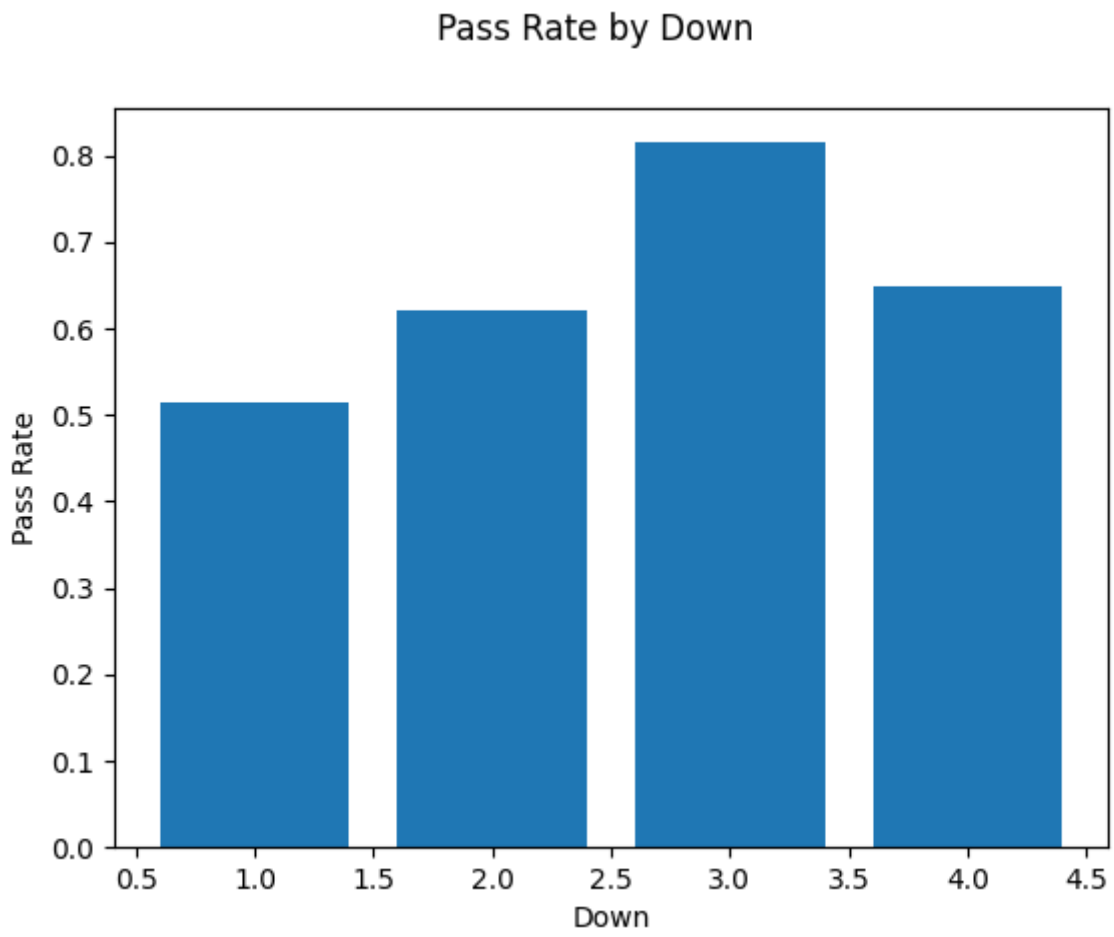
Goal

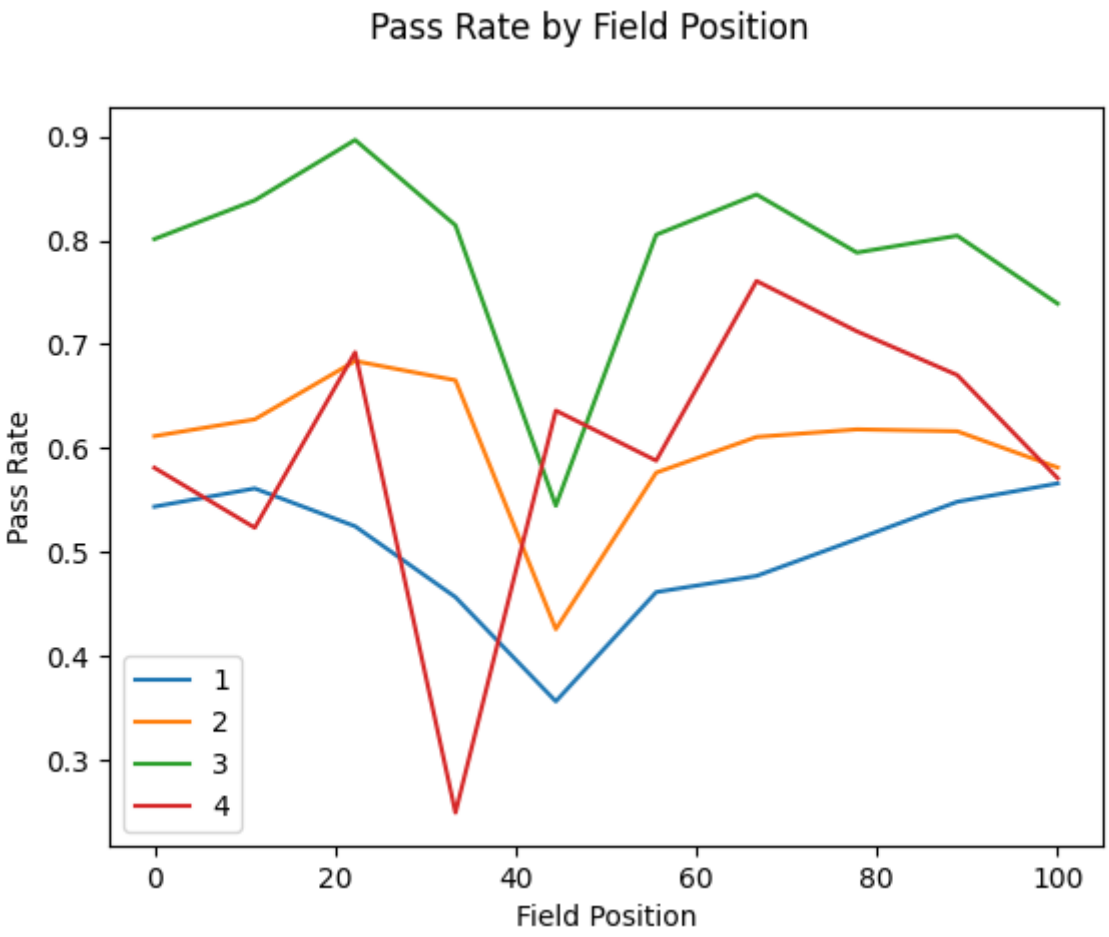
Build a predictive model that determines whether the next NFL play will be a run or a pass.

Exploratory Data Analysis

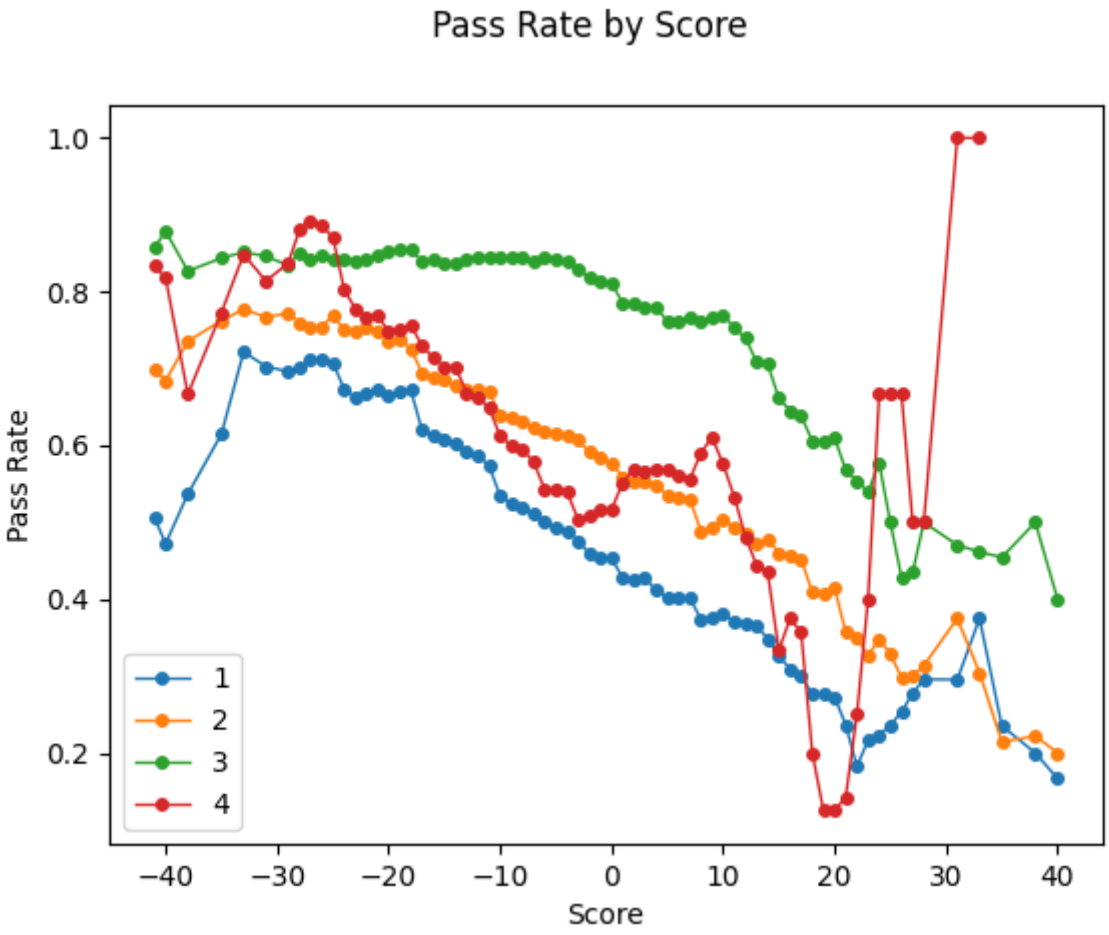
There are over 19,500 run or passes across two seasons in the dataset, each with over 90 features describing the plays. Teams passed on 61.2% of the plays, but this varies when conditioned on other factors. Teams passed the most on third down and least often on first down.



