APRENDIZAGEM COMPUTACIONAL

GRUPO 60

PROJECTING WNBA PLAYOFF TEAMS

SLAM DUNK ANALYTICS

JUAN BELLON - UP201908142
AFONSO PINTO - UP202008014
JOÃO REIS - UP202007227

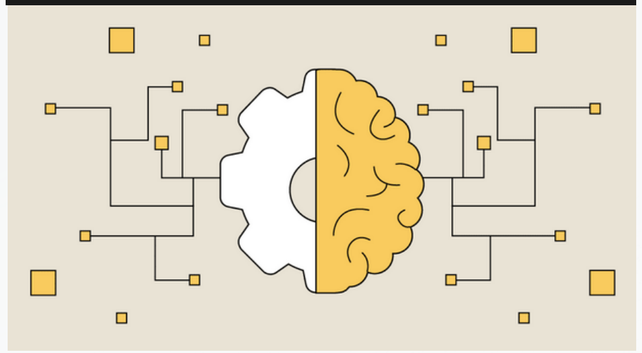
PROBLEM DEFINITION

Objective

- Develop a predictive model to forecast which basketball teams will qualify for the upcoming season's playoffs.
- Focus on historical performance trends, team dynamics, and individual player impacts.

Challenges

- Avoiding overfitting the predictive model to historical data, which may not accurately represent future scenarios.
- Ensure the model adapts to yearly changes in teams, players, and strategies.



DOMAIN DESCRIPTION

Dataset has ten years of statistics

awards_players coaches players
players_teams series_post teams teams_post

Predict playoff qualifiers for upcoming season
Understand key performance indicator influencing playoff qualification

EXPLORATORY DATA ANALYSIS

Number of teams

The number of teams each season is inconsistent, as franchises are created and dissolved

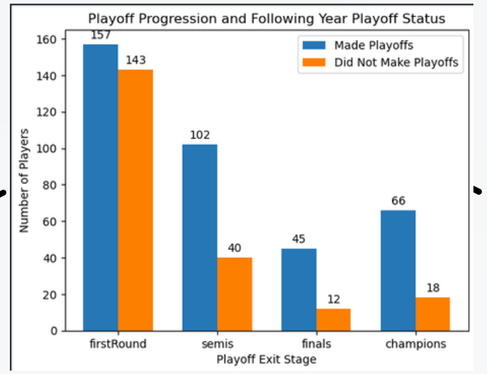
Playoff Experience

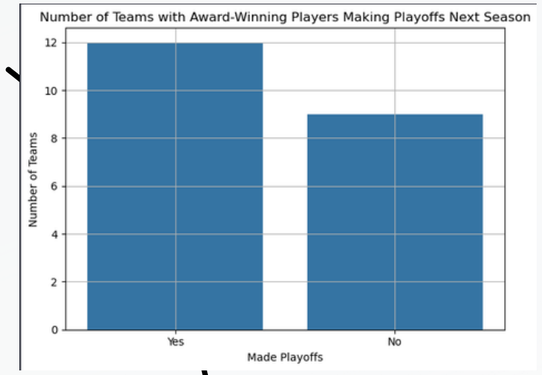
Teams with prior playoff experience are more likely to qualify again.
Higher likelihood for those who advanced further in previous playoffs.

Awards

franchises with awards winning players are more likely to go to playoffs

```
Teams that appear in one year but not in the following one (Years 1 to 10):
Team: Miami Sol, Year They Stop Appearing: 4
Team: Orlando Miracle, Year They Stop Appearing: 4
Team: Portland Fire, Year They Stop Appearing: 4
Team: Utah Starzz, Year They Stop Appearing: 4
Team: Cleveland Rockers, Year They Stop Appearing: 5
Team: Charlotte Sting, Year They Stop Appearing: 8
Team: Houston Comets, Year They Stop Appearing: 10
Teams that start appearing in one year (Years 2 to 10):
Team: Connecticut Sun, Year They Start Appearing: 4
Team: San Antonio Silver Stars, Year They Start Appearing: 4
Team: Chicago Sky, Year They Start Appearing: 7
Team: Atlanta Dream, Year They Start Appearing: 9
```





EXPLORATORY DATA ANALYSIS

3-Point Shooting

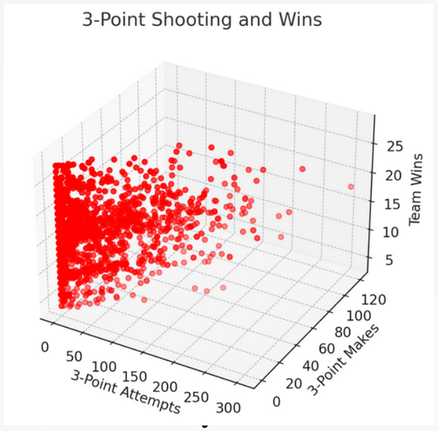
Teams with higher 3-point shooting efficiency (more makes with fewer attempts) tend to win more games.

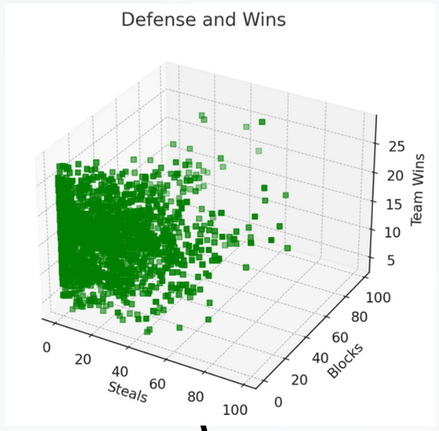
Aggregation

Data was grouped by playerID and year, then aggregated by taking the first value of categorical data and summing numerical values, enhancing its analysis suitability.

