Capstone Two Report
AUDL Ultimate Frisbee: Score Margin and Win Probability Prediction

Background

The American Ultimate Disc League (AUDL) is a men's professional Ultimate Frisbee league established 2013 in North America, currently consisting of 24 teams across four regional divisions. The AUDL modified some standard rules of the Ultimate Frisbee to increase marketability. Most importantly, games are played in four timed-quarters, instead to of a given score. If both teams have the same score at the end of regulation, then they play up to two overtime periods to declare the winner (two ties were found in the data). Sport specific discussion will be limited in the main body of the report, but the reader can refer to Appendix I – Statistics Glossary for a description of game flow, and the context input, target, and engineered features within it.

Problem Statement

Eleven years of game results, summary statistics, and player data are now available with the conclusion of the 2023 season. This project seeks to use the most basic game summary statistics to predict the outcome of the game: the winning team and the difference in the two teams' scores. The final models should provide a basis for understanding the remaining summary statistics, and the data pipeline should exemplify collecting, cleaning, and exploring AUDL data for future studies.

AUDL Data

Collection

Game statistics were collected from the AUDL website using their REST API (documentation). For this project, one endpoint was used with date specification to retrieve a list of all games played and their game IDs (URL). Each game ID was used with another endpoint to collect game statistics (example URL). Future work will make use of other endpoints to provide more detailed game statistics. The JSON returns were read into pandas DataFrames, and cleaned data was persisted in the Parquet format. Each record is a unique game, specified by its game ID, and has seven descriptive features and thirty numerical features. Records for 1,526 games were collected. Refer to Appendix I – Statistics Glossary for detailed feature information.

Cleaning

Data was thoroughly cleaned, explored, and checked prior to model building [1]. Descriptive features (date, location, team names, etc. . .) were not used as final model inputs for this project's scope, but were helpful in evaluating results. Some of these features were used to demonstrate one-hot-encoding and dummy feature creation. For example, **week** was encoded and could be used as a season chronology feature.

Missing values were assessed, and each feature was checked to ensure that values of **0** made sense and were not null placeholders. For example, if either **throws** or **completions** were not recorded (missing or 0), then the other was also marked as suspicious. Following a similar process of feature-specific checks and explorations, other games were found with faulty records. Two game records were missing all features. Duplicate games were not present, but the data checks yielded some dubious records. It is very unlikely that both **home** and **away** teams have equal values for all statistics in a game. Such records likely indicate that one of the two team's stats were not recorded.

Outliers were assessed manually and checked with background knowledge. Automated detection was explored using Local Outlier Factor and IsolationForest algorithms provided by scikit-learn. Most values made sense given their associated features, or were found in games with incomplete data. For example, records with the highest values of **home blocks** were believable, as **away turnovers** and **away throws** were relatively high. Rudimentary visualizations of the outlier detection algorithms indicated possible utility as a final data cleaning step. Scaling data prior to detection seemed to improve detection for Local Outlier Factor, but not for IsolationForest (graphs in repository).

Selection

Before removing records with missing or faulty data, "basic" features inputs were identified. Because the scoring team will almost always starts the next point on defense, the final score of the game can be analytically determined from a number of features (Appendix I – Statistics Glossary). Therefore only **throws, completions, blocks,** and **turnovers** were kept as basic features and used as model inputs**.** The other features will not be presented below, but can be seen in the EDA notebook. Culling non-basic features left a much greater portion of records with complete data, especially as some game statistics were not introduced until 2019 [2].

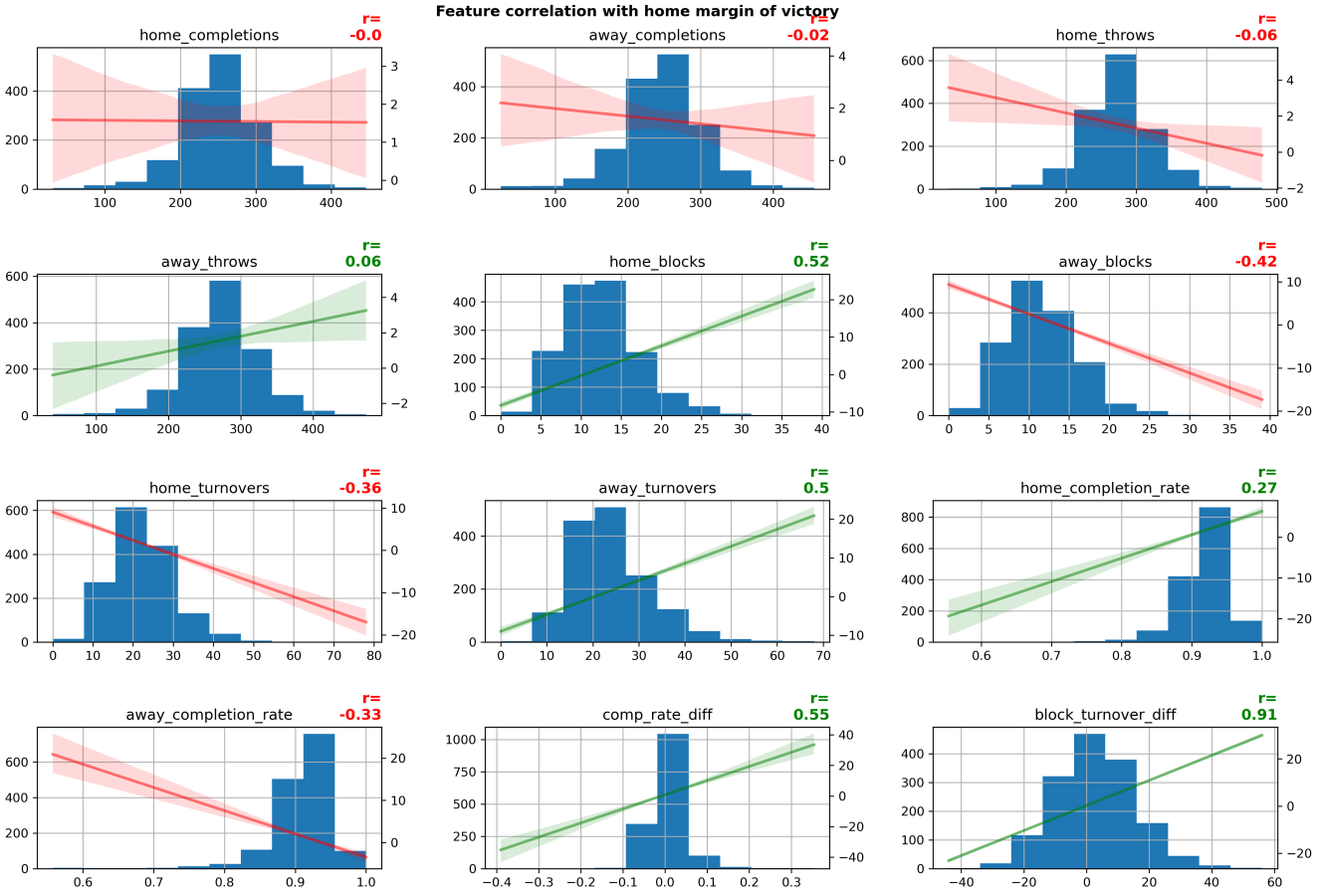
Target Features, Feature Engineering

Target features were defined using **home score** and **away score**. The continuous target for regression models, **home margin**, is the difference of the two. The binary target for classification models, **home win**, is true if **home score** is greater than **away score**.