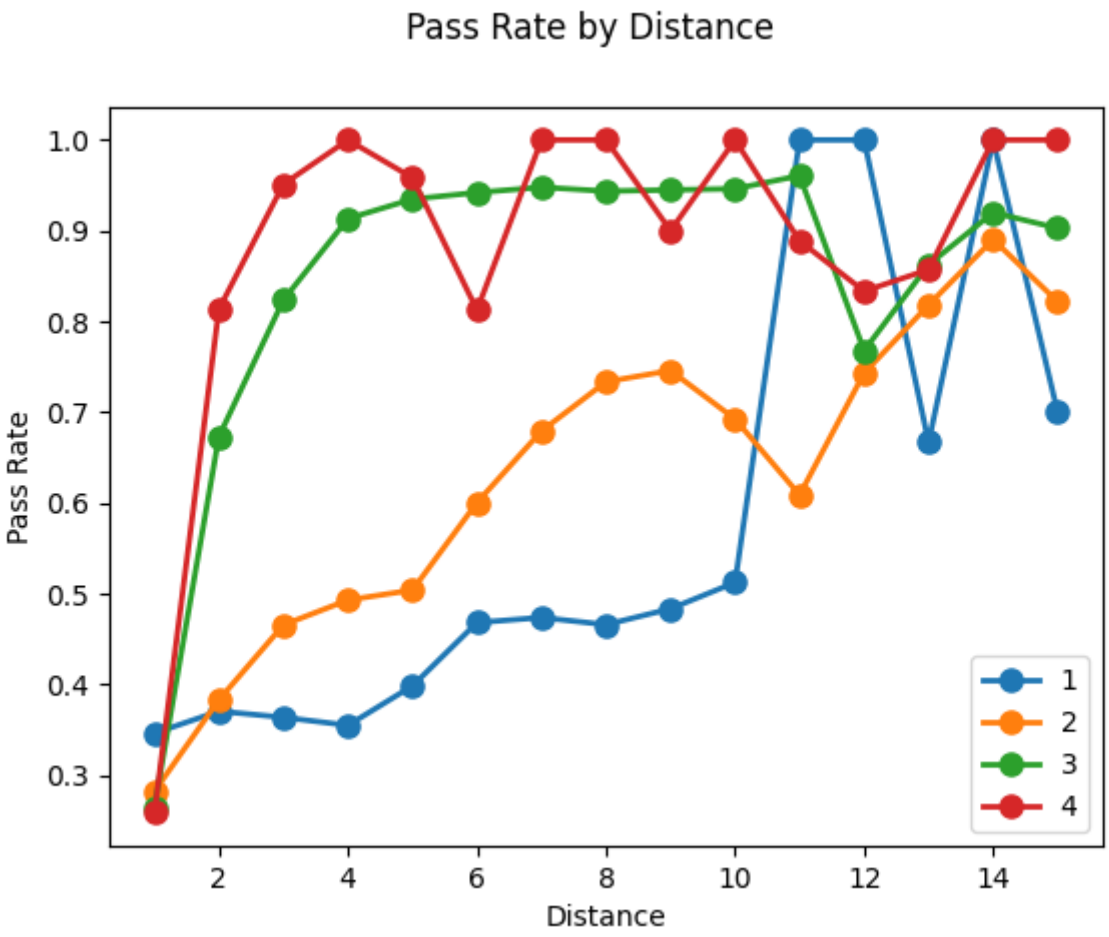
Teams tend to pass less around the middle of the field (~50 field position). This could be in order to ensure a couple extra yards to get into/improve field goal position.



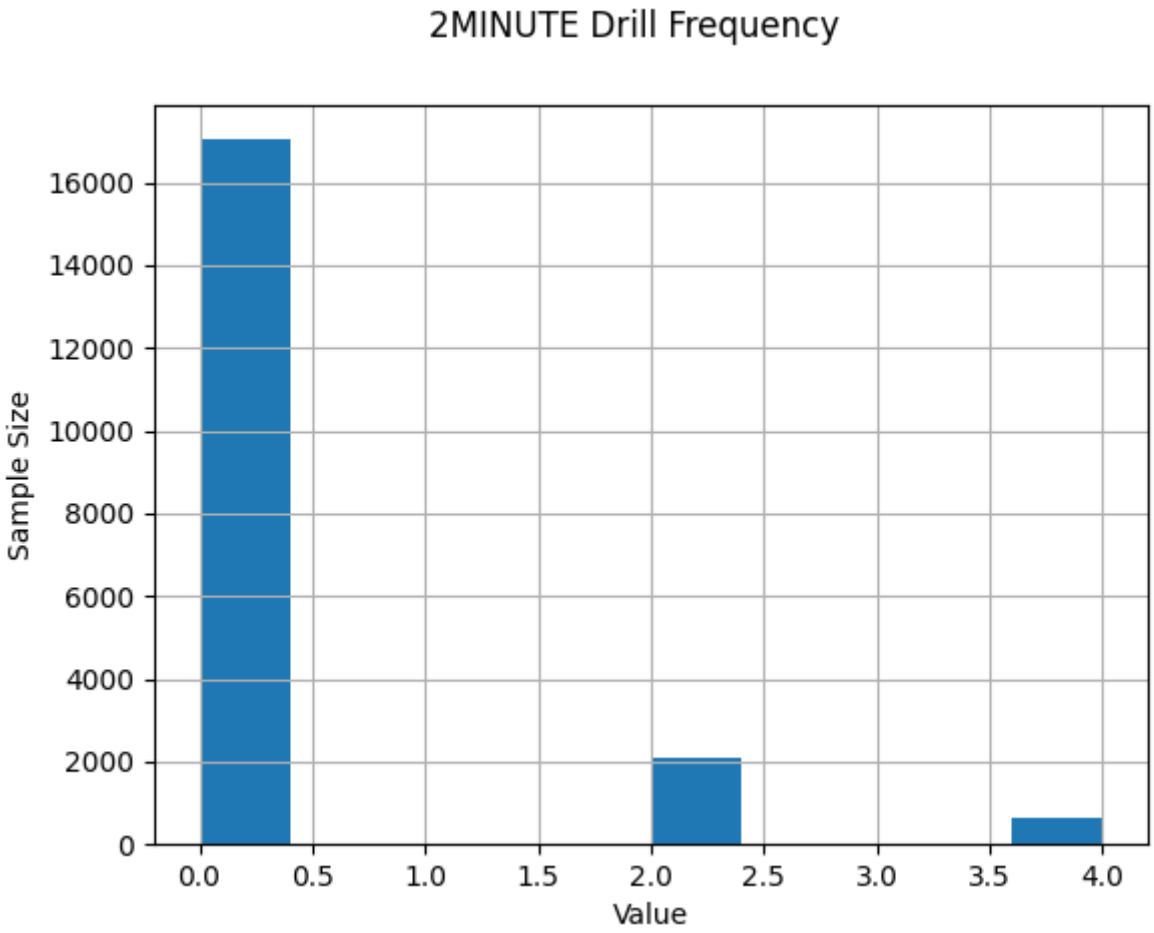
We see more consistent trends when looking at passing rates by score differential - teams with the lead are running the ball more to eat clock, while teams trying to catch up are throwing more often. We also see a similar trend in relation to downs as teams running the ball less in later downs when conditioned on score (for the most part - see fourth downs when the offensive team is up by at least two scores).