Defense

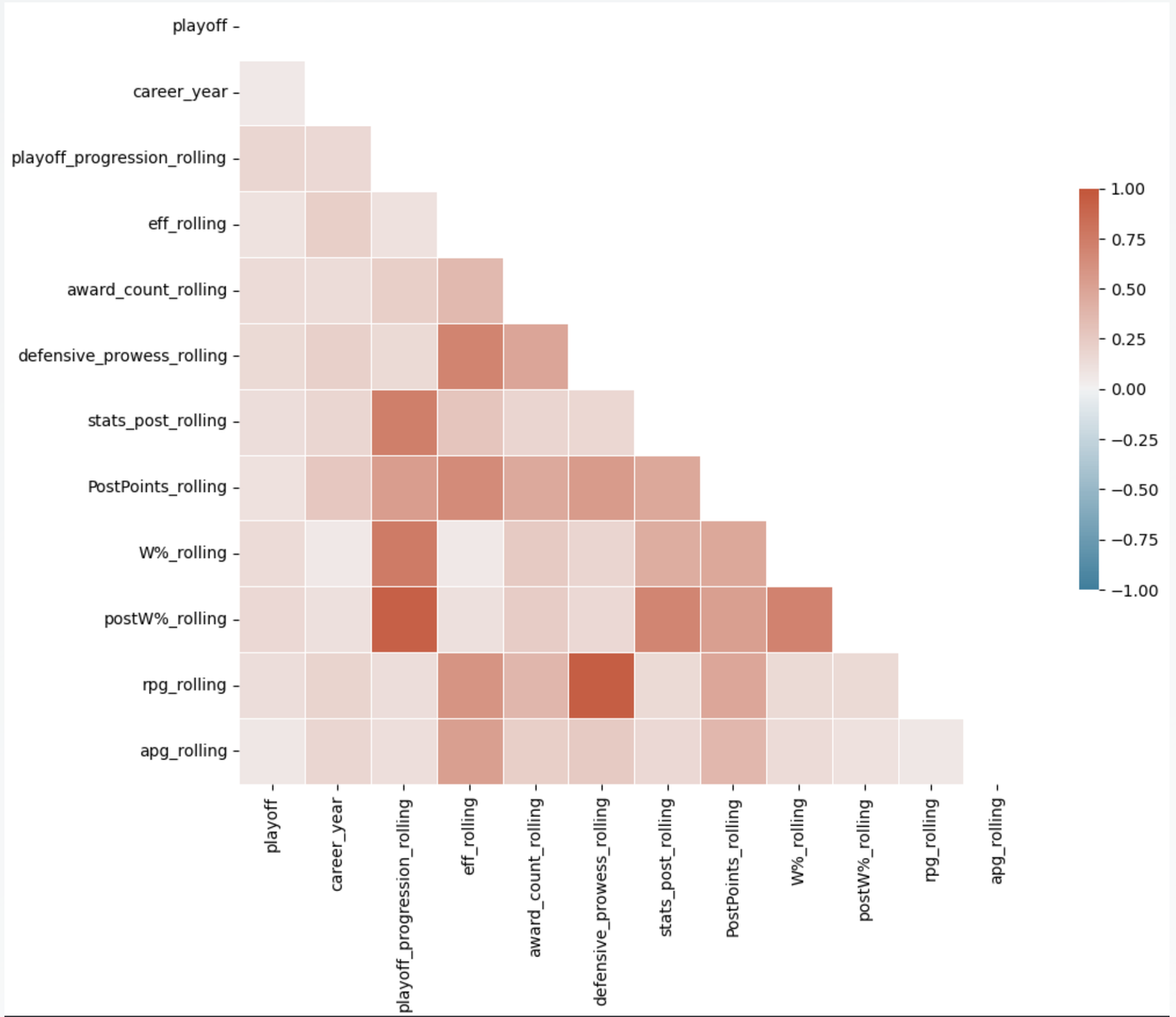
Effective defensive plays, as indicated by steals and blocks, are crucial for team victories





EXPLORATORY DATA ANALYSIS

FEATURE IMPORTANCE



DataSet	Values
players_teams	lgID, ftMade, ftAttempted, fgMade, fgAttempted, threeMade, threeAttempted,GS, GP, PostftMade, PostftAttempted, PostfgMade, PostfgAttempted, PostthreeMade, PostthreeAttempted, PostGS, PostGP, ft%, fg%, three%, gs%, Postft%, Postfg%, Postthree%, Postgs%, career_year
coaches	lgID, won, lost, post_wins, post_lost, total_games, W%, total_p_games, postW%
series_post	lgID
team_post	lgID
players	lgID, college, collegeOther, deathDate
awards_players	lgID
teams	lgID

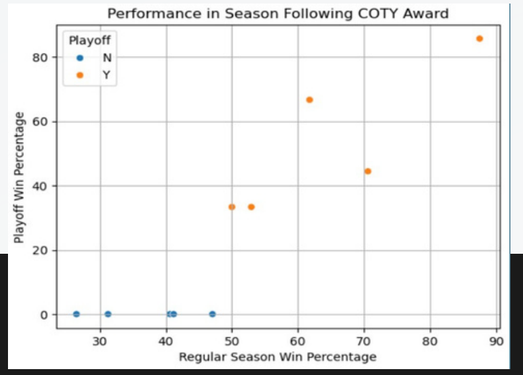
DATA PREPARATION

ft% - Free Throw Percentage
fg% - Field Goal Percentage
three% - Three-Point Percentage
gs% - Games Started Percentage

SHOOTING EFFICIENCY

Devised a reward system that allocates points to teams reflecting their achievements in last season's playoffs.

TEAM PAST SUCCESS



Created a new column that stores the count of awards won by players in a year, but not coaches as only half of award winning coaches made it to the playoffs in the following year.

AWARD COUNT

DATA PREPARATION

Postft% - Postseason Free Throw Percentage
Postfg% - Postseason Field Goal Percentage
Postthree% - Postseason Three-Point Percentage
Postgs% - Postseason Games Started Percentage

PLAYOFF PERFORMANCE

efg% - Effective Field Goal Percentage
ts% - True Shooting Percentage
drb% - Defensive Rebounds Percentage

ADVANCED METRICS

ppg - Points Per Game
rpg - Rebounds Per Game
apg - Assists Per Game
spg - Steals Per Game

GAME STATISTICS

DATA PREPARATION

We found missing values on the columns **college**, **pos** and **collegeOther**.
Found the best wayto deal with this was create a new category for those without college, and dropped players who had no position, as we found out that data was correlated with results.

MISSING VALUES

Found players who had impossible **height** and/or **weight** values, and ended up dropping those rows.

