# Run-Pass Oracle (RPO)

I. Introduction

This submission aims to build a better methodology for predicting, pre-snap, whether a team will pass the ball. To do this, we created the **Run-Pass Oracle** model, built on top of a LightGBM architecture. We managed to achieve performance that surpasses state-of-the-art literature, while still maintaining interpretability.

Much of this performance is driven by feature engineering, including metrics to track pre-snap factors such as **Rectified Motion Value (RMV)**, **Defensive Congestion Index (DCI)**, and **Tempo Including Contextual Knowledge (TICK)**. These features heavily utilize the provided tracking data, and show a big boost over extant models, many of which barely eclipse 70% accuracy (those that do coming with major caveats). Perhaps most notable among thesse is Ben Baldwin's expected pass model; however, as the graphic below demonstrates, its predictions are still noisy:
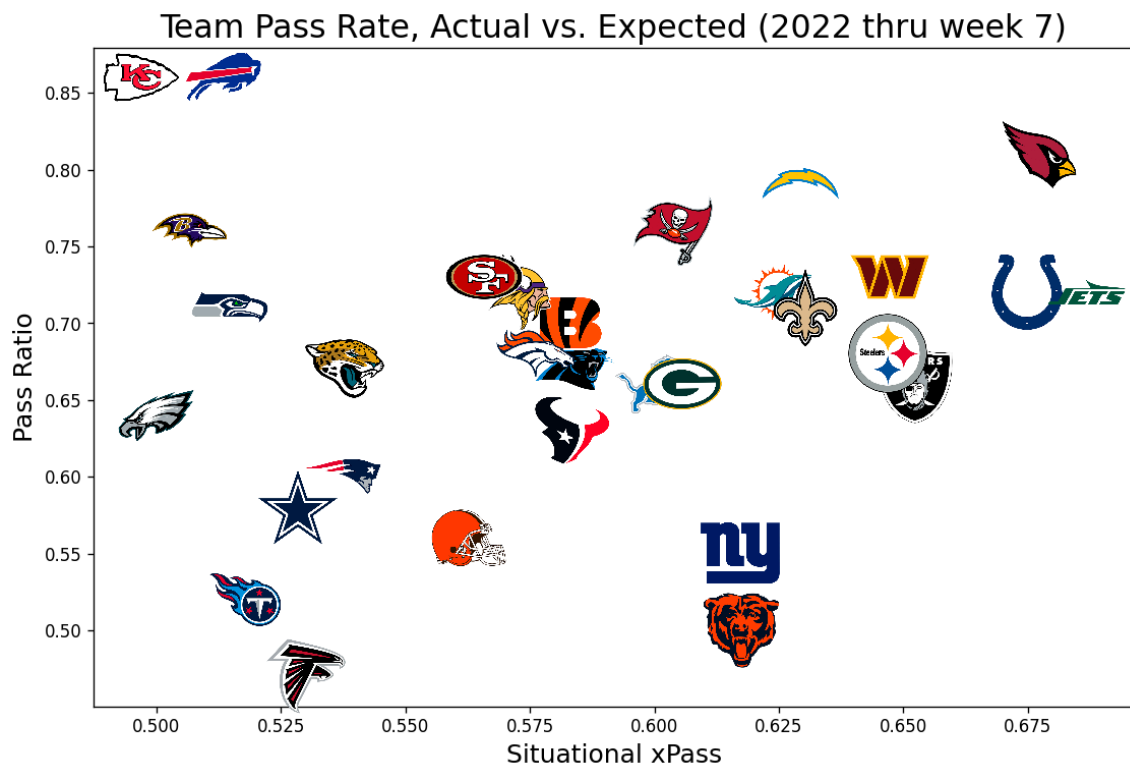
Baldwin's model includes important context like down and distance, and knows a little about team strength due to using Vegas odds. However, it doesn't incorporate the rich tracking data utilized in our model, and also misses significantly on teams like the Chiefs and Bills, whose superstar QB's make them far more pass-happy than expected. By incorporating tracking data, our model gets nearly a ten-percent boost over earlier literature on both our

test and validation sets, providing hope for generalizibility, while still demonstrating immediate usefulness.

II. Engineered Metrics & Other features

a) Feature overview

Our model encompasses seven features, many of which aggregate less-important features into more informative metrics:

- **Situational xPass**: An application of Ben Baldwin's pass rate model. While it is the dominant feature in our model, aggregating many important features, it still acheives only 70% accuracy on its own, and doesn't utilize tracking data.