Unsurprisingly, teams tend to pass more the further away from the first down line they are, and tend to run the ball in earlier downs when within 10 yards.



About 10% of the plays in the dataset were 'Two Minute Drill' situations. Teams passed about 74% of the time in these situations.



Prior Work

Richard Anderson with Open Source Football estimates the probability of a QB Dropback with an XGBoost model using the features:

- Down (limited to 1,2,3)
- Yards for first down
- Yard line
- Score Differential
- Quarter
- Time remaining in half
- Number of timeouts for the offense and defense

With 100K training examples from 2016-2019, he achieves 69.1% accuracy.

Feature Engineering

Heuristically we can think of our features as three different sections: game information, team tendencies, and team talent.

Game Info

Game Info comprises of the descriptive state of the game at the time of the play, such as week, quarter, down, distance, etc. The "base model" only includes these features:

- WEEK
- QUARTER
- SCOREDIFFERENTIAL
- SCORE
- DISTANCE
- DOWN
- FIELDPOSITION
- DRIVE
- DRIVEPLAY
- OFFTIMEOUTSREMAINING
- DEFTIMEOUTSREMAINING
- HASH
- SPOTLEFT
- 2MINUTE
- CLOCK

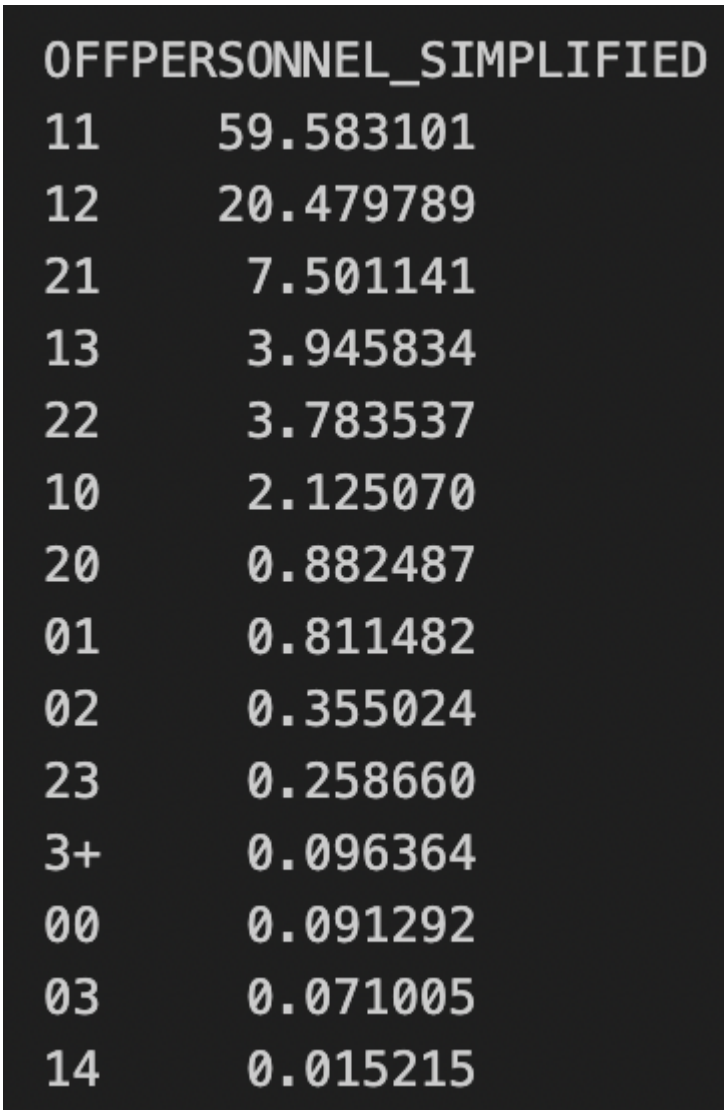
Team Tendencies

Team Tendencies encompass additional PFF features describing an aspect of the play in greater detail, e.g. Hurry, Play Action, Pass Depth, Time to Pressure, Middle of the Field Open or Closed (MOFOC). Before the start of each play, we can calculate a summary of these metrics over the team's previous plays to use as a feature to predict what they will currently do. If the middle of the field was open last play, is a team more likely to run the ball? What if it has been open 50% of the time throughout the game? Or if the defense has historically always left it open?

For each of these stats, I record the previous result, the average over the previous plays in the current game, and the average over all previous plays by the team. The averages don't start being recorded until 10 and 100 plays respectively (how far back to go or how long to wait before registering can be tuned).

Personnel Groupings

I simplified the representations of the offensive and defensive personnel groupings, due to sample size and cardinality concerns. There were 40 unique offensive personnel types yet only six were present in more than 1% of the samples. By ignoring receiver-eligibility, additional quarterbacks (+Q), and grouping all three running back sets together, this was trimmed down to 14 distinct categories.



OFFPERSONNEL_SIMPLIFIED	
11	59.583101
12	20.479789
21	7.501141
13	3.945834
22	3.783537
10	2.125070
20	0.882487
01	0.811482
02	0.355024
23	0.258660
3+	0.096364
00	0.091292
03	0.071005
14	0.015215

For the 39 distinct defensive sets, they were replaced with three new features recording the number of linemen, linebackers, and defensivebacks on the field during the play.

Apriori I'm uncertain which of these two approaches would be best - leaving the feature as categorical (like offensive personnel) or creating new continuous variables (e.g. number of linemen). Representing the formation as continuous counts of each position could allow the model to learn to focus on whichever parts of the formation it deems most valuable. For example, if an optimal split was 'are there less than 3 lineman', the model could not do this split easily in the categorical representation. However, if an optimal split was a specific formation such as '4-2-5', the model would need at least four splits under the continuous encoding to representation ('linemen' < 4 (yes) -> 'linemen' < 2 (no) -> 'linebackers' < 3 (yes) -> 'linebackers' < 2 (no)), compared to the one under the categorical encoding. The decision should be based on how informative one thinks each component of a defensive formation is independent of the others (e.g. how useful a split such as 'number_of_linemen < threshold could be). One could validate this empirically by trying each and seeing which model performs better.

Team Talent

Team Talent comprises of how good each team is at running and passing the ball.

- historical_yards_per_carry
- historical_yards_per_pass_attempt

- historical_yards_allowed_per_carry
- historical_yards_allowed_per_pass_attempt

Models

I sought a middle ground between simpler baseline models (e.g. linear regression) and powerful yet complex models such as neural networks. Neural networks are generally the most powerful and expressive models available, as evidenced by recent advancements in deep learning and natural language processing. Ideally we would treat our input as a sequence of plays and use a sequential neural architecture, yet they are less interpretable than other methods. Additionally, they can be more complex to train and would have required more data preprocessing to convert the data into sequences.

Therefore the first model to start with was XGBoost, as it has been demonstrated to be one of the leading classification algorithms across many tasks. XGBoost also natively handles missing inputs nicely and is more interpretable with easily computable quantitative feature importance metrics. Additionally, a decision-tree-based framework might also closely align with how a coach or player mentally works through the decision to run or pass.

I ran several models with different subsets of features. Our baseline is considered to be the best naive guess; i.e. what accuracy one would get by predicting the most common class every time (in this case, a pass). The "Base Model" only uses the Game Info features previously mentioned. The "Continuous Features Model" doesn't use the categorical features within the team tendencies feature grouping, such as Offensvie Personnel and Center Pass Block Direction. The "Teams Included Model" adds as features the label for the offensive and defensive team. The "All Features Model" incorporates all engineered features with basic hyperparameter tuning, while the fully tuned model has more comprehensive hyperparameter tuning.