Basic features were combined in an effort to better correlate with targets and to maintain interpretability from an Ultimate Frisbee context. **Completion rate** was calculated for each team, and the **completion rate difference** between the teams was also added. Features describing a change in possession were aggregated, **blocks** minus **turnovers**, for a given team; and then the difference between teams, **block-turnover difference** was added. Feature distributions before and after cleaning steps were examined, and feature correlation with each target variable was assessed [3]. Refer to Appendix I – Statistics Glossary for more details.

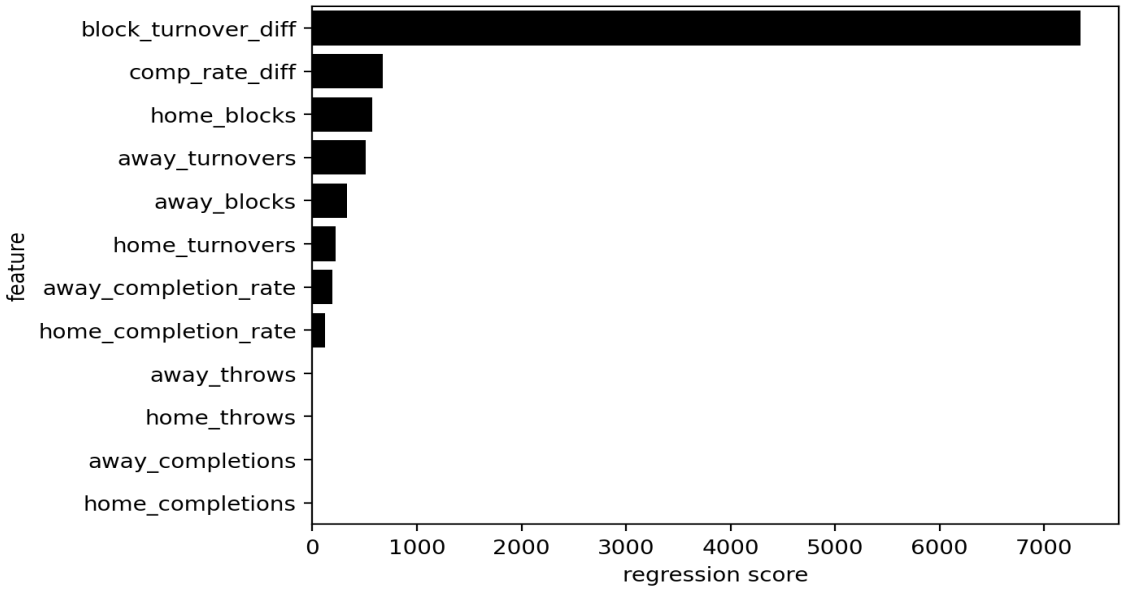
Exploration

The histograms below show that most features are distributed normally, although others present obvious skew. The **home margin** regression lines and correlation coefficients with suggest benefit to the engineered features over their components. **Completion rate** correlate more strongly to **home margin** than **throws** or **completions** alone, and the **completion rate difference** correlates more strongly than either completion rate alone. It also has a more normal shape than its related features. A similar pattern is followed for **turnovers**, **blocks**, and **block-turnover difference**. Intercorrelations between the features are shown with a correlation heatmap [4]. The strongest relationships stem from inherent relationships, such as **throws** and **completions**. Collinearity was addressed during pre-processing for certain models. Feature interactions were further explored for the final model [12].



x-axis: feature values *left y-axis: histogram bin count* *right y-axis: home margin target*

F-statistic and p-values were used to score features for regression and classification [5]. The regression scores below highlight the importance of features involving possession change and show that **completion rate difference** brings throw and completion features into relevance with **blocks** and **turnovers**.



The final 1,521 records with twelve features were split 4:1 into training (1,216) and testing (305) sets for model evaluation. A random seed was set to ensure consistent splits and repeatable model evaluation across studies. Target feature stratification was not imposed with the split, but was confirmed after the fact [6]. Data was not normalized prior to model evaluation, as the effect of normalization was studied for each model.

Model Comparison

A number of machine learning algorithms were evaluated, and optimal pre-processing conditions and tuning parameters were determined for each one. This report will focus on the regression models predicting **home margin**. Classification models were evaluated with a similar process (notebook). Initial studies are briefly described, then evaluation methods and criteria are discussed in detail for the final model selection step. A blend of the tuned GradientBoosting and CatBoost regressors was selected for the final deliverable.

Initial studies explored feature selection (top features selected using scores above) and z-score data normalization. K-nearest-neighbors (kNN) and linear models were best with seven or eight out of the twelve features [7]. Of those, the kernel ridge model and kNN performed better without normalization. The boosting and bagging ensemble models were best with all features, and normalization was helpful or did not affect performance of the better models. Automated outlier removal during cross-validation did not improve model performance. The following models were selected from these studies and progressed for direct comparison.

Model Descriptions | Final Deliverable

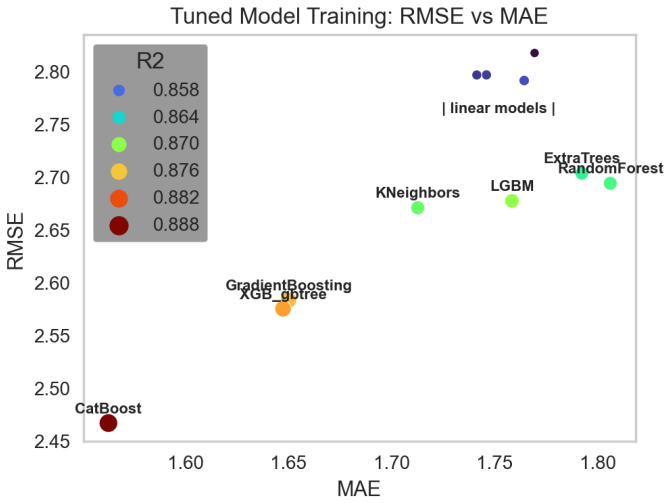
Name [name in graph]	Pipeline (pre-processing)	Package
Ridge	Select 8 features, normalize	scikit-learn
Kernel Ridge	Select 7 features	
StochasticGradientDescent (SGD)	Select 7 features, normalize	

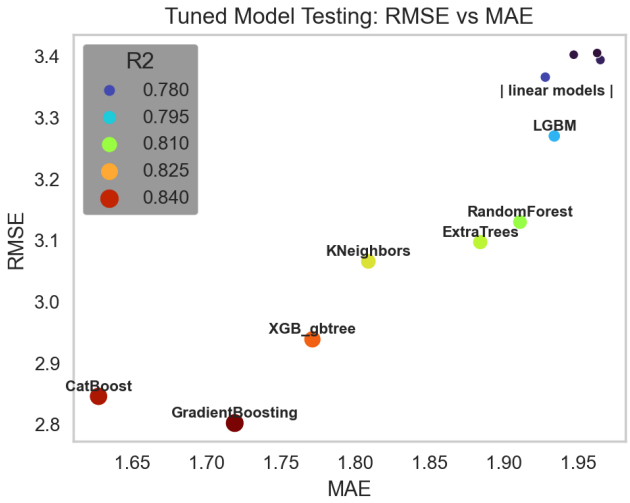
kNN [KNeighbors]	Select 7 features	scikit-learn
RandomForest	Normalize	
ExtraTrees		
GradientBoosting		
LightGBM [LGBM]		LightGBM
CatBoost		CatBoost
XGB (tree) [XGB_gbtree]		XGBoost
XGB (linear) [XGB_gblinear]		
Final Blend (Voting Regressor)	Combination of GradientBoosting and CatBoost pipelines	scikit-learn + CatBoost

Model Metrics

Name	Abbreviation
Coefficient of determination	R2
Root mean-squared-error	RMSE
Mean absolute error	MAE
Mean absolute percentage error	MAPE

Hyperparameters were tuned with **RandomizedSearchCV** from scikit-learn. Some combination of each model's R2, MAE, and RMSE improved with tuning. Tuned models were compared by cross-validation (training) and test set performances. Scoring metrics for models were mostly correlated, although MAPE was not always suitable for testing, as division by zero can occur with the **home margin** target. Training and testing scores were compared for the models to evaluate overfitting. Most models showed a similar trend between the two scoring sets, but LightGBM did not generalize as well as the others [8].