- **Rectified Motion Value (RMV)**: Number of players in motion post-lineset, minus the number of players who shifted. Though seemingly simple, it is by far the most performative way of incorporating motion into our model we've found (see part d of this section for more)

- **Run Formation Flag**: Captures which formations negatively impact pass rate, such as I-formation (see part c of this section)

- **Tempo Including Contextual Knowledge (TICK)**: This metric combines our **Tempo** metric with other contextual info, such as player weight at key positions, to understand how play-calling tendencies combine with team personnel to predict pass rates (see section e)

- **Defensive Congestion Index (DCI)**, i.e., the mean pairwise distance between players on defense, with more recent frames weighted more heavily. The clear benefit of this is that we can pick up nuanced info that coarser features, such as offensive formations (e.g., 2x2) and men-in-box counts, may miss.

- **FAD (Final Acceleration Difference)**: Difference in maximum acceleration between the fastest- and second-fastest-accelerating players in the last few pre-snap frames. Specifically, these are the 15th-to-last through fifth-to-last frames, in order to provide a realistic buffer before the ball is actually snapped.

TODO: find graphic for here; maybe dots demonstrating 'defensive congestion index?

b) Feature impact analysis

Below are our shapley values for our dataset. Given the breadth of features that our **Situational Expected Pass** metric encompasses, it makes sense why it would be so

important. It does not, however, render the other features entirely useless, as they each, on average, still makes significant contributions to our prediction values.
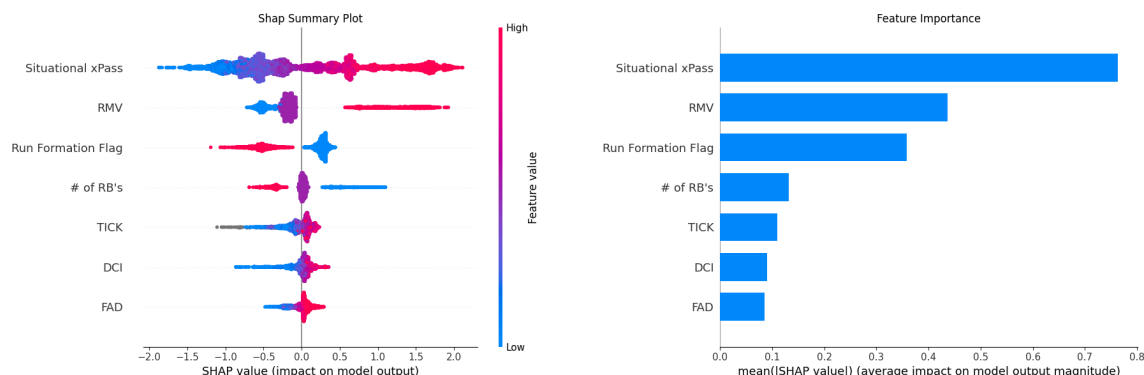
Given most of our features are positively correlated with passing, we see an expected result where higher feature values (red) imply higher a pass rate (more red points to the right). The inverse—i.e., low (blue) values depressing pass rate—is also generally true, excepting our **negative formations** feature, which, when flagged as True, should lower expected pass rate (see next section). One key thing to note is how while the mean Shapley values paint a meaningful disparity between our four lower features, they each exhibit similar distributions in our violin plot, and are thus closer to equal importance than the prior graph would imply.

c) Examining Formational Impact

Our model heavily relies on the concept of "negative formations", i.e., formations that heavily favor runs. This feature is an important example of how necessary it is to encode concepts that, however obvious they may seem to humans, are still necessary to teach our models about. One of these formations is **I-Formation**, a heavier personnel grouping used almost exclusively for runs or play-action that is perhaps *the* stereotypical run formation. Similarly self-explanatory is **Pistol**, used to great effect by teams like the Ravens to take advantage of QB Lamar Jackson's running prowess. Less explicable is the **Single-back** formation's inclusion (TODO: Explain it)

```
In [1]:   import plotly
          plotly.io.read_json('image/cincy_drive.json')
```