OUTLIERS

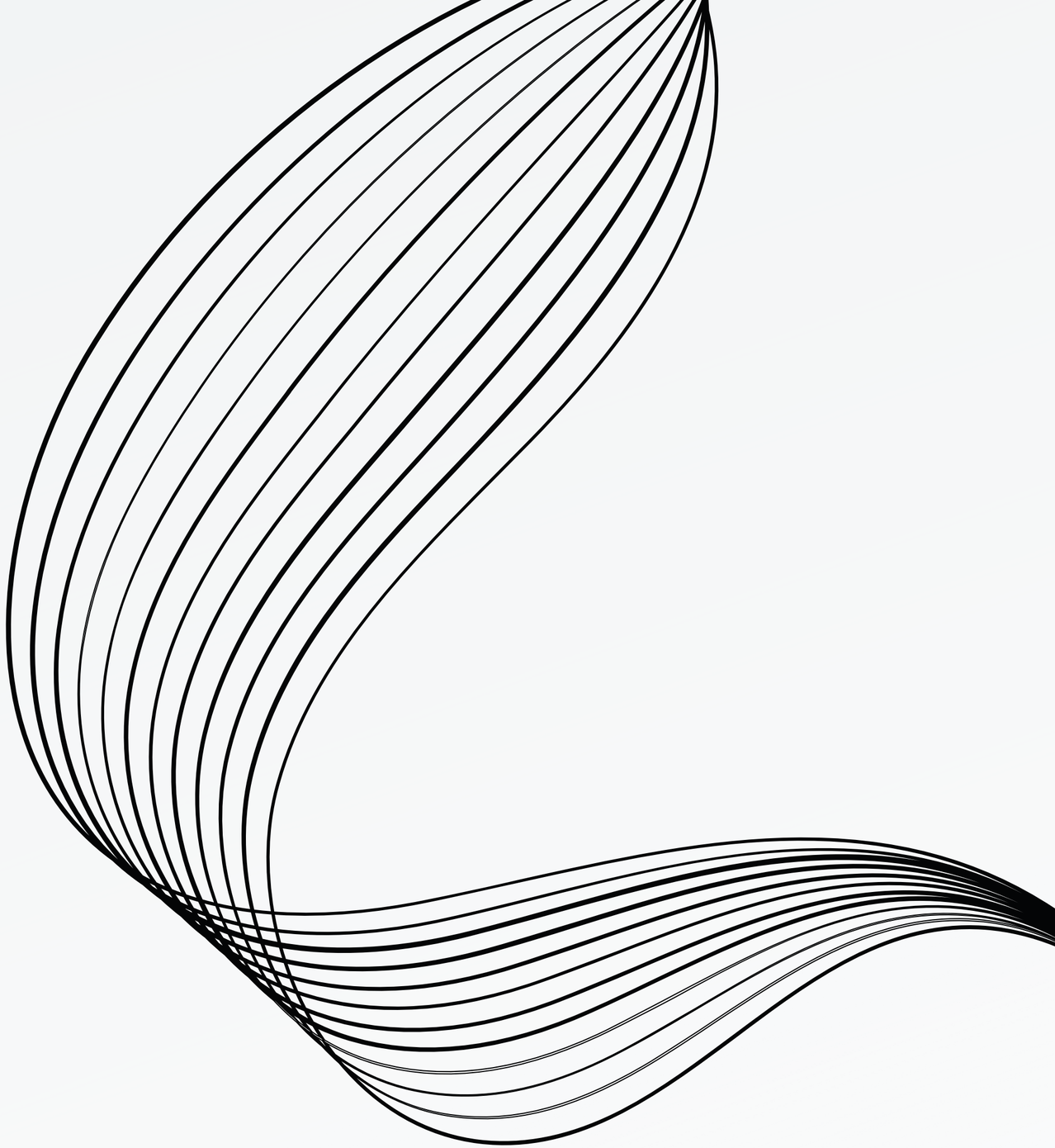
Mapped string values to int in order to be compatible with models used for the following columns:

playerID
teamID
bioID
college
confID
playoff

DATA CONVERSION

IMBALANCED DATA

We used Tomek Links in conjunction with SMOTE to better balance our data upon splitting (the first cleans overlapping points, while the second balances the dataset by increasing the number of samples in the minority).
Used filter method to select column combinations that give the best results



EXPERIMENTAL SETUP

We adjusted the model to base its predictions on the previous year's statistics by modifying the training and testing data accordingly.

RandomForest
KNN
LGBM
Bagging
Logistic Regression

ALGORITHMS

Accuracy

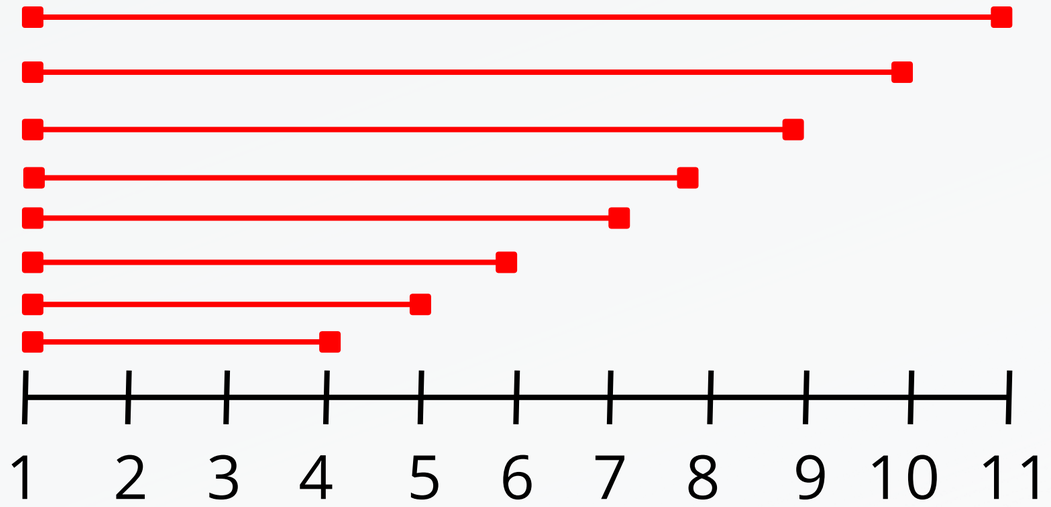
EVALUATION

Grid Search
Randomized Search

HYPERPARAMETER TUNING

ROLLING AVERAGES METHOD

Utilizing a Rolling Averages approach, we maximize past data in our analysis to produce more accurate forecasts. In order to concentrate on historical performance, this method computes averages for important player measures over a three-year period, excluding the current year. Since the first three years of data don't provide us a complete three-year history, we don't use them. By eliminating unnecessary columns from the data and setting the current year's data in our test set to zero, we can clean up the data when we use this method. The dataset expands with each passing year while being relevant for the intended year. In this sense, our model gains insight from historical patterns, enhancing its predictive power for subsequent iterations.



RANDOM FOREST

Builds several decision trees using various dataset subsets, averaging their predictions for regression or classifying data by majority vote, hence improving accuracy and decreasing overfitting.

KNN

Forecasts a data point's label by examining the 'k' nearest labeled data points, applying an average for regression or a majority vote for classification. **White box** as it doesn’t rely on hidden layers or complex mathematical transformations

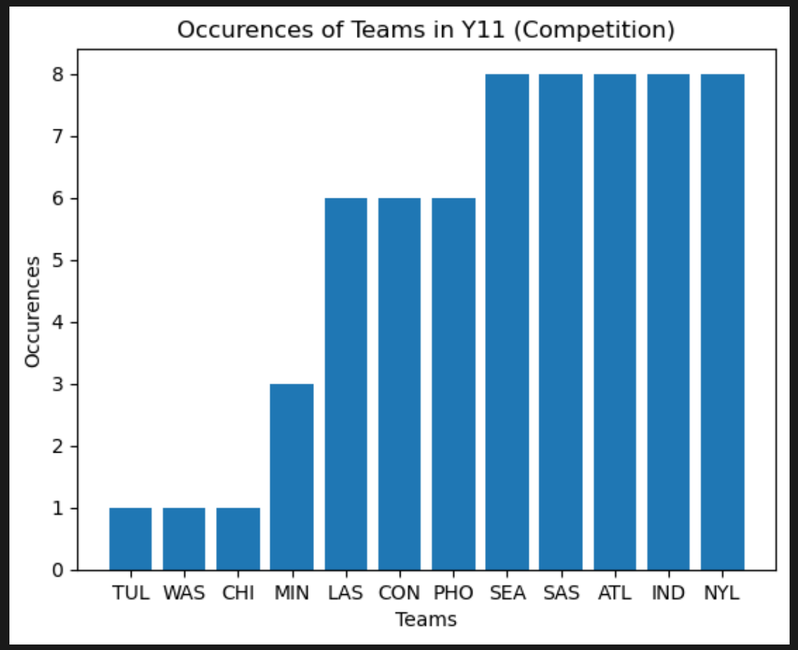
LGBM

LightGBM grows trees leaf-wise rather than level-wise, which can lead to better accuracy with lower data sizes and less computing power. It's highly effective for large datasets and can handle categorical features intrinsically. However, due to its complexity and the opaque nature of how it iteratively improves its predictions, LightGBM is generally considered a **Black box** model, meaning humans do not easily interpret its internal decision-making process.

