Introduction

The objective of this project is to use Machine Learning techniques to predict the outcome of Future NBA games with historical data.

Problem and Motivation

Sports betting books are generally unreliable and do a poor job at estimating the results of sports games. The susceptibility of the books to high variance has led to scandals in recent years, as well as a decrease in user enjoyment