The above graph shows just how big a tell formations are in whether a team will pass or not. Above an example from a week 7 **Cincinnati Bengals** drive, who were one of the most pass-happy teams over the course of the 2022 season, with an overall pass rate of 62%. Our model thinks that, given all the data we provide it, the Bengals were more likely to pass than run on almost all the plays of the drive.
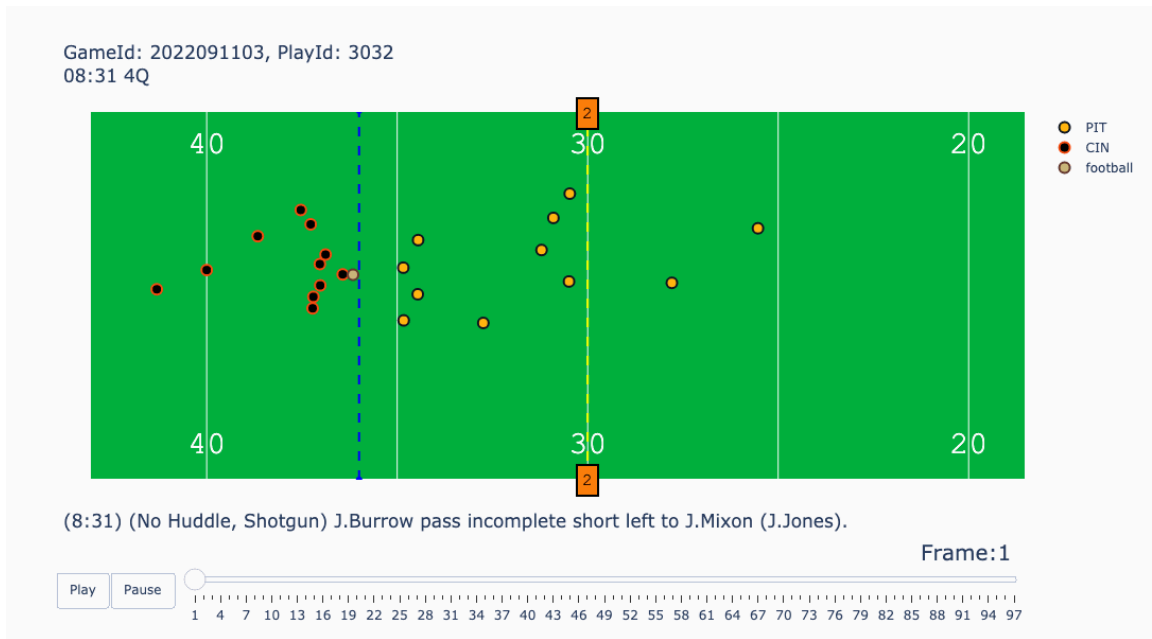
This of course tracks with the common understanding that Bengals QB Joe Burrow strongly prefers to pass from Shotgun formation. The Bengals lining up in one of our flagged "run formations" would thus be an even bigger tell than usual that they're going to run th ball.

d) Motion

Below is a typical example of pre-snap motion from a Week 6 matchup between Miami and Minnesota. In it, we see Miami (left, offense) running back Raheem Mostert move across the formation right before the snap. Such an example is where our **FAD (Final Acceleration Difference)** metric proves most useful, since it tells us how much faster the likely motion player (in terms of acceleration) is going than the second-fastest player. Since only one player can be in-motion when the ball snaps—though many teams push this rule to its limits —this intuitively helps us suss out whether a player went in motion on a play.
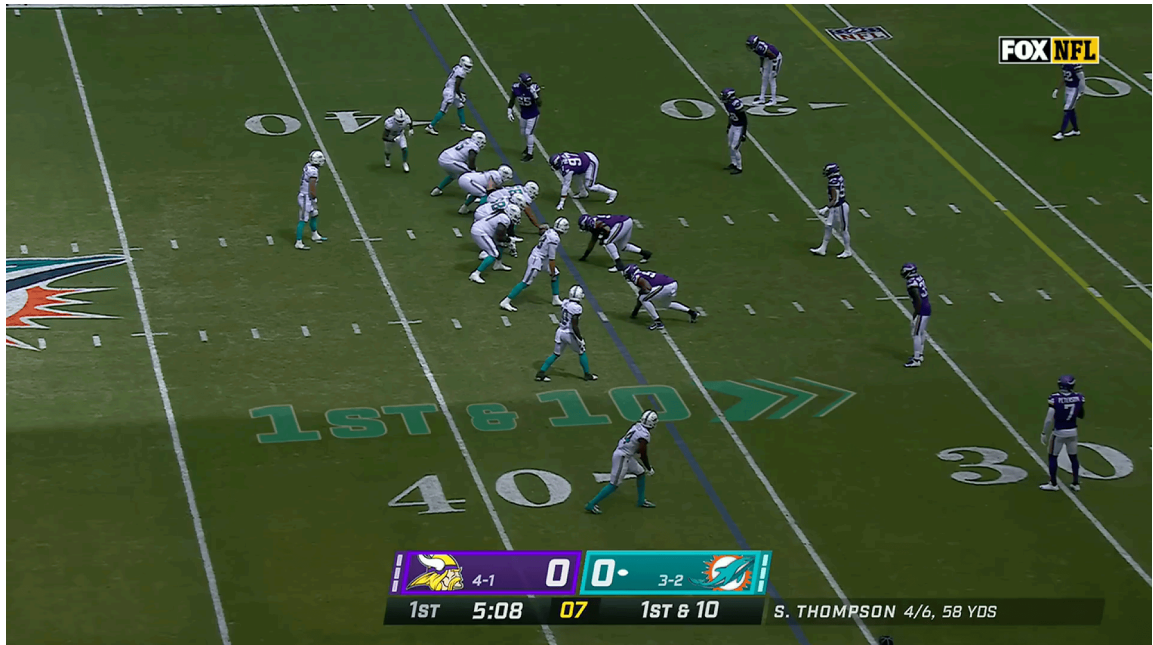
The whole purpose of **FAD**, however, is to clean up inconsistencies in **RMV (Rectified Motion Value)**, one of our most useful features. RMV is seemingly simple, subtracting the number of shifted players from the number in motion. Yet it it's also surprisingly effective, in large part because it encompasses the main things we want to know about pre-snap movement. The biggest benefit is that it essentially removes some noise from our data, since many of the players shifting pre-snap are erroneously market as "in motion", due to the speed at which they shift. Thus, while it seems coarse, **motion-momentum** helps us differentiate shifts from actual motion plays.