LOGISTIC

It predicts the probability that a given input belongs to a particular category, often visualized as an S-shaped curve (sigmoid function). The model outputs coefficients for each feature, providing insight into the relative impact of each feature on the prediction, making it a **White box** model.

AGGREGATION METHOD



example of Year 11

For each year, it compiles all the predicted teams for both conferences from various models. The method then identifies the four most frequently expected teams in each conference by counting occurrences and selecting the top four. Once these top teams are identified, they are combined to form a list of the top eight teams across both conferences. The actual units that made it into the playoffs for that year are then used to evaluate the accuracy of these predictions. The accuracy is calculated as the proportion of correctly predicted teams out of these top eight. This approach allows for a conference-specific analysis and provides a comprehensive view of the overall playoff landscape for the year.

EVALUATION

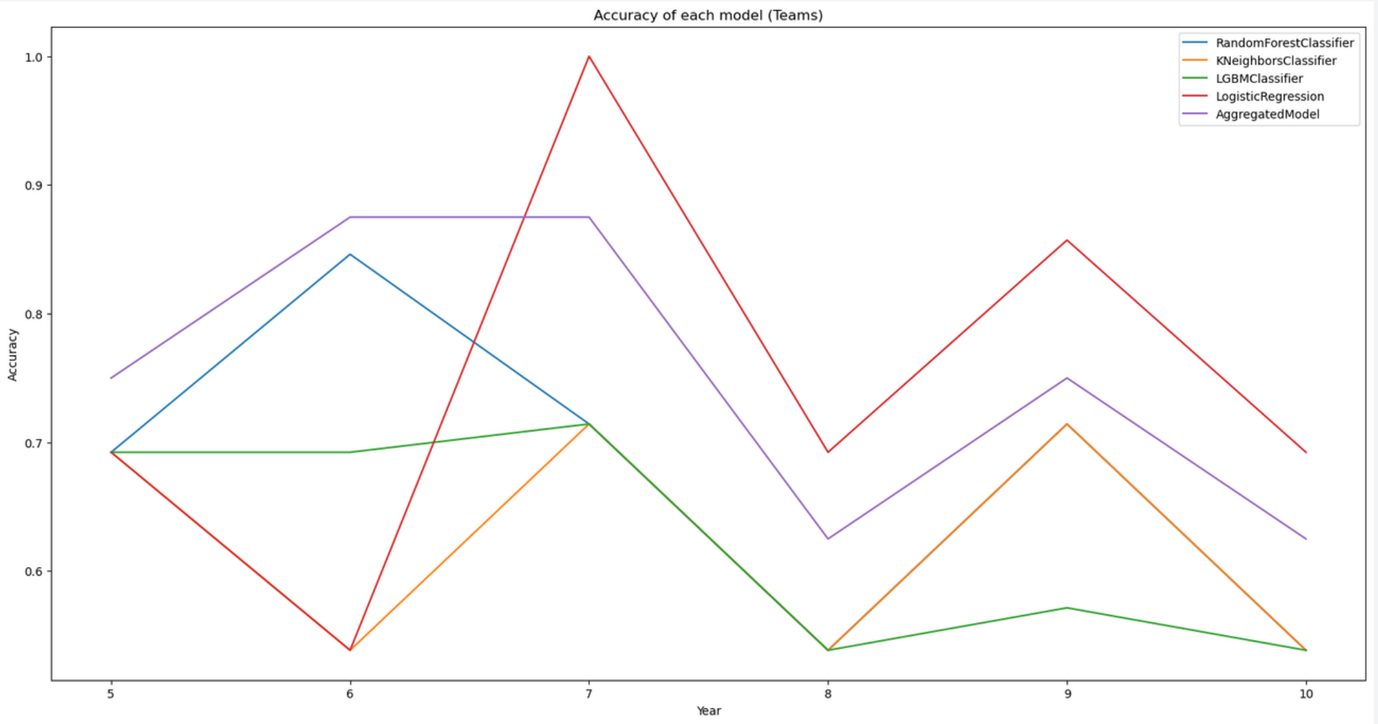
01 **ACCURACY** – GENERAL MEASURE

02 **PRECISION** – MINIMIZE FALSE POSITIVES

03 **RECALL** – MINIMIZE FALSE NEAGATIVES

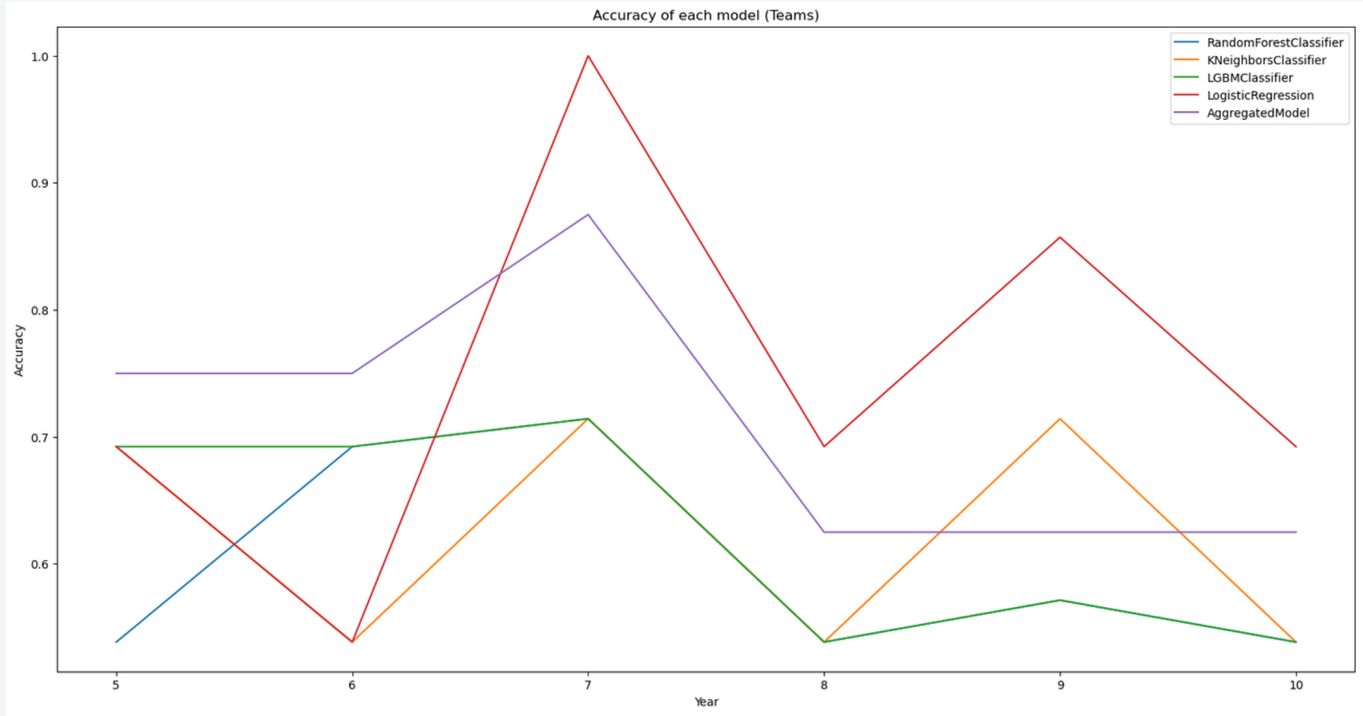
04 **FOWLKES-MALLOWS** – BALANCED VIEW OF 02 AND 03