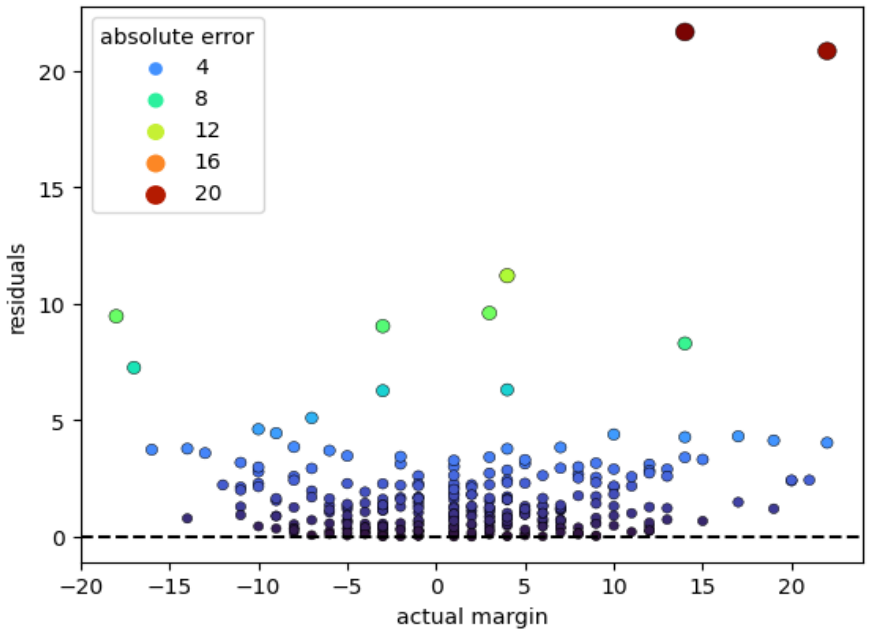
As shown above, there are clearly 3 best models in terms of R2, RMSE, and MAE: CatBoost, GradientBoosting, and XGB (tree based boosting). Blends of the top models, VotingRegressor meta-estimators, were then evaluated to see if performance could be further improved. Tabular results of these models and selected blends are shown below. The CatBoost+GradientBoosting blend provided

the best result, as it improved some aspects of each of its component models. It may harmonize the better generalization of GradientBoosting and the lower error of CatBoost. In practice, the residual analysis was almost identical for the final model blend and the CatBoost model alone. See the final model submission file for all model parameters and metrics.

	R2	RMSE	MAE	MAPE
voting_GBR-Cat	0.849	2.789	1.644	0.438
GradientBoostingRegressor	0.847	2.801	1.719	0.459
voting_GBR-Cat-XGB	0.847	2.809	1.663	0.449
CatBoostRegressor	0.843	2.845	1.627	0.435
XGBRegressor	0.832	2.938	1.771	0.496
KNeighborsRegressor	0.817	3.065	1.809	0.504

Final Results

The final model and its building efforts satisfy the problem statement. Game summary statistics provided from the AUDL's API were used to create reasonably accurate models to predict the team's difference of scores and the winning team. Predictions could be improved, but performance was limited by design. Only the most fundamental statistics, and their meaningful combinations, were used for model building, and now their importance to winning can be interpreted from the final model [11]. Additionally, the model's worst predictions are probable indications of invalid records that should have been removed during cleaning [9] [10].



More importantly, a repeatable data collection process and a basis for its cleaning and evaluation were established. Obvious indicators of invalid data have been identified and faulty game records can be documented to streamline future work. Extensions of the final model and future studies are proposed in the next section.

Future Recommendations

Given the performance of the final model, it is worth expanding the work to include team and match-up specific predictions. Team labels were ignored for this analysis, but the model's errors were not team independent [13]. New models could target the input features for the current model, such that the outcome of future games, say the upcoming 2024 season, could be predicted. The data pipeline can be adapted for online learning as the season progresses.

The majority of features returned in the game summary statistics were not used for the win margin and win probability models. While these features were not desirable as aggregate game statistics, increasing their granularity to a per-point or per-possession basis will allow deeper exploration. Throws and completions were found to have little to no value for this study, but considering their finer distributions, and what results those lead to, may increase their predictive power.

Much more data is available from the AUDL API, such as roster information for each point and positional tracking (XY) of the disc and each of its events. This work provides a template to collect such data and assess more problem statements.

Appendix I – Statistics Glossary

Introduction

Games are played between two teams. A team is on offense when it has possession of the disc. It can move the disc around the field with throws. An incomplete throw, or holding the disc for too long, results in a change of a possession. If a team catches the disc in the opposing endzone (regardless of which team threw the disc), the point ends with that team scoring. The scoring team then starts on defense and "pulls" the disc to other team, which will start on offense. This is similar to a kickoff in American Football, or like football, if the starting team booted the ball as far as they could to the other team. Games are considered similarly to tennis: the team starting on **offense** is expected to score (hold), and teams often consider their scores on **defense** (breaks) as importantly as the overall **score**.

Not all of the numerical features were used as model inputs, as indicated in the tables. The final score for each game was used to create the **Target Features**, and some of the **Standard Features** were combined to create **Engineered Features.** Features were renamed for clarity and brevity during import and again for this report. See the Data Wrangling notebook for original data returns and feature names. **Descriptive Features** that appear in the figures and discussion are also mentioned.

Also refer to: https://www.theaudl.com/stats/glossary.

Standard Features

The following are provided with each game summary. Statistics are provided twice, once for the **home** team, and once for the **away** team. Some terms have the additional distinction of **offense** and **defense** (O/D), indicating that a given team started the point on O/D, and gained the statistic. A coin toss determines the whether the **home** or **away** team starts on **offense** or **defense** for each quarter. After the first point, the scoring team will be start on defense and the team that didn't score will start on offense.

Term	Description	Insight
Score*	Final score for a given team.	The home and away score together describe the final result. Used for target feature definition.
Throw	Pass attempt. Player throws disc.	The team with the disc changes their position by throwing to one another.
Completion	Successful pass attempt, thrown disc is caught by the same team.	Most throws in the AUDL are successful. Missed throws are significant.
Blocks	The team without the disc gains possession, *forced error(?)*	Unsure about distinction/overlap between blocks and turnovers. Data exploration raised more questions than answers.
Turnover	The team with the disc loses possession, *unforced error(?)*	

Hucks*	Long pass attempt, player throws disc more than *(?)* yards.	Unsure about cutoff. Default distance metric for the AUDL is yards. Hucks were not tracked until 2021.
Hucks Completed*	Successful long pass attempt by the same team.	
(O/D) Score*	Successful point for a team. One team scores in a point, one does not.	Scoring on offense is considered a “hold”, and scoring on defense is considered a “break”.
(O/D) Point*	Simply a count of points played for a team, starting on O/D.	
(O/D) Possession*	Count of possessions each time a team is playing a point. Scoring without a turnover is one possession.	Unlike scores and points, possessions could be considered a “basic” statistic and incorporated in the models. However, it is more valuable in combination with features that were removed.
Redzone Possessions*	Possessions occurring within a certain distance *(?)* of the endzone (scoring target).	Unsure about cutoff. Redzone statistics were not tracked until 2021.
Redzone Scores*	Successful redzone possessions.	