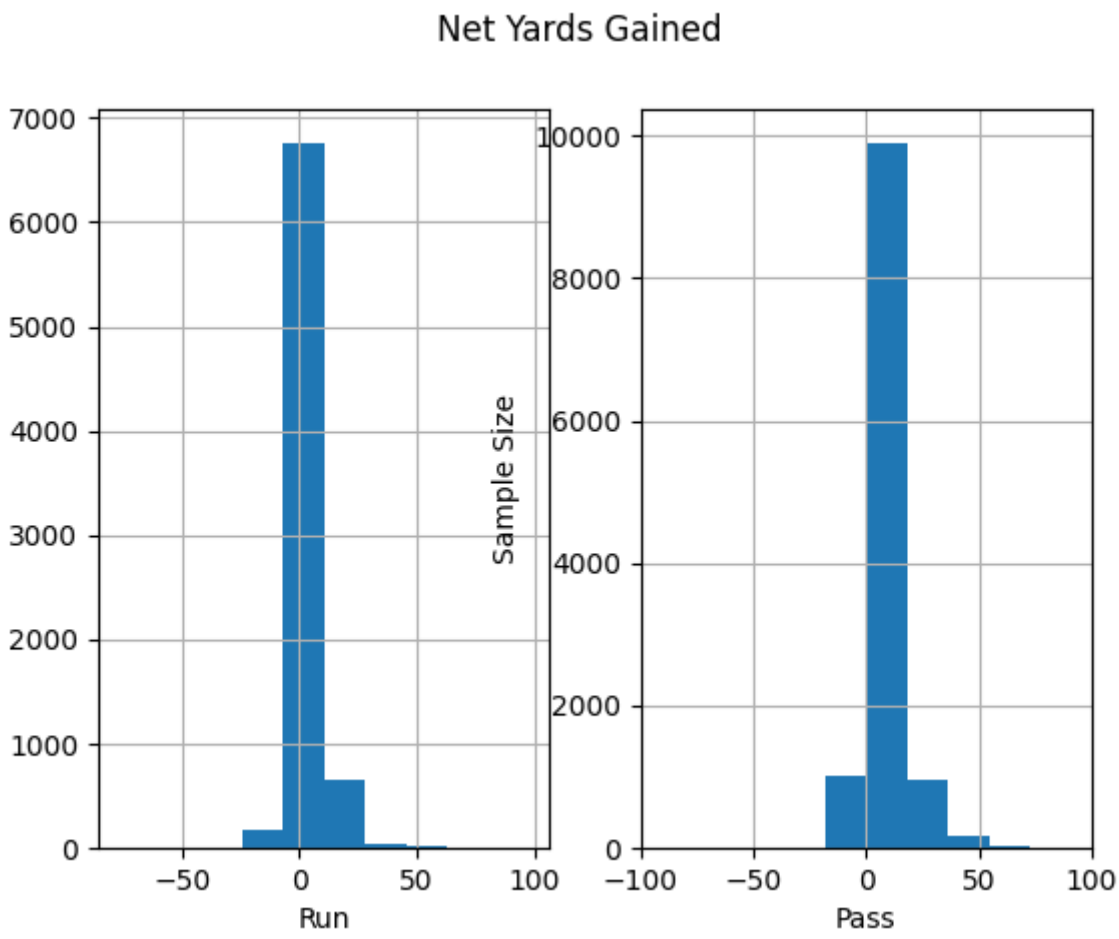
Results

The naive baseline of always predicting a pass is generally correct around 60-62% of the time. The final fully tuned model using all the features performs best with the highest validation AUC, validation accuracy, and test accuracy. While it is a 10% difference in accuracy from the naive baseline, the extra engineered features add only marginal improvements.

Model	Val LogLoss	Val AUC	Validation Accuracy	Test Accuracy
Naive Baseline:	___	___	62.64%	60.95%
Base Model:	0.557	0.769	69.73%	69.52%
Continuous Features Model:	0.564	0.762	69.73%	69.63%
Teams Included Model:	0.556	0.763	69.88%	69.04%
All Features Model:	0.546	0.772	70.53%	70.07%
Fully Tuned All Features Model:	0.539	0.779	71.33%	71.81%

Final Model

Presented are more in-depth classification metrics for the final model. The model is tuned on validation accuracy, yet the needs of a team may support tuning with respect to a different measure. For example, it is likely that a false negative would be considered a worse outcome than a false positive; i.e. a wrong run prediction can hurt a team more than a wrong pass prediction, since the yards gained distribution has fatter tails for passes than runs (the visualization is small in the plot, but one can see a bit more blue over the 50 xtick on the right).



Intuitively, this is roughly saying that a team fully committing to a run can be burned by a deep throw more than a team fully committing to a pass can be burned by a huge run. So in high-leverage situations one might want to focus on run precision (minimizing the number of false run predictions).

	precision	recall	f1-score	support
R	0.64	0.53	0.58	749
P	0.74	0.82	0.78	1256
accuracy			0.71	2005
macro avg	0.69	0.68	0.68	2005
weighted avg	0.71	0.71	0.71	2005

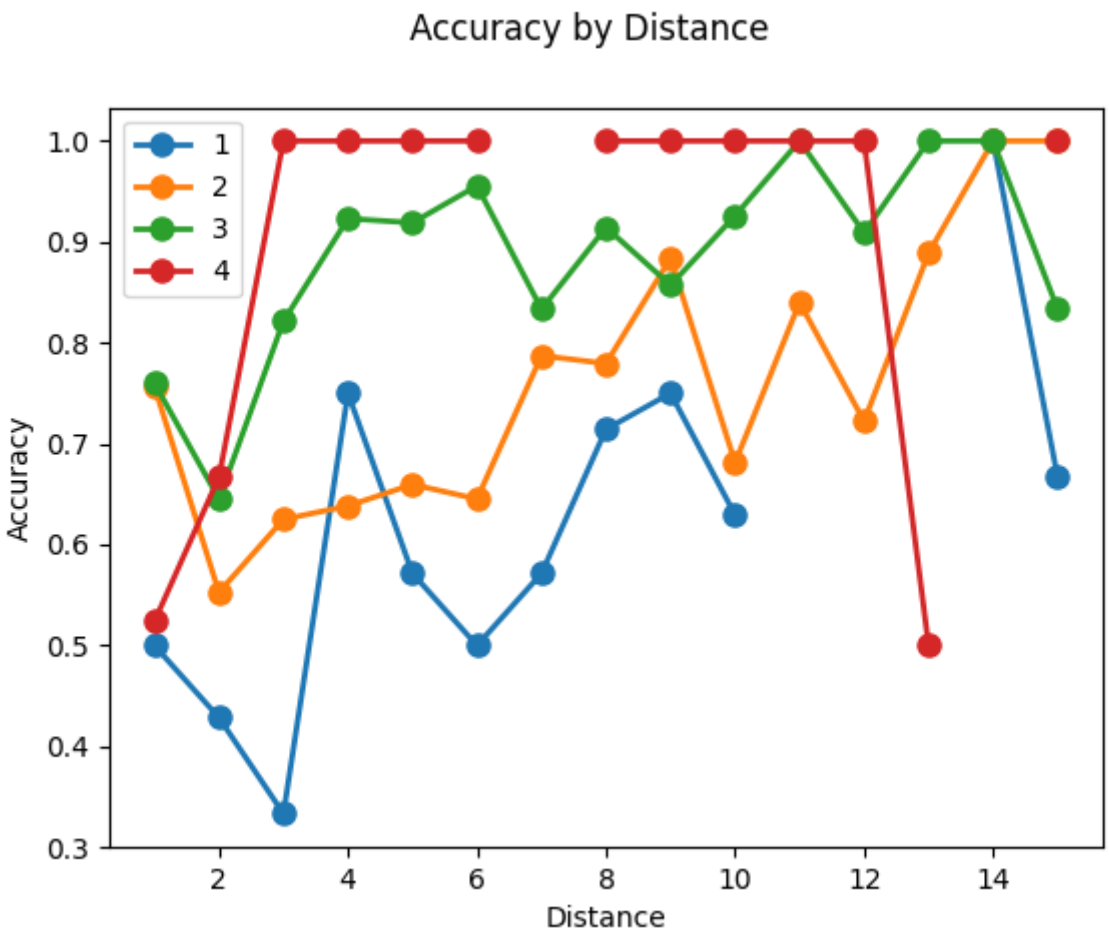
We can see that the model does a better job overall with passes than runs, with better precision and much better recall.

Confusion Matrix:	**Pred Run**	**Pred Pass**

Confusion Matrix:	Pred Run	Pred Pass
True Run	394	355
True Pass	220	1036

Validation Examples

As one example of digging into the model outputs, I looked at model validation accuracy by down and distance. We see accuracy (roughly) improve as down and distance increase - where passing decisions should get more predictable.