GameId: 2022091103, PlayId: 3032
08:31 4Q

(8:31) (No Huddle, Shotgun) J.Burrow pass incomplete short left to J.Mixon (J.Jones).

Frame:1

Above is such an example, where a quick exchange on the left side results in two shifting players erroneously being marked as "in motion". We see at the end of our graphic, however, that there's a real motion occurring, which we want our metric to reflect. Thus, by subtracting what our data says are 2 shifted players from our 3 in motion—as RMV does— we get the actual correct number of players in motion, which is 1.

e) Tempo Including Contextual Knowledge (TICK)

A key concept we sought to include in our model is the idea of **tempo**, i.e., the pace at which a team marches down the field. The core tempo metric itself is fairly simple: within an ongoing drive, we consider the mean time a team's snapped the ball at, the pass rate for the current drive, and the mean EPA this drive. Perhaps counterintuitively, mean EPA negatively relates to pass rate on a drive, so we subtract it. This is perhaps because if a team has already had a couple explosive (i.e., high-EPA) plays, and the drive is still ongoing, they may now be in the red zone, and thus now prefer to run the ball.

$$tempo = .1 * mean\_clocksnap * drive\_pass\_rate - mean\_epa$$

While this tempo information is mildly useful on its own, we also want to incorporate info about the *personnel* teams use to better gauge team pass tendencies. Enter the **xpass_bmi** metric, which seeks to incorporate both contextual and historical data:

$$xpass\_bmi = off\_xpass * box\_ewm\_dl\_bmi - .4 * qb\_pass\_rate\_ewm$$

off xpass—not to be confused with from **xpass_situational**—bakes in weight info and the like implicitly, as the presence of run-blocking tight ends, for example, will depress expected pass rate.

$$off\_xpass = \frac{1}{N} \sum_{i=1}^{N} player\_xpass_i$$

where an individual player's xpass, calculated through the week prior, is defined below, where m is the current week, n is the number of plays in a given game, and x is whether the player's team passed the ball (note: n is usually used for a constant, but is kind of variable here; how to best represent?)

$$player\_xpass_m = \frac{1}{m-1}\frac{1}{n}\sum_{i=1}^{m-1}\sum_{j=1}^{n}x_{ij}$$

(define dl bmi etc)

$$TICK = tempo * xpass\_bmi$$

TODO: off_xpass, pass_rate_ewm (do a sum over 1 thru N where N is the latest week preceding this one?)

$$box_ewm_dl_bmi = box_ewm*mean_dl_bmi

III. Model

a) Model design

Our model itself was a LightGBM model, with a maximum depth of 5 as to prevent overfitting. For the 9 weeks of tracking data provided, we trained from weeks 3-7, using week 8 as a validation set, then week 9 as test proper.