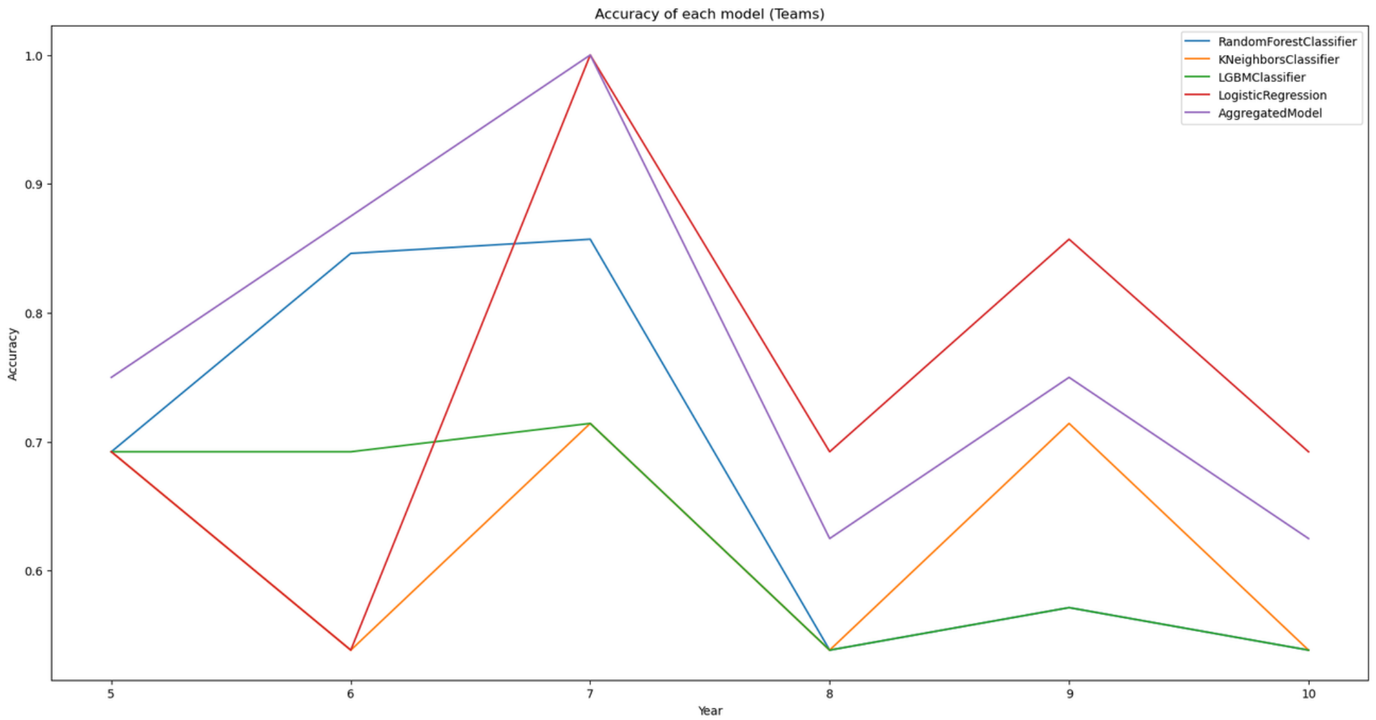
05 **ROC-AUC** – APPROPRIATE FOR BINARY CLASSIFICATION PROBLEMS, INSENSITIVE TO IMBALANCE