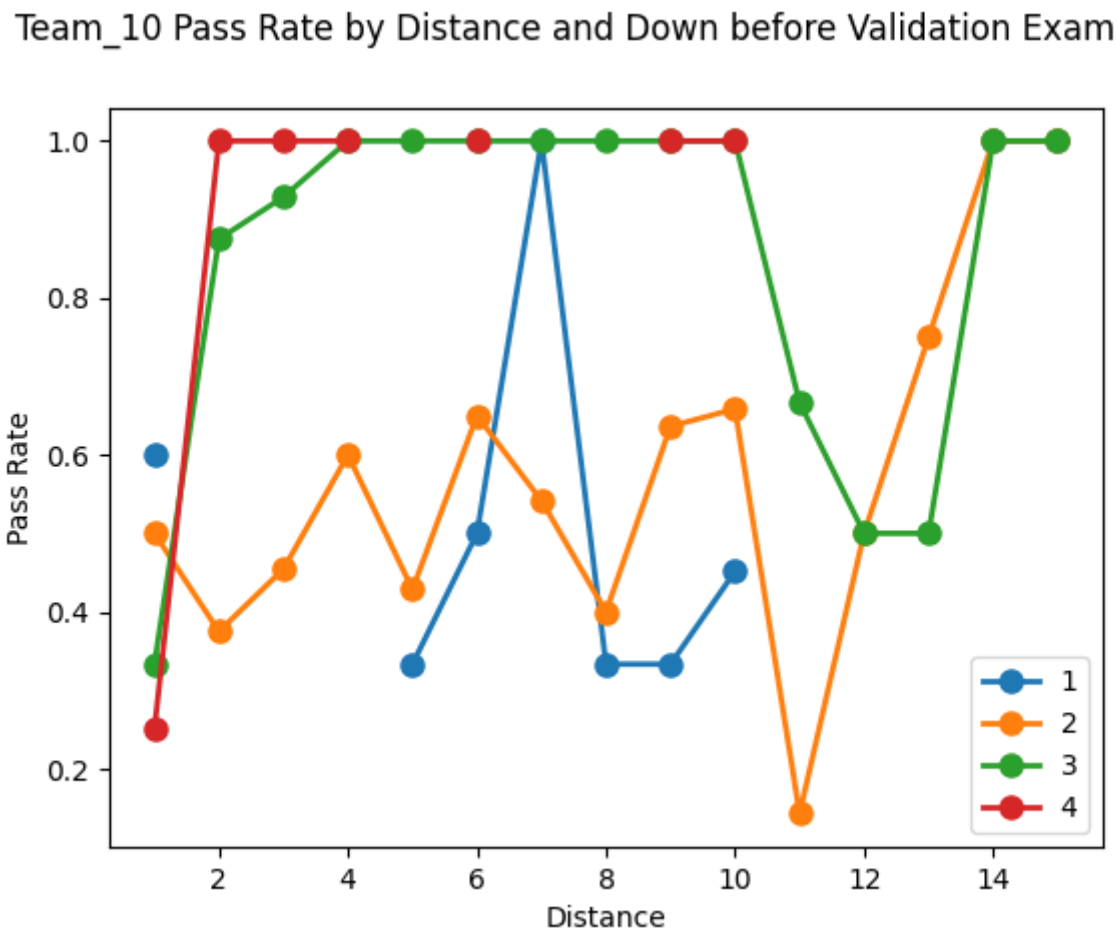
I was curious to see why the model performed so poorly for 1st and 3 situations. In general teams pass about 35% of the time in this situation (can be seen in the exploratory data analysis), and this was also exhibited in the training data (32% of the time). There were only six validation samples with this specific criterion, with four of them ending up as passes. The model predicted run for all of them.

prob_RUN	prediction	RUNPASS	correct	GAMEID	PLAYID	WEEK	QUARTER	SCOREDIFFERENTIAL	SCORE	DISTANCE	DOWN	FIELDPOSITION	DRIVE	DRIVEPLAY
0.658726	0	1	0	19715	4184540	8	2	0	7.07	3	1	3	5.0	1.0
0.578476	0	1	0	19715	4185252	8	4	3	21.24	3	1	3	NaN	NaN
0.584692	0	1	0	19721	4206617	8	1	0	0.00	3	1	3	1.0	10.0
0.681615	0	1	0	19722	4207138	8	1	-7	7.00	3	1	3	3.0	3.0
0.662808	0	0	1	19729	4212291	8	1	0	0.00	3	1	3	2.0	2.0
0.653906	0	0	1	19729	4212485	8	2	0	7.07	3	1	3	5.0	11.0

It's hasty to draw broad conclusions from such a small sample, but one interpretation could be that the model isn't accounting for much outside of the basic game state. Expicitly, once we know the game information like down and distance, the model doesn't have much to draw on for discriminating if a specific

example will be a run or pass, so it defaults to the most likely from the base average. This is further evidenced by the fact that the engineered features realized little performance gains. One solution to this may be better encoding team running and passing tendencies.

The first two examples were from GAMEID=19715 from offensive team Team_10. If we look at all of the team's previous samples before this play, we see that they have passed more often on earlier downs (compared to the graph we say in the exploratory data analysis):

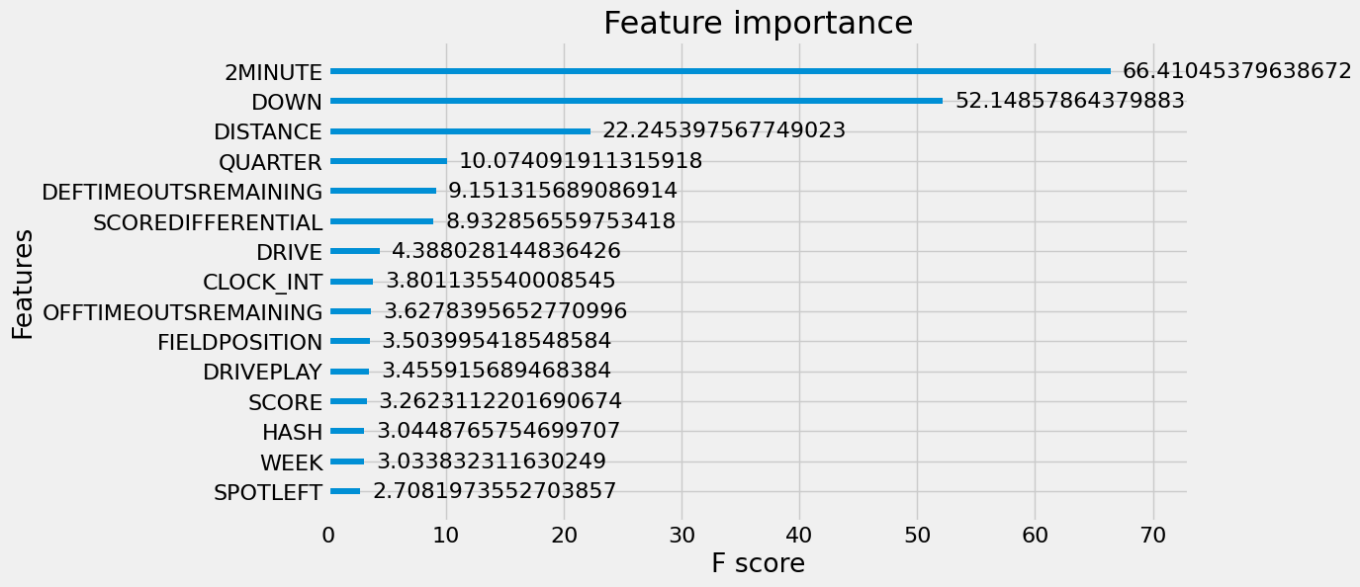


While on average teams pass less than 50% of the time when within five yards in first or second down, Team_10 has passed way more frequently in these situations. This could have indicated that they'd be more likely to pass in the two misclassified validation samples. Better capturing these tendencies in the feature set could help the model get these kind of examples right in the future.

Feature Importance

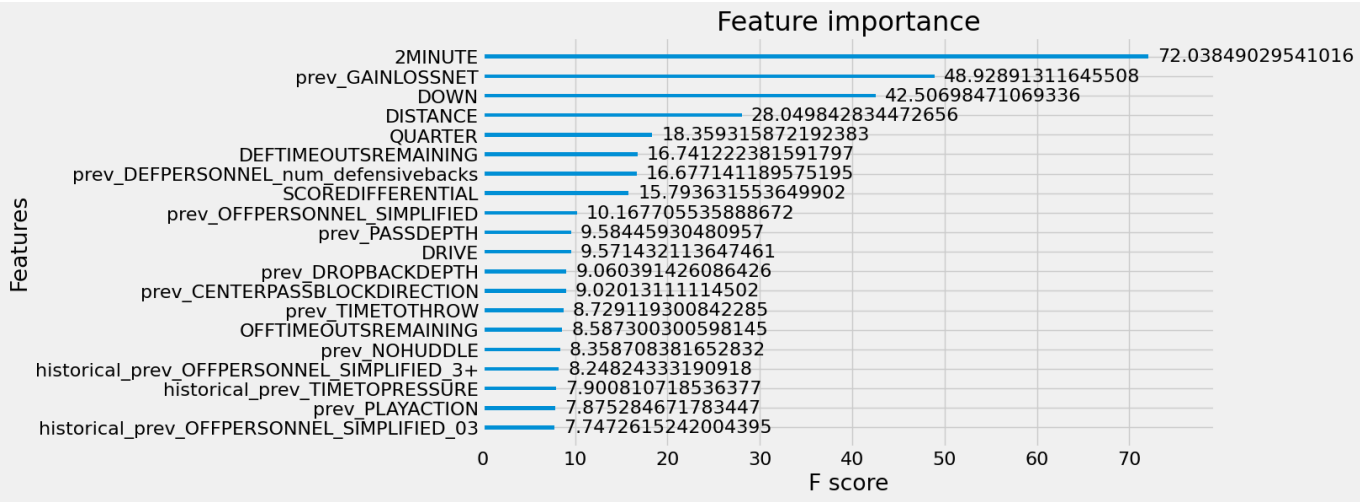
Gain

A feature's gain importance is the improvement in accuracy brought by a feature to the branches its on; i.e. how useful the feature is to classify the outcome. Below is a graph of the top features by gain for the base model. The two minute drill feature is by far the largest, followed by down and then distance.