The first two weeks are excluded for their heavy reliance on prior-year information, which is much noisier than current-season info. This is an issue acknowledged by members of the football analytic community, including FTN's Aaron Schatz. He provides a weighted version of his DVOA metric that weighs recent weeks more highly as the season goes on, becoming more predictive as a result.

Thus, given that our model is set up to predict on mid-season plays, fitting on more recent data is empirically sound. The benefit of this is seen in how the model gets a 3% performance boost in test over validation after re-fitting, showing the likely benefit of additional data to our model.

(TODO: Make slightly less info-dense/horizontal version of this)
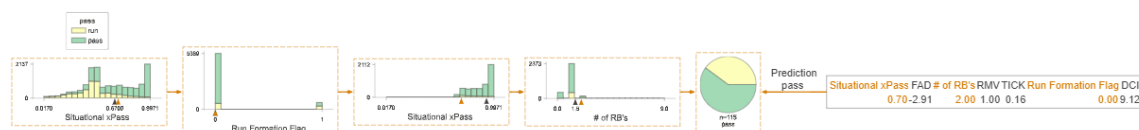
The above graphic demonstrates what a typical tree for our LightGBM model looks like. Though formational and situational (down, distance, etc.) info are quite important, it's crucial to note that all our created metrics are utilized. That we can incorporate all these features to

a meaningful extent while limiting our maximum depth to 4 (to prevent overfitting) bodes well for our model design.

b) Model performance

As previously discussed, our model beats most play-level benchmarks in the public literature for play-level pass prediction. The model we present is, in our opinion, the best blend of interpretability and generalization we've achieved.

Our model shows a slight bias toward predicting teams will pass; however, this is to be expected, since teams generally pass more frequently than they run. Notably, its misses are essentially equal between events that were runs and passes, which bodes well for its ability to generalize.
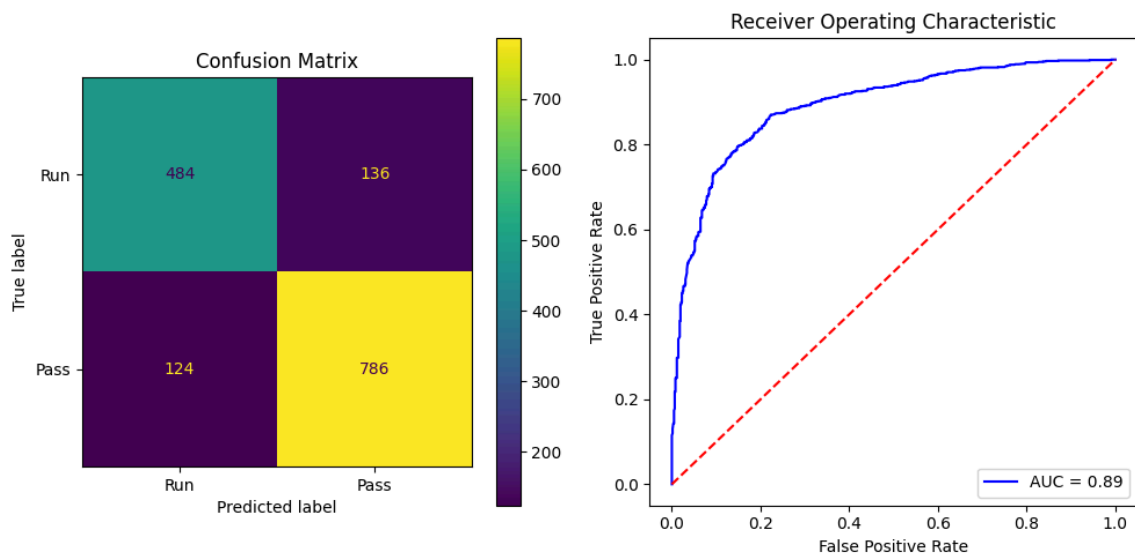
IV. Conclusion

V. Citations

Joash Fernandes, Craig et al. 'Predicting Plays in the National Football League'. 1 Jan. 2020 : 35 – 43.

Goyal, Udgam. (2020). Leveraging machine learning to predict playcalling tendencies in the NFL.

Marius Ötting, Predicting play calls in the National Football League using hidden Markov models, IMA Journal of Management Mathematics, Volume 32, Issue 4, October 2021, Pages 535–545, https://doi.org/10.1093/imaman/dpab005