Precision

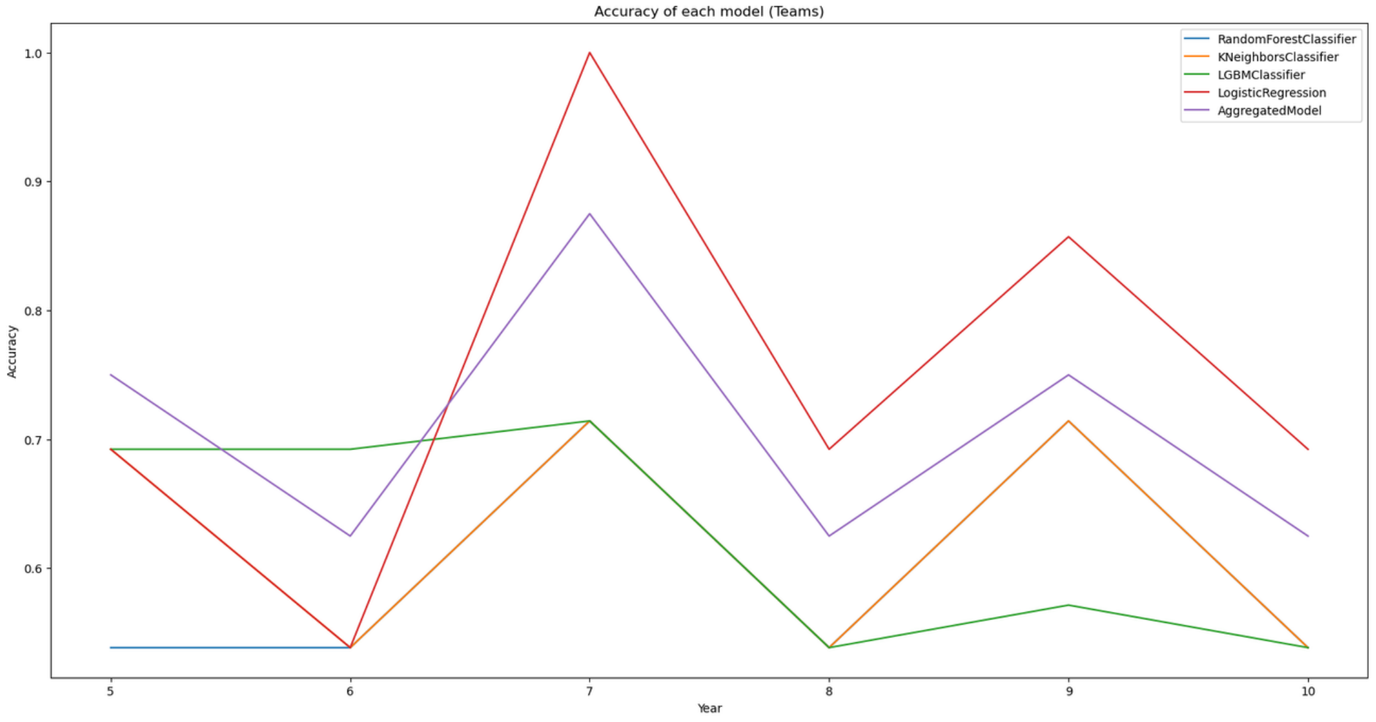
Fowlkes-Mallows

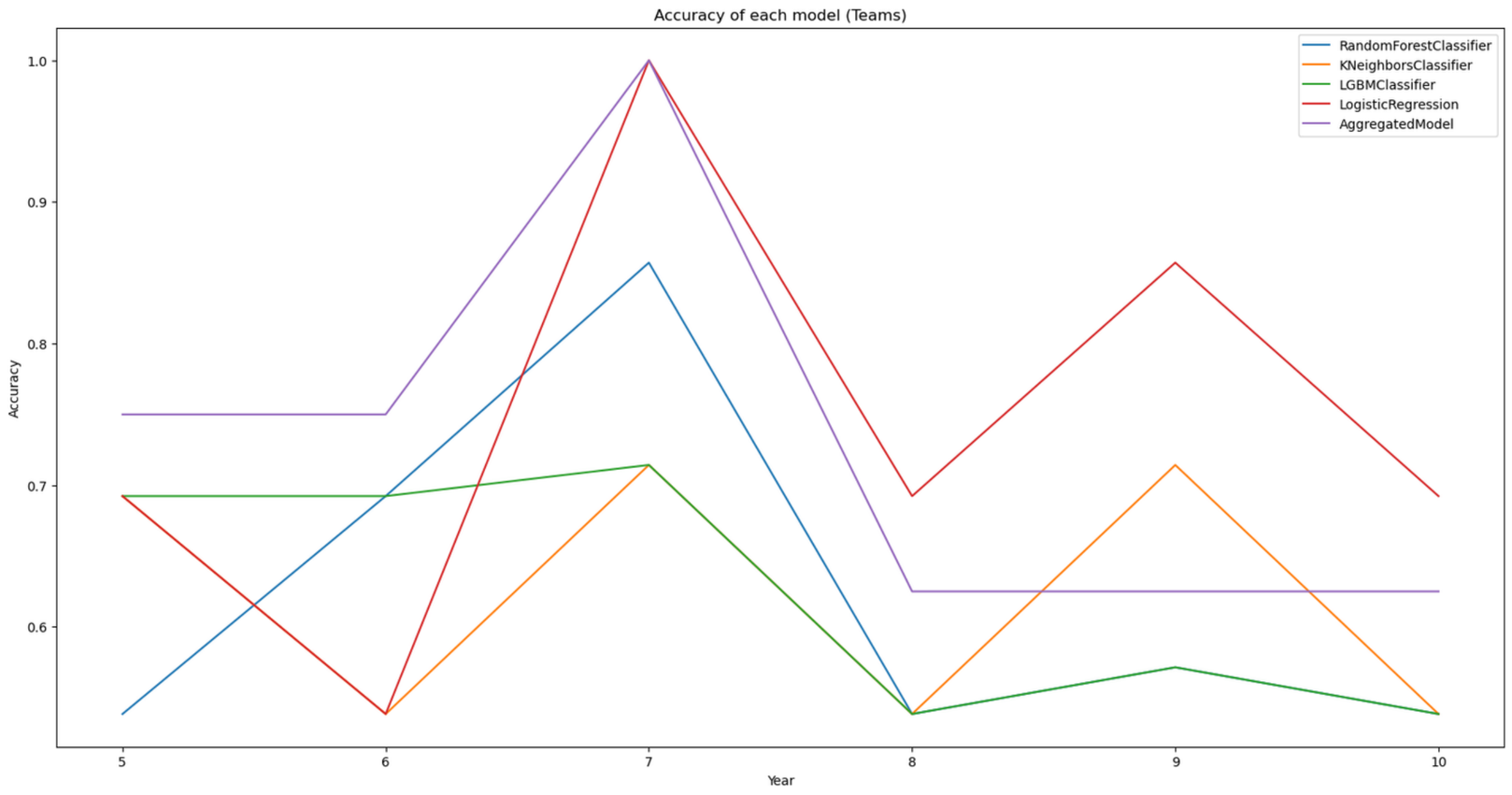




ROC-AUC

Accuracy




Accuracy of each model (Teams)
RandomForestClassifier
KNeighborsClassifier
LGBMClassifier
LogisticRegression
AggregatedModel
1.0
0.9
0.8
0.7
0.6
Accuracy
5
6
7
8
9
10
Year

Recall

PARAMETER TUNING

GRID SEARCH

We used **GridSearchCV** for Hyperparameter Tuning.
This library assesses the model performance for every combination of hyperparameters by methodically going over a specified hyperparameter grid.

RANDOM GRID SEARCH

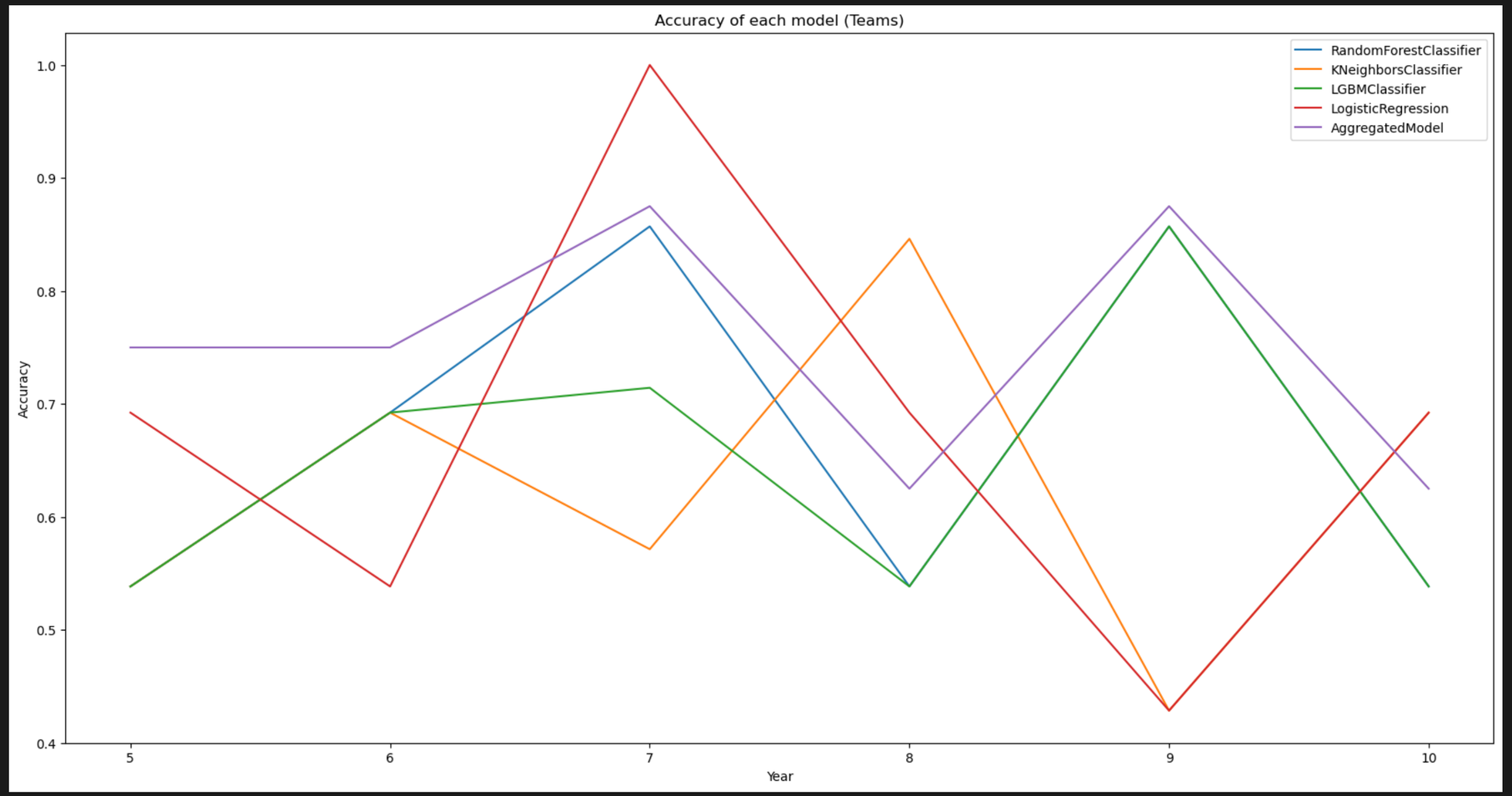
We employed **RandomizedSearchCV** for Hyperparameter Tuning. This library evaluates model performance across a range of hyperparameter combinations by systematically exploring a specified hyperparameter grid, but unlike GridSearchCV, it selects combinations randomly. This approach allows for a more efficient search over a large hyperparameter space, as it can sample a subset of the grid, reducing the computational burden.

