperparameter tuning used to optimize models
- All optimized models contain large # of features
- Random Forest gives best results



Regression Model	Ordinary Least Squares	Ridge	Random Forest	XGBoost
# of Features	38	45	37	41
Tuned Hyperparameters		alpha = 0.44	n_estimators = 25, max_depth = 5, min_samples_leaf = 1	learning_rate = 0.11, n_estimators = 50, max_depth = 3, subsample = 1.0, colsample_bytree = 0.3, gamma = 0.0, min_child_weight = 1, reg_alpha = 0
Top five features	V_Off_Pass, H_Pts, V_Pass_Metric, Wind, H_Pass_Metric	Wind, V_Off_Pass, H_Pts, V_RZ_Def_Pct, grass	V_Off_Pass, H_Off_Pass, H_Pass_Metric, V_RZ_Def_Pct, V_Pts_Opp	H_TO_Lost, H_Off_Pass, V_Off_Pass, V_TD, H_TD
Test MAE	10.701	10.516	10.505	10.571

Threshold Tuning

- Threshold of deviation of Total Score from Over/Under is applied to optimize betting strategy
- A threshold of -1 (Under) and +2 (Over) is most optimal, yielding a seasonal return of 42% when applied to the test set



Threshold	Training Set	Test Set	Threshold	Training Set	Test Set
-5	$92 $0.55	-$24 -$0.19	All Over	$179 $0.24	$42 $0.05
-4.5	$112 $0.53	-$5 -$0.03	0.5	$175 $0.27	$31 $0.04
-4	$120 $0.47	$7 $0.04	1	$173 $0.31	$18 $0.03
-3.5	$118 $0.38	$32 $0.12	1.5	$163 $0.33	$53 $0.09
-3	$144 $0.39	$39 $0.12	2	$165 $0.40	$29 $0.06
-2.5	$163 $0.38	$29 $0.07	2.5	$154 $0.43	$13 $0.30
-2	$194 $0.39	$19 $0.04	3	$132 $0.44	$9 $0.03
-1.5	$204 $0.35	$8 $0.02	3.5	$109 $0.46	$33 $0.13
-1	$222 $0.34	$9 $0.01	4	$96 $0.51	$47 $0.25
-0.5	$225 $0.29	$21 $0.03	4.5	$83 $0.57	$63 $0.44
All Under	$222 $0.26	$45 $0.06	5	$75 $0.63	$47 $0.43

Conclusions

- Passing yards is most important single statistic for Total Score.
- Wind, temperature have significant effect on Total Score, not adequately taken into account in Over/Under determinations.
- No dominating feature in the prediction of total score, many features exhibit similar levels of importance – all optimized models employ a large number of features.
- Random forest model with 37 features produces the best results
- Threshold of Total Score with respect to Over/Under can be applied to optimize results