** feature not used as model input*

Target Features

Target features were designed to be independent of team and to avoid double-counting of games. Each game was considered from the home team’s perspective.

Term	Description	Insight
Home Score Margin	Home score minus away score. Continuous target for regression models.	Ties, resulting in a margin of 0, are rare by design of overtime periods. This was not explicitly input to the continuous target.
Home Win (Probability)	Home win is true if the home score is greater than the away score. Binary target for classification models.	Instead of considering ties as a result, the binary classification is maintained by *not* considering ties as a win.

Engineered Features

Some standard features were combined to emphasize their importance on game outcomes.

Term	Description	Insight
Completion Rate	A given team’s completions divided by their throws. Normalizes throws and completions across games.	Imbalance between implied missed throws (blocks, turnovers) and reported throws suggests inconsistent classification.

Completion Rate Difference	Difference between two team's completion rates. Home team minus away team (home team's perspective).	Distribution much less skewed than throws, completions, or completion rate.
*(H/A) Blocks + (A/H) Turnovers**	Overall turnovers for a team. The home team's blocks are added to the away team's turnovers, and vice-versa.	Blocks and turnovers may not be consistently classified.
Block-Turnover Difference	Overall turnover difference for the two teams. Home team's perspective (positive correlation with target features).	Downstream effects of variable classification

** intermediate feature, not used as model input*

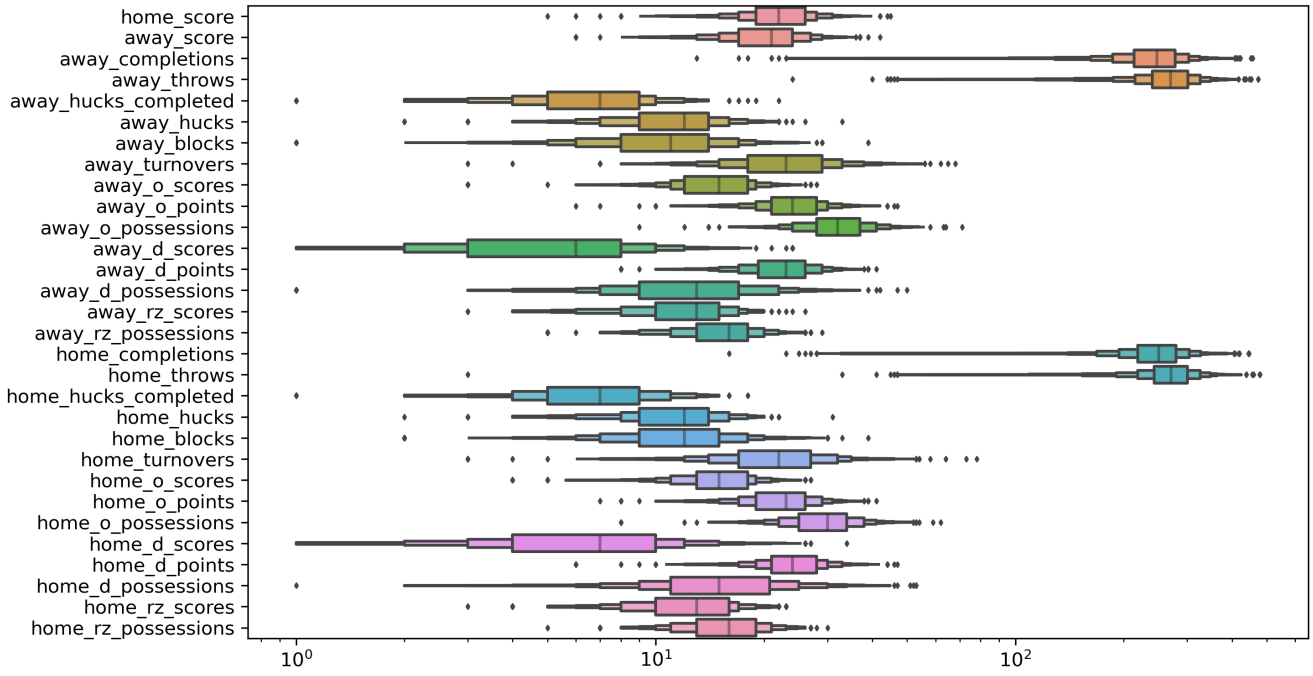
Descriptive Features

A few of the remaining features provided during import are shown below. For future analyses, some of these could be encoded and used for model inputs. For this study, they were simply used to group and analyze results.

Term	Description
Date, Timezone	Timestamp for the game record. Start and end times provided.
Location	Location ID of the game, specific to actual venue and not team.
Week	Week of the regular season or play-offs. Indicator of season progression.
Home, Away	Teams playing in the game.

Appendix II – Additional Figures

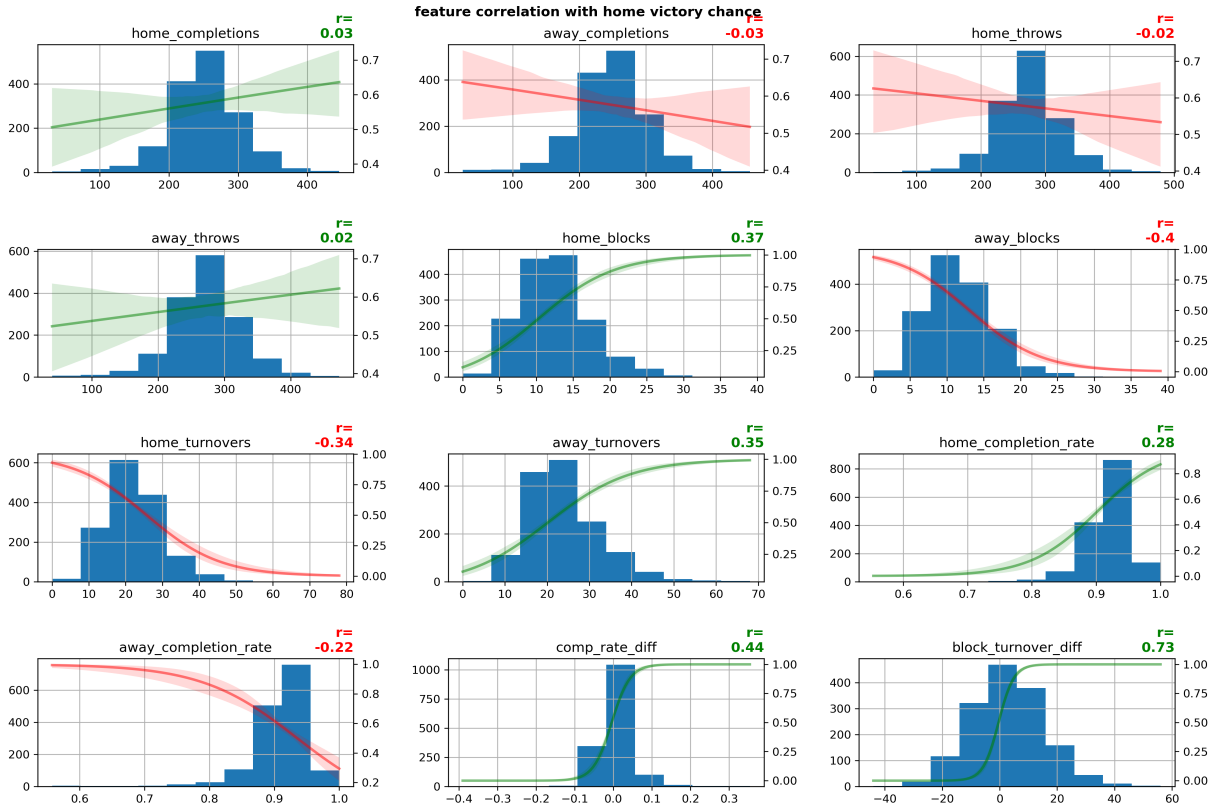
[1] Imported Feature Distributions – Boxplots show feature distributions prior to any data cleaning.