While we might expect score differential and time remaining to have a larger importance than otherwise shown, that importance may be being distributed to the two minute drill feature. The most important aspects of score differential and time are when the game is close and running out of time, which significantly overlaps with the two minute feature. Thus if the model is already accounting for this information through the two minute feature, the remaining predictive for each is much less. This is in contrast to the other higher features like down and distance, which are not as directly correlated with other features that can take away some of their predictive power.

Below is the plot for the final model's top 20 features by gain. A lot of the variables at the top stay the same, which makes sense as we should still get a lot of mileage out of knowing the high-level basics like down, distance, quarter, score, and timeouts remaining. Previous net yards gain is the new second most important feature, while the number of previous defensivebacks and previous offensive personnel used are among the top ten most important.

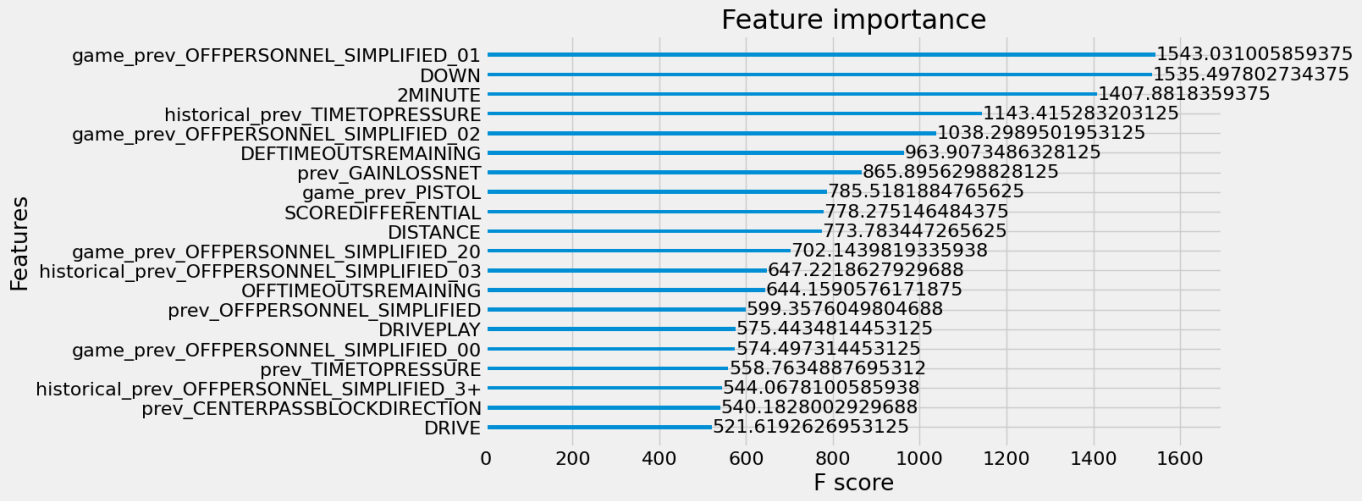


A lof of the important features are descriptions of a previous pass: previous pass depth, dropback depth, time to throw, play action. I wonder how much of this is the predictive value of these features or just them acting as a simple proxy for if the previous play was a pass or not. I lean to the former, but one of the next things I would look at are how passing rates change based on previous pass calls. For example, if you just called a play designed for a long throw, are you more likely to pass again or run? Does it depend on the success of the pass? If there are visibile trends there, this would support these strong features predictive power.

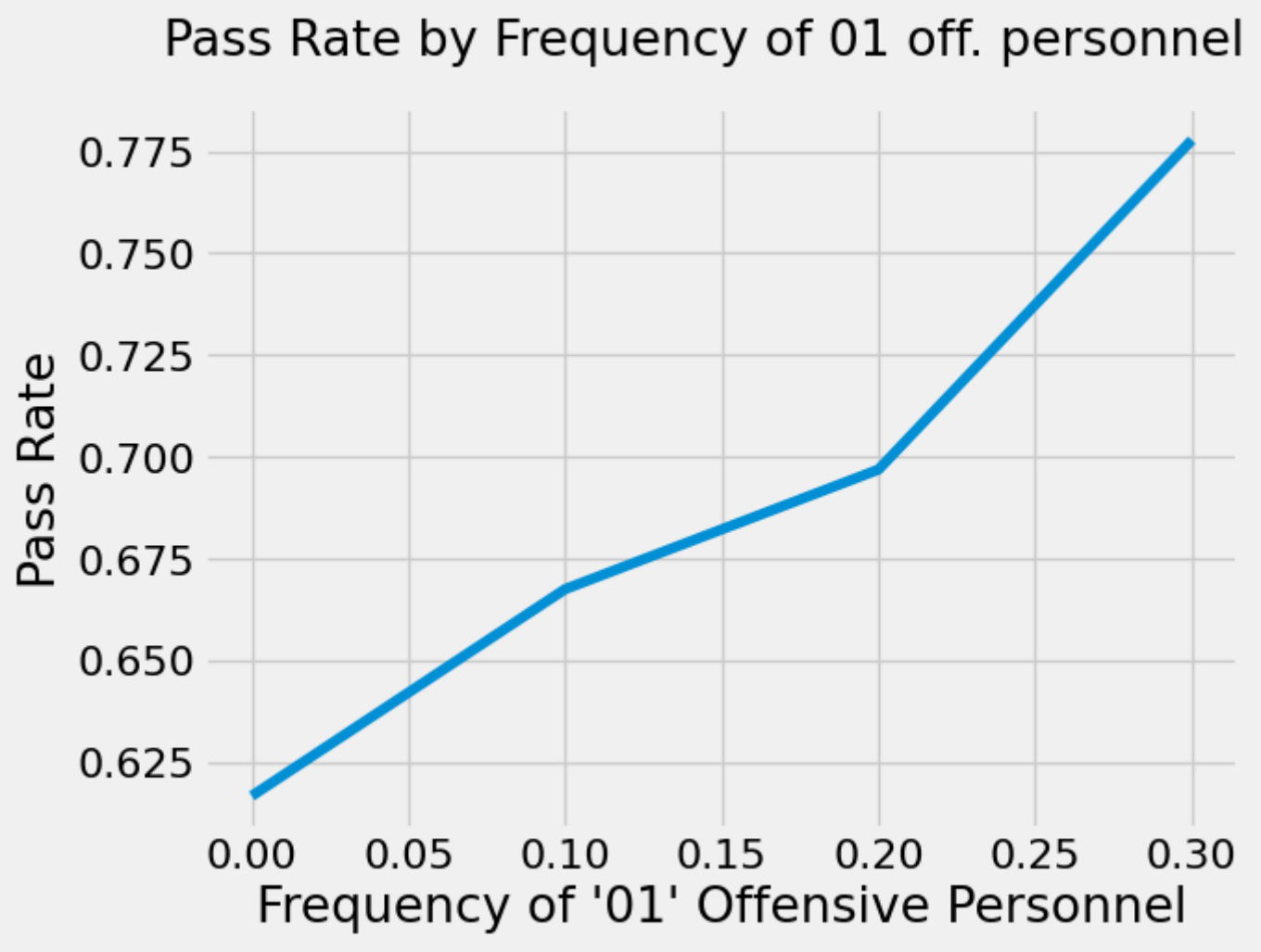
Coverage

We can also look at the top features by coverage. Instead of telling us which features were important to forecasting the call, this tells us the extent of which a feature is used in a model's decision-making process, regardless of its predictive power.