COMPARISON

As expected, GridSearch tends to yield better results. This is primarily because GridSearch exhaustively searches through the entire specified hyperparameter grid, ensuring that it evaluates every possible combination. However, this thoroughness comes with a significant cost in terms of computational time and resources.

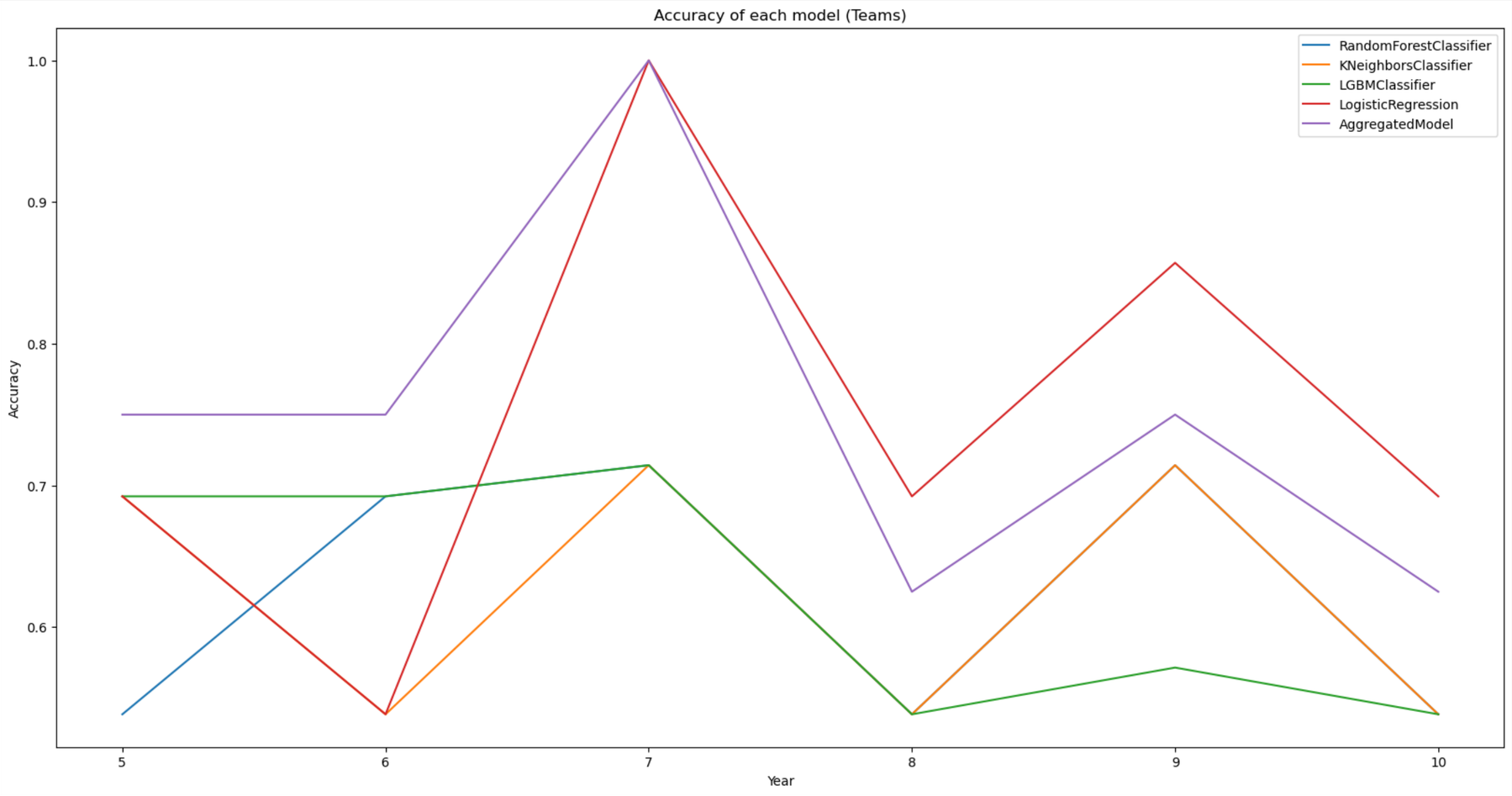
RESULTS

No Oversample
No Parameter Tuning



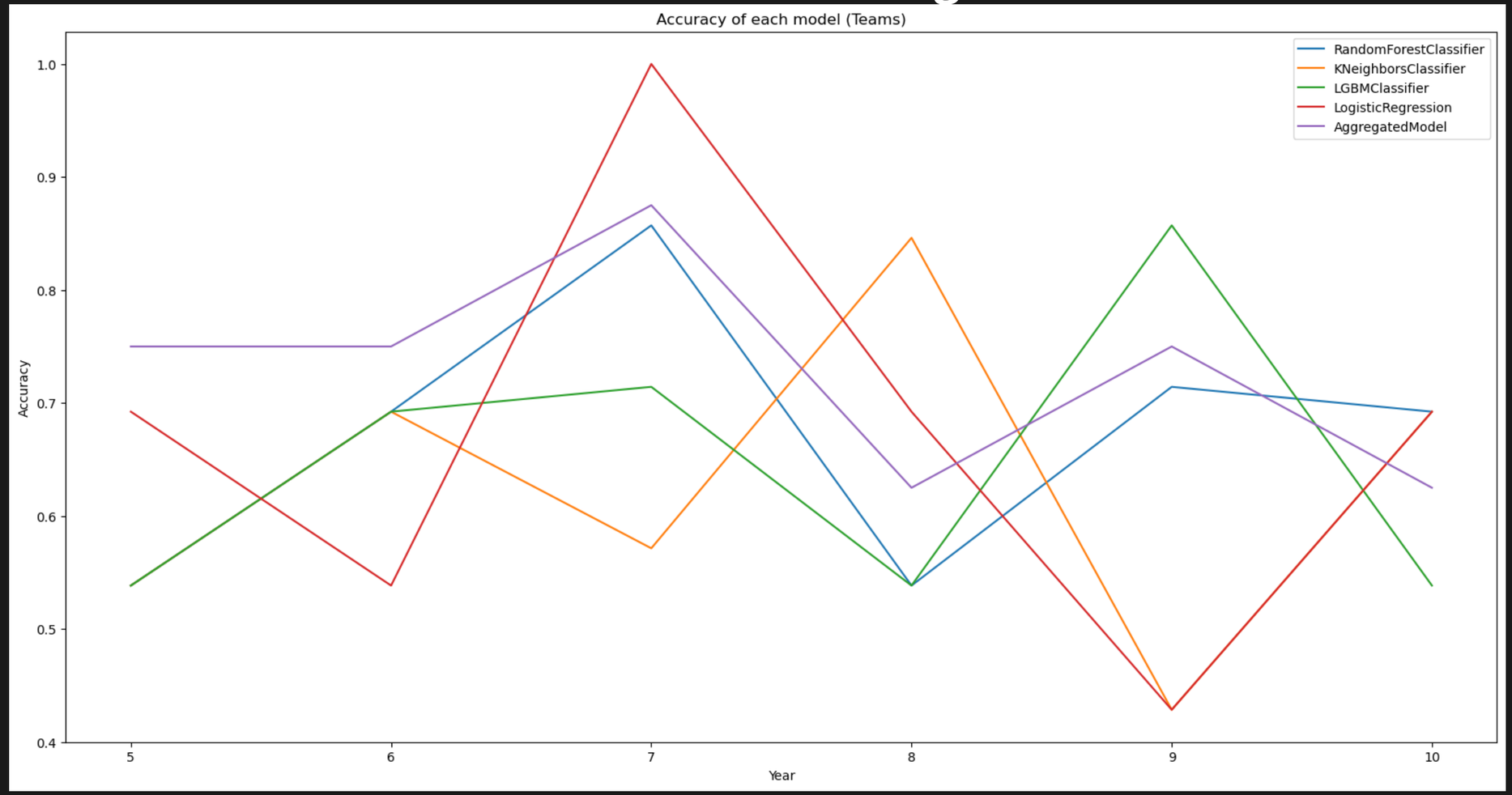
RESULTS

Oversample
No Parameter Tuning



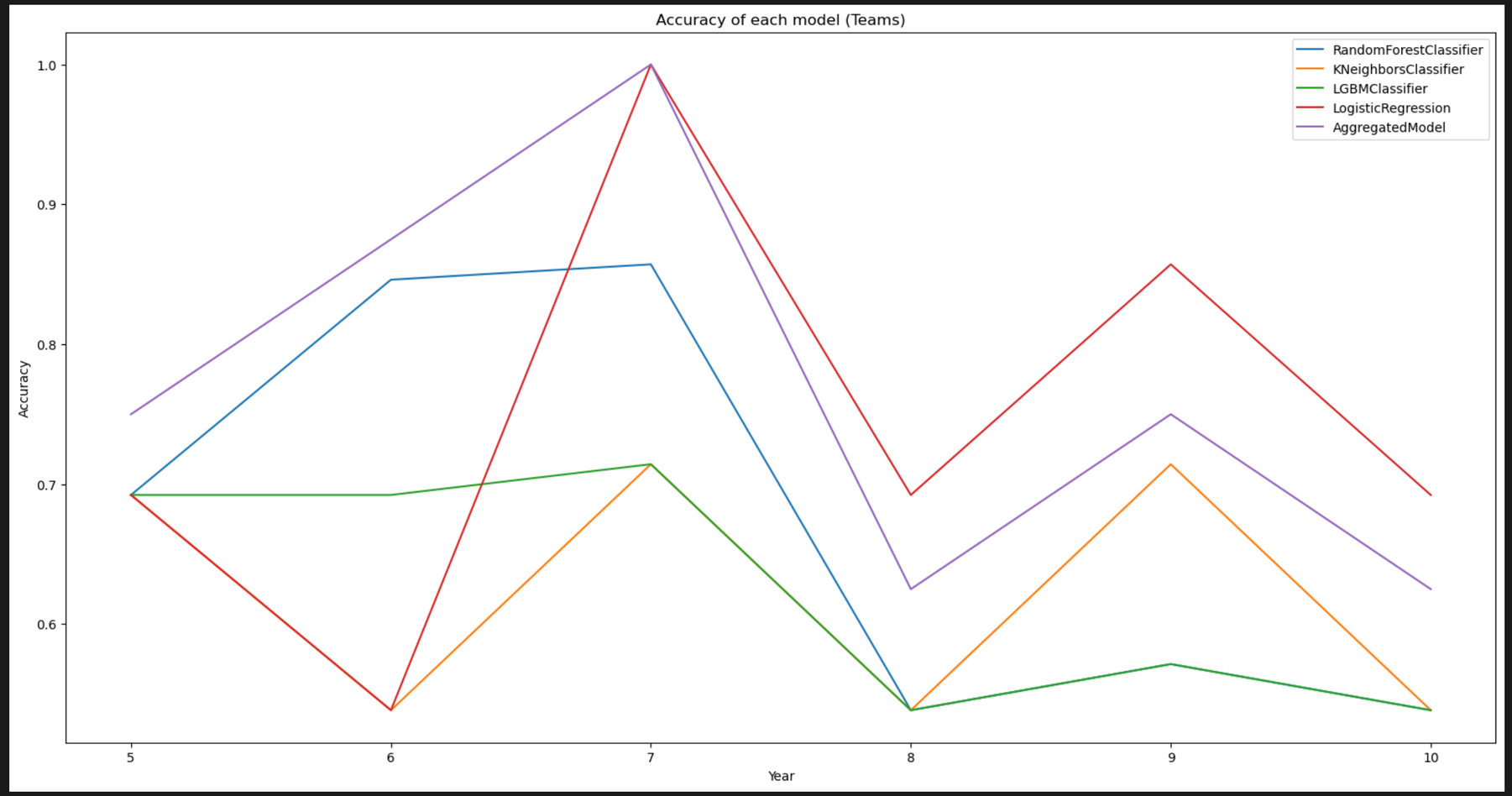
RESULTS

No Oversample
Parameter Tuning



RESULTS

Oversample
Parameter Tuning



CONCLUSIONS

- By incorporating additional features, we could continuously assess and experiment with various algorithms, significantly enhancing our grasp of the issue and the outcomes.

- Utilizing previously untapped tables for feature engineering yielded the most substantial advancements.

FUTURE WORK

- Deepen knowledge of the mechanics behind algorithms and methodologies, such as the nuances of hyperparameter optimization.
- Invest heavily in feature engineering, in order to keep improving results.