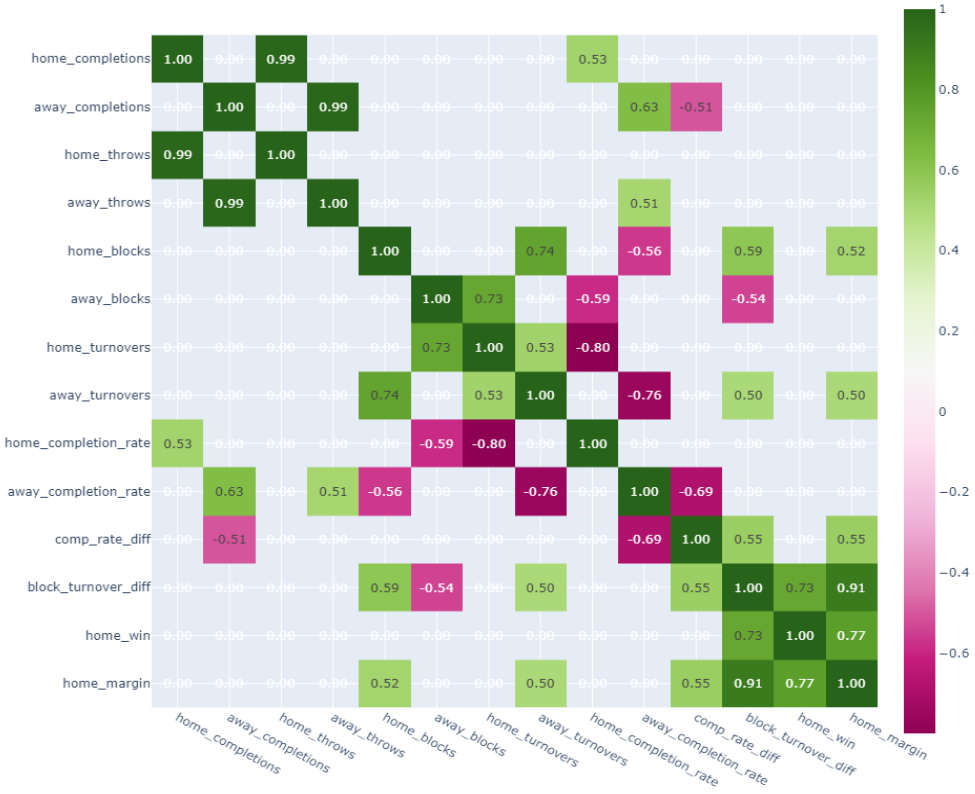
[2] Data Missing by Season – Hucks and Redzone statistics were not recorded prior to the 2019 season.

	2012	2013	2014	2015	2016	2017	2018	2019	2021	2022	2023
away_hucks_completed	100%	100%	100%	100%	100%	100%	100%	98%		1%	
away_hucks	100%	100%	100%	100%	100%	100%	100%	98%			
away_rz_scores	100%	100%	100%	100%	100%	100%	100%	98%			
away_rz_possessions	100%	100%	100%	100%	100%	100%	100%	98%			
home_hucks_completed	100%	100%	100%	100%	100%	100%	100%	98%	1%		1%
home_hucks	100%	100%	100%	100%	100%	100%	100%	98%			
home_rz_scores	100%	100%	100%	100%	100%	100%	100%	98%			
home_rz_possessions	100%	100%	100%	100%	100%	100%	100%	98%			

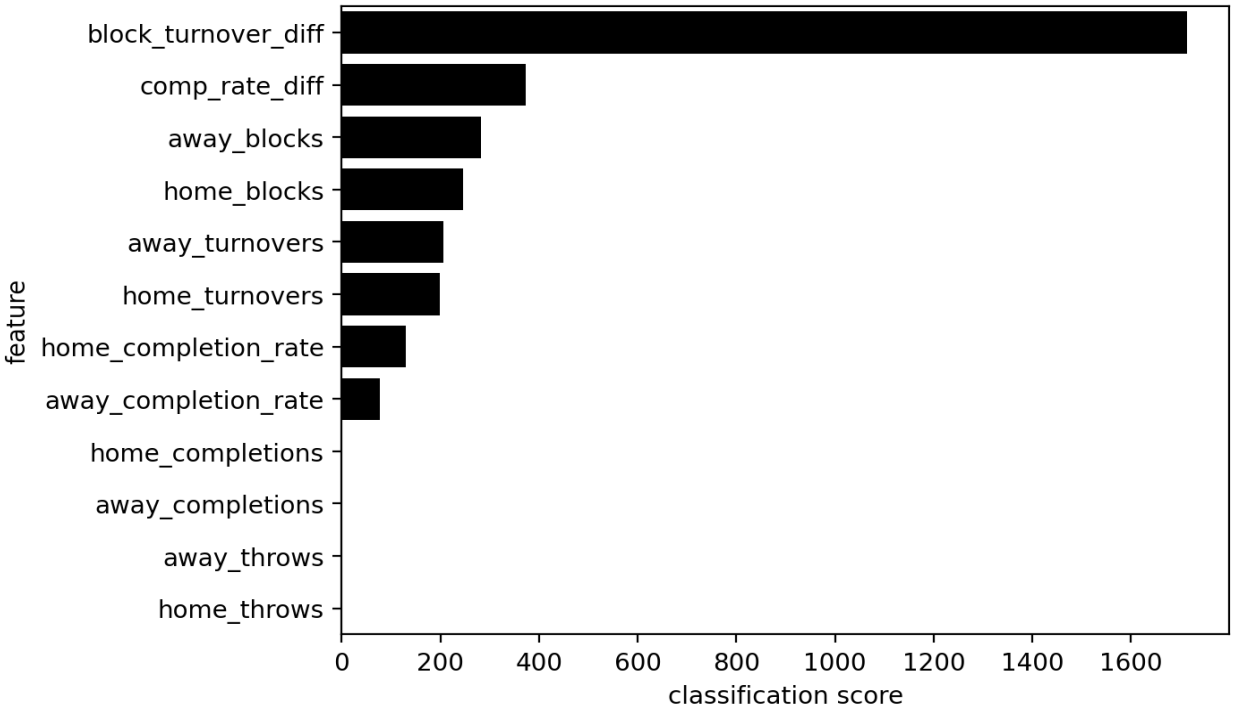
[3] Cleaned Feature Distributions, relation to Home Win – Histograms overlaid with logistic regression for **home win**.