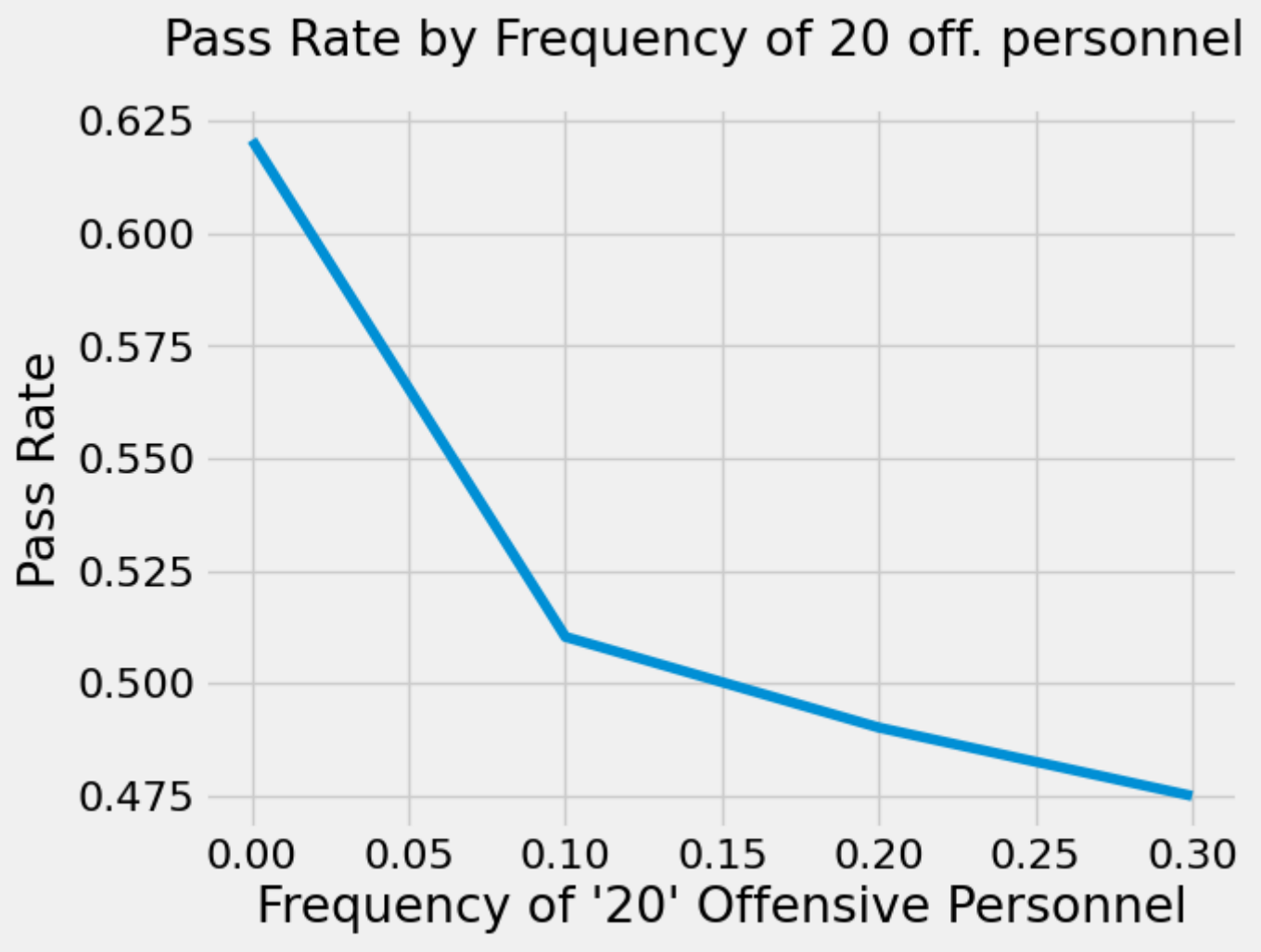
Here we see a stark difference - the base features are not nearly as clustered at the top (though still well represented). The offensive team's rate of using '01' personnel during the game is the most important feature, with '02' and '20' offensive personnel rates being high as well. Historical time to pressure is 4th most importance while a team's rate of using pistol formation in the game is 8th most.



This trend gives us insight into one facet of how the model is making decisions - it is using features that describe the offensive team's formation trends (personnel groupings) and ability to defend against incoming defensive players (historical time to pressure, previous time to pressure, previous center pass block direction). It's easy to hypothesize why these descriptors may be useful and also shows up in the data:



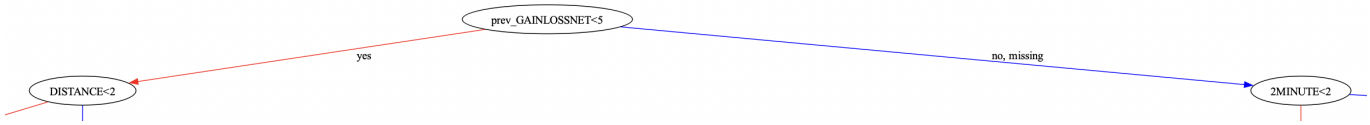
A team's pass rate goes up as their game rate of '01' personnel usage increases. The inverse holds for '20' personnel - a team is more likely to run the ball if they've used '20' personnel a lot during the game.



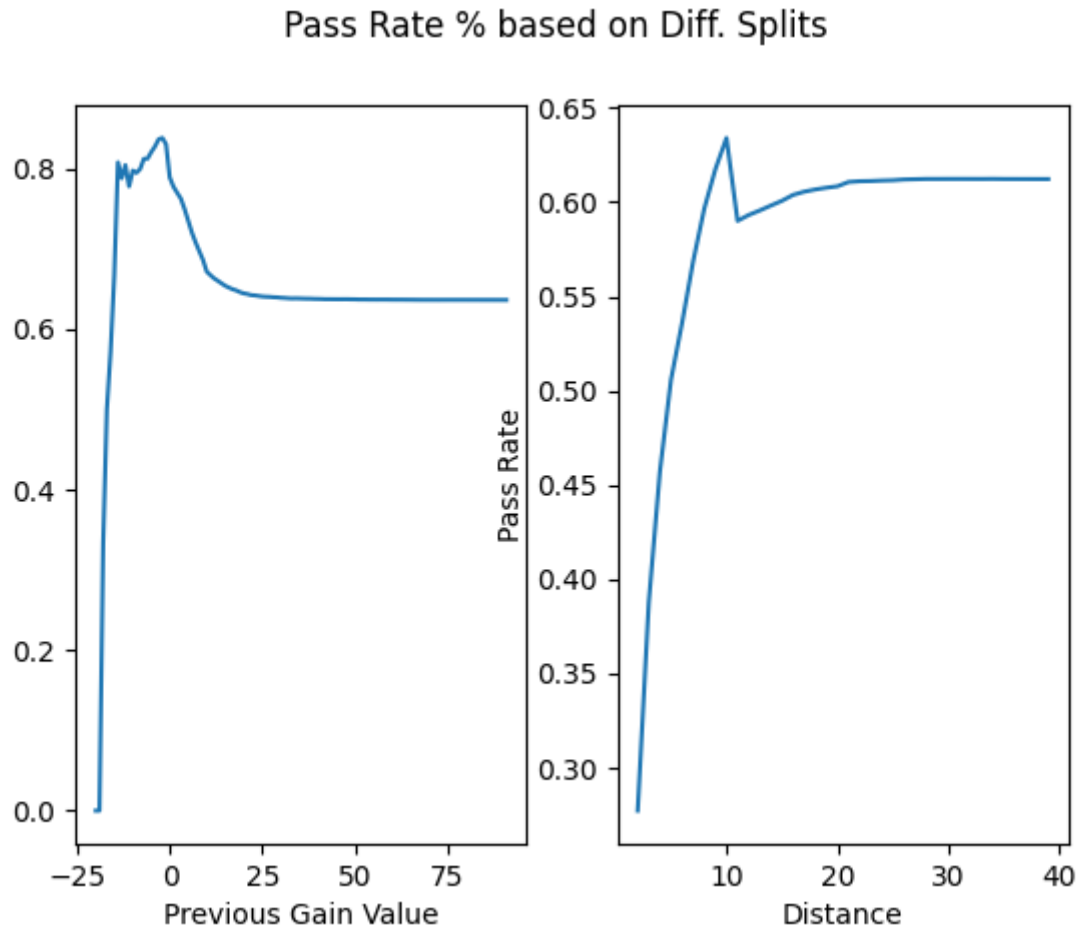
Tree Structure

We can also get an idea about some of the relationships learned by the model from examining the tree structure. Since this is a boosted model consisting of hundreds of trees, we cannot examine every decision tree, nor will any specific one contribute an overwhelming influence to the final prediction. But we can still look at the first tree.