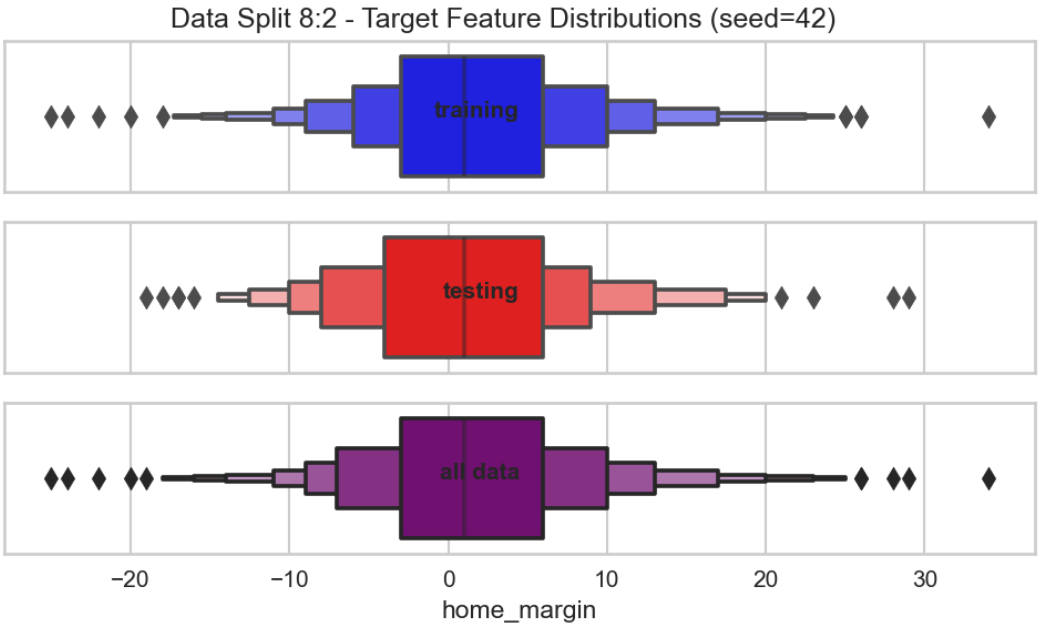
[4] Feature Correlation Heatmap – Color coded intercorrelation coefficients for values > |0.5|



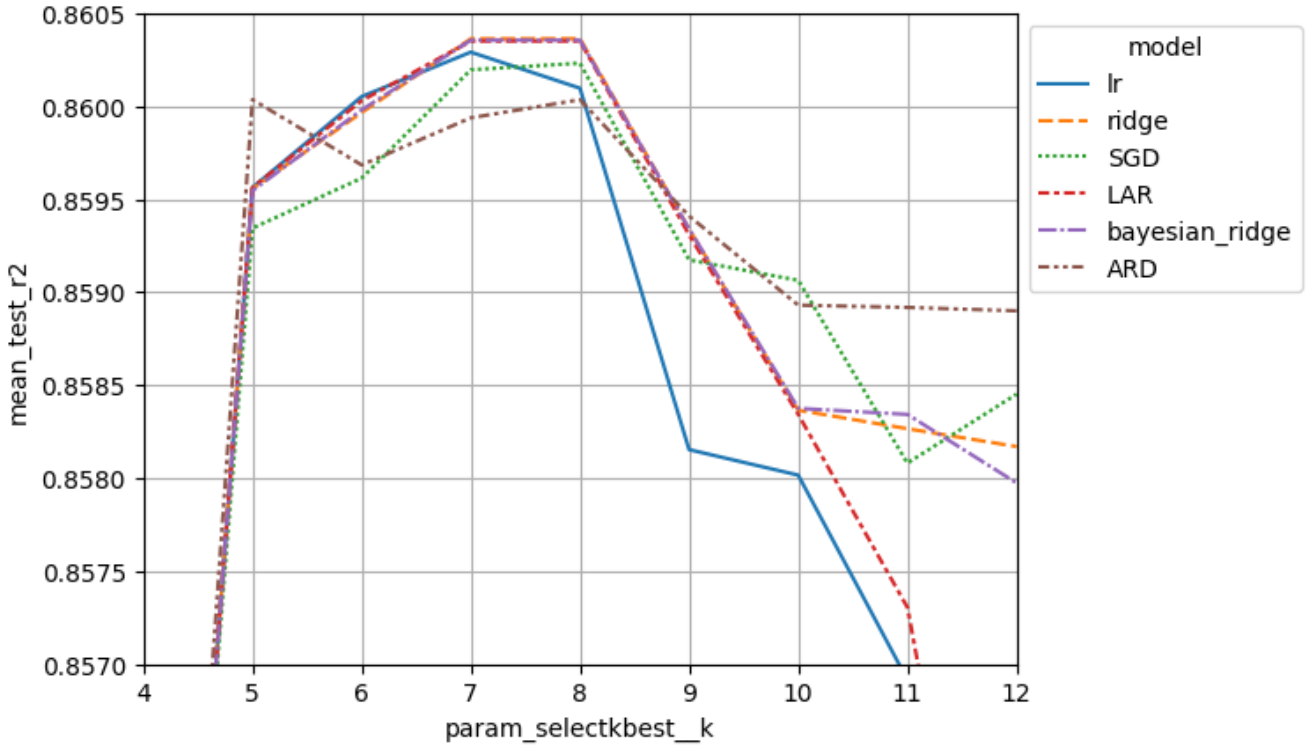
[5] Feature Significance for Home Win – Results from sklearn's SelectKBest with *f_classif* score function. Graph in report used *f_regression*.



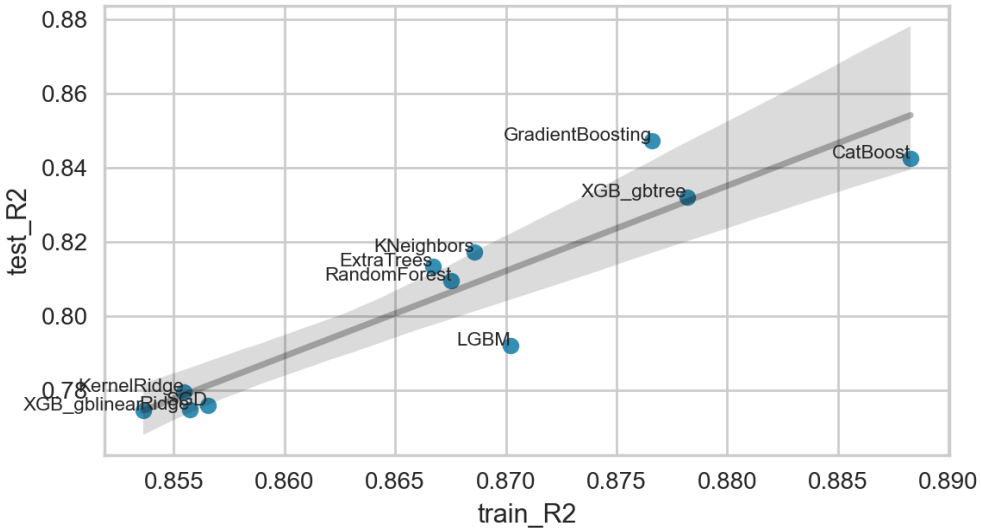
[6] Stratification Check – Target feature distribution in train and test sets.

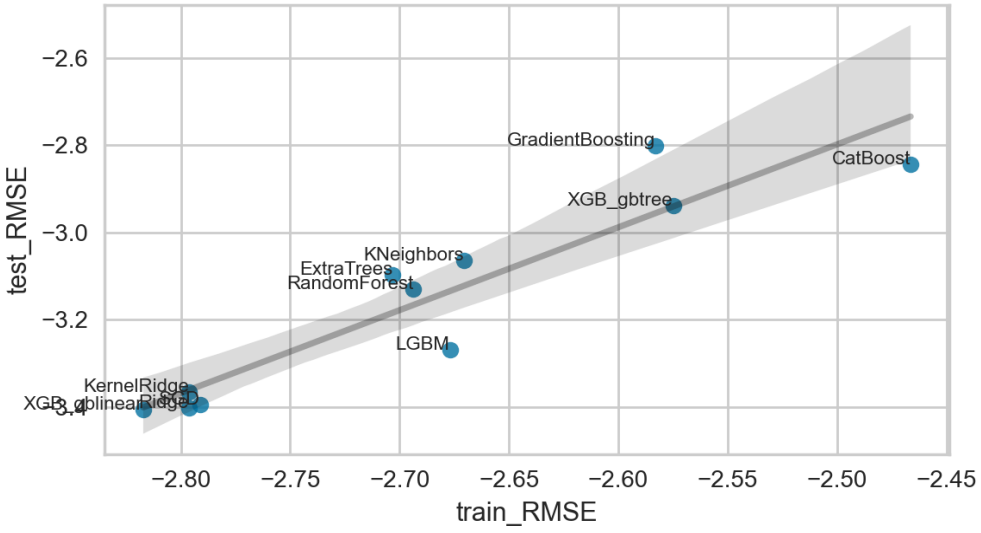


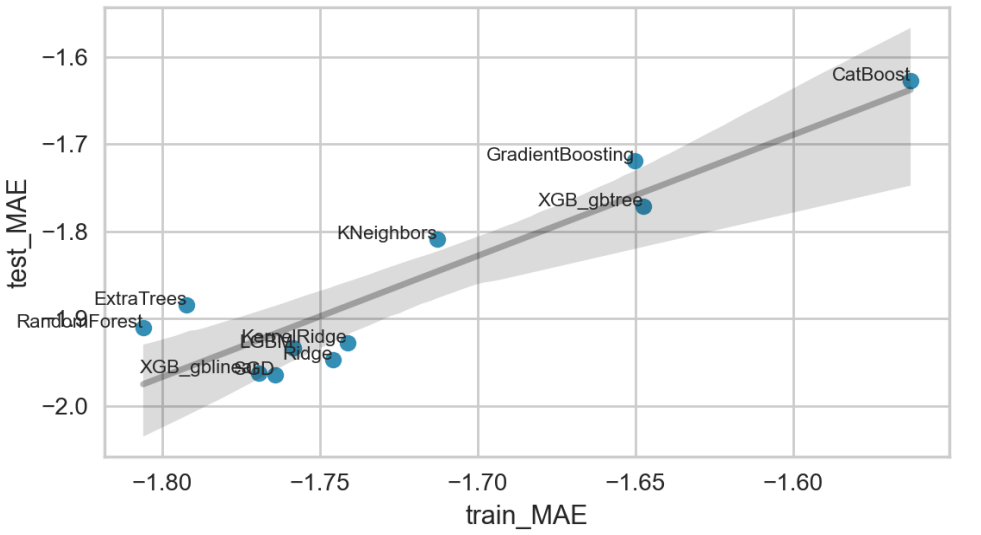
[7] Linear Model Feature Selection – Model's R2 vs number of features selected, zoomed. Only ridge and SGD discussed in report. Other linear models show similar behavior.



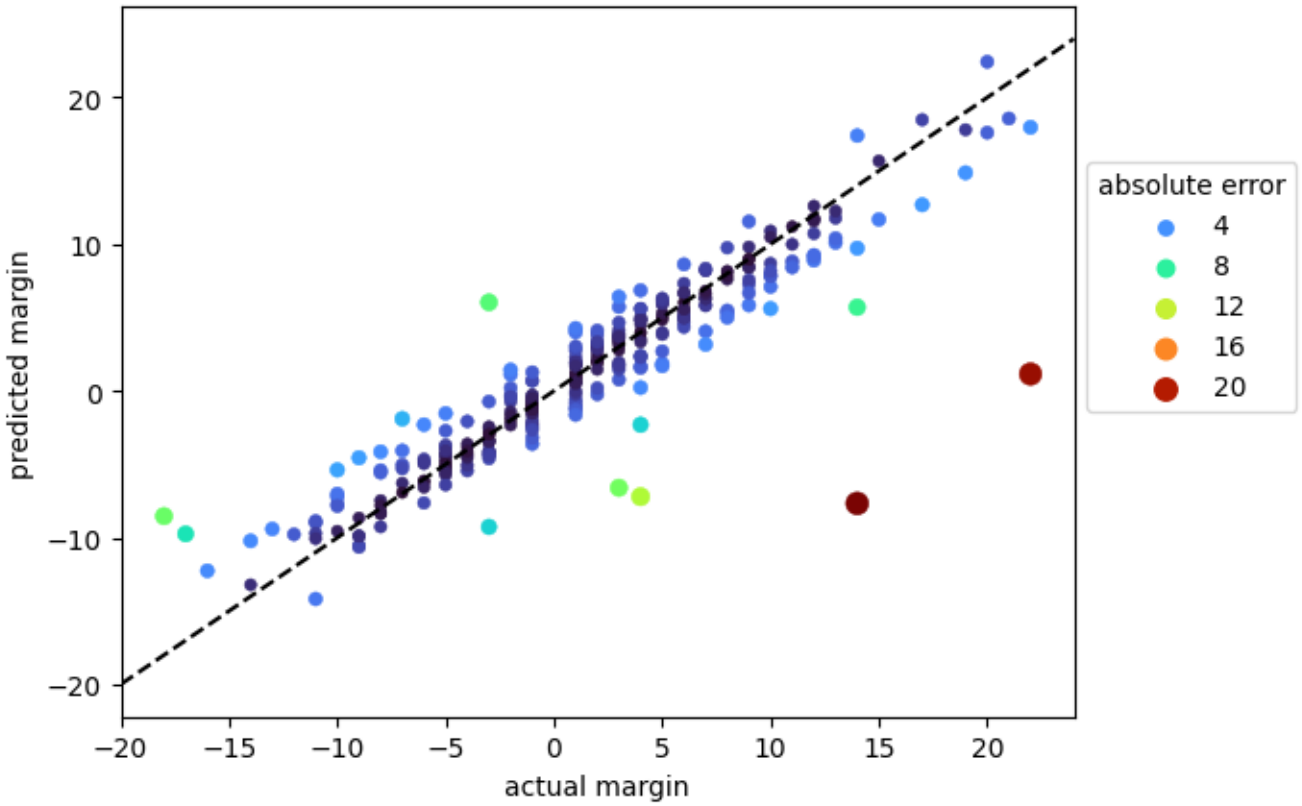
[8] Overfitting Analysis – Model test vs train scores for select metrics. ↗ indicates better performance



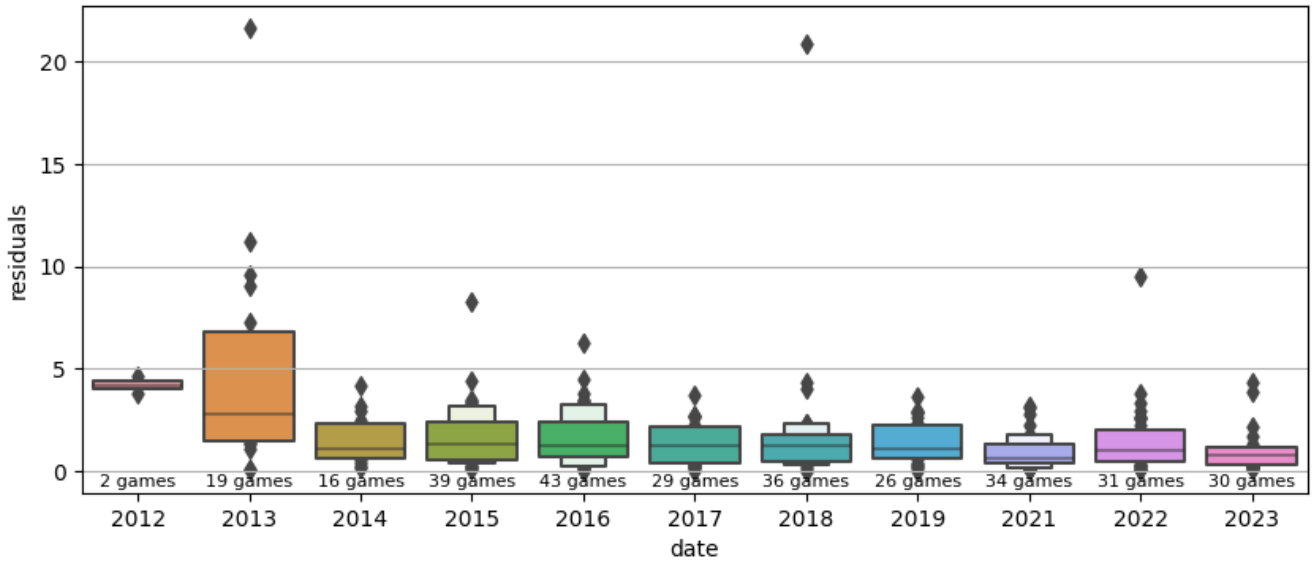




[9] Residual Analysis – Predicted margin vs actual margin. Points colored and sized by absolute error.