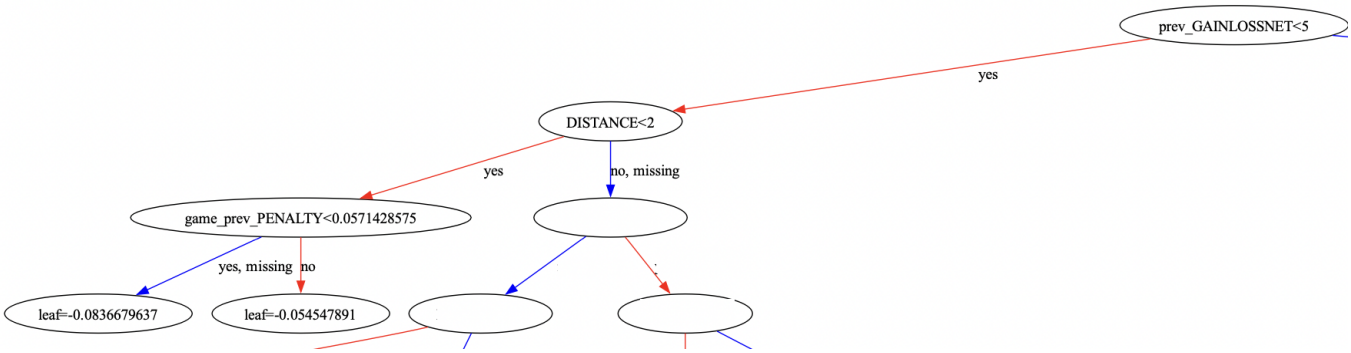
We saw that previous net gain/loss in yards was the second most important feature, and in the first tree it is the root node. The splitting decision is whether or not the previous play gained five yards.



The likely reason this decision was made for the root node is that other important features like down and distance have more complicated interactions between them. For instance, in the exploratory data analysis we saw pass rate by distance significantly differ depending on if its before third down or not. If we solely look at distance without extra features like down, then previous yards gain is a more discriminatory cutpoint: we can draw a clear vertical line in the previous gain graph (left) as opposed to the distance graph (right).



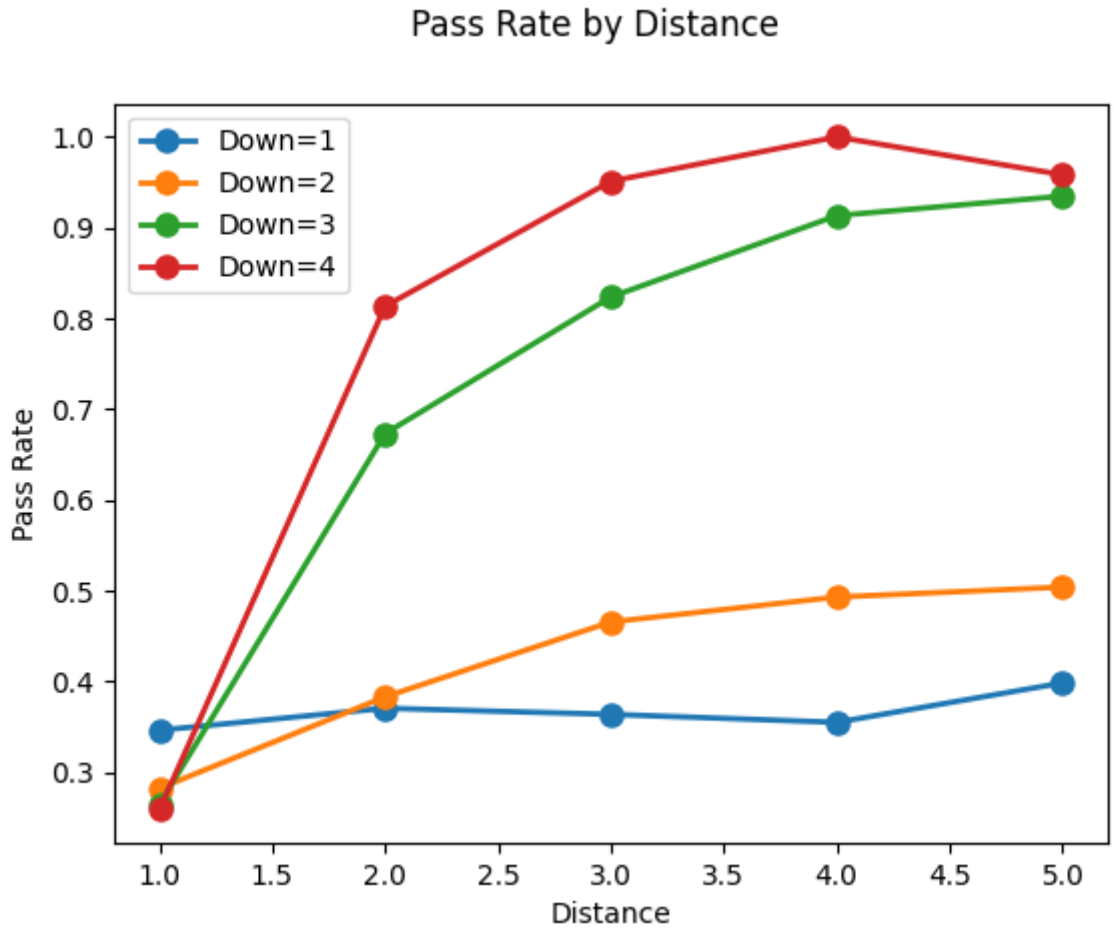
If we keep following the tree left, our next split is "Distance < 2" followed by one more split that ends in two leaves with prediction scores of -0.083 and -0.055 (while they're not exactly log odds, you can think of them as proportional to probabilities, so the lower the score the lower the probability of a pass).



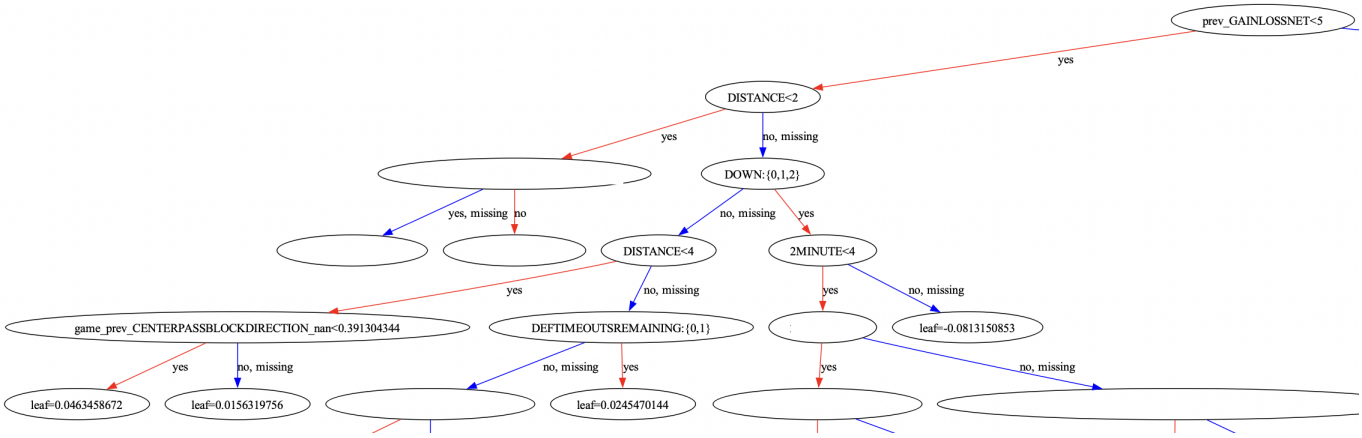
At a high-level this is telling us that when a team gained less than five yards on their previous play and is less than two yards from a first down, the model is more likely to predict a run than a pass. This aligns with our intuitions that a team is more likely to run when so close to a first down. This is confirmed empirically as teams ran the ball 72.3% of the time when less than two yards from a first down.

We should note that this tendency is not affected by the previous yards gained by a team. Teams ran the ball 71.7% of the time when less than two yards and gaining less than five yards in their previous plays - effectively no difference. Instead this is more indicative of the general trend in this tree to use 'Distance < 2' as a powerful split early in branches, regardless of the preceding node. The reason for this is evident in the exploratory data analysis, where we inspected pass rates based on down and distance. While pass rates

significantly diverged as a function of distance conditioned on downs, all pass rates were low for distances less than two, regardless of down.



Speaking of this relationship...this is exactly what this side of the tree captures. If distance is greater or equal to two, the next split is if its before third down or not.



We can see that the leaves on the no side of the 'DOWN: {1,2}' split have positive scores - indicating a higher probability of a pass. The leaf on the yes side of the 'DOWN: {1,2}' split have negative scores, i.e. more likely to run, which is what we saw in the above graph.

Future

To do:
Implement sequential architecture
Historical Pass/Run tendencies
Previous outcome (Pass, Run, Sack)
Permutation importance or SHAP values