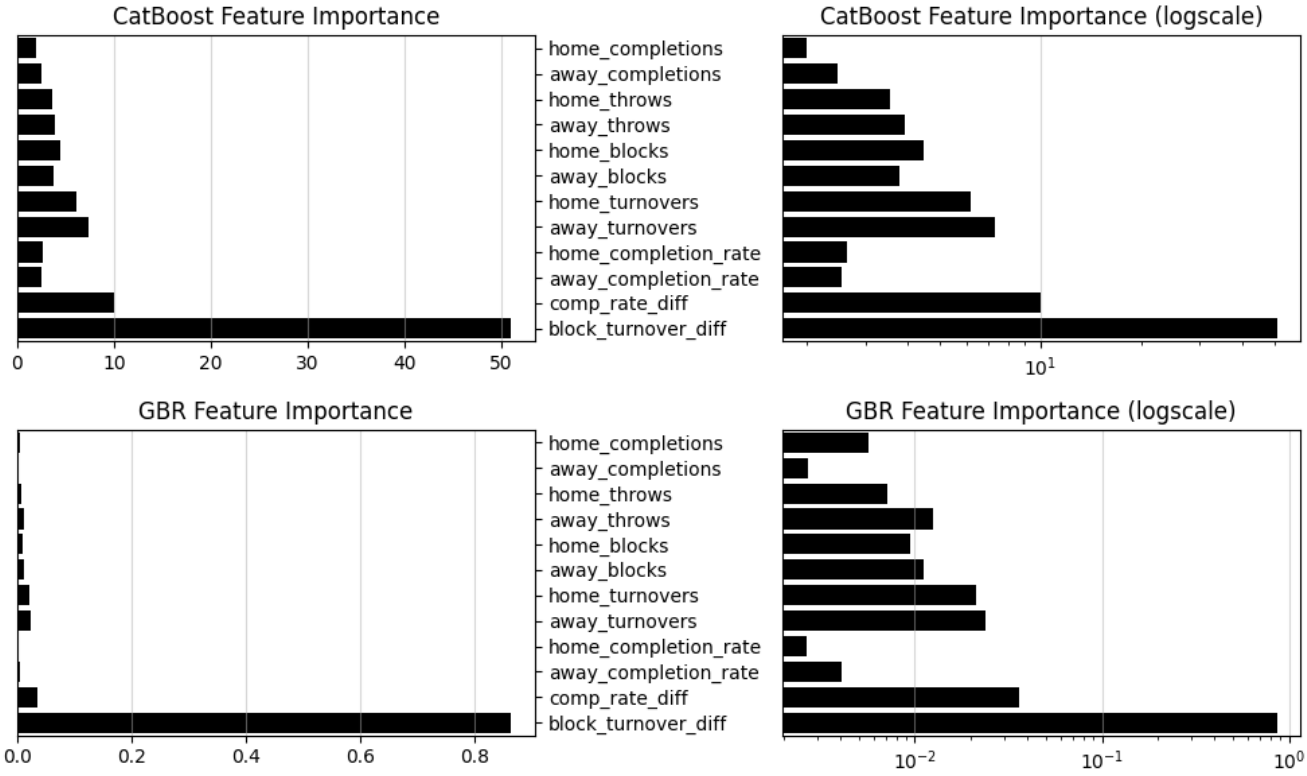


[10] Residual Analysis by Season – Residual vs Season for test data.

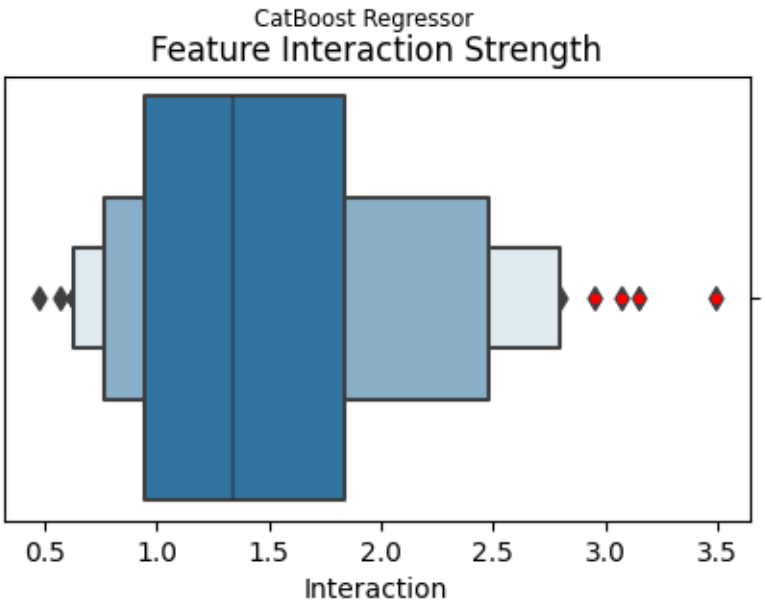


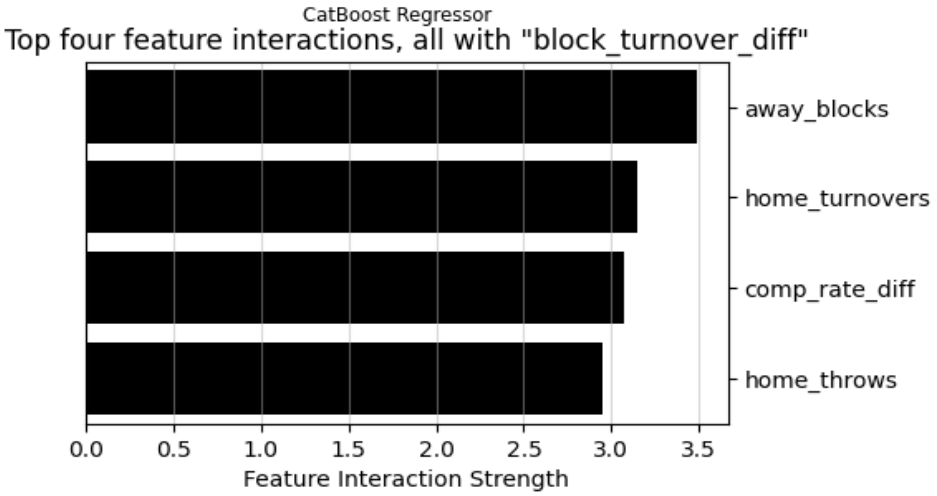
2013-05-04-DC-NY: predicted -8, actual 14, residual 22. Away_turnovers were unlikely to be 0, given home_blocks were 19.
2018-07-14-DET-PIT: predicted 1, actual 22, residual 21. Away stats were probably not recorded, only throws differ (208 vs 209) from home. This difference allowed record to survive cleaning.

[11] Top Models' Feature Importance – Feature importance for the two models comprising the final deliverable.



[12] CatBoost Feature Interactions – Feature interaction strength for the CatBoost Regressor. Top four feature interactions are presented in the bar chart below.





[13] Residual Analysis by Team – Test set error distributions for each team's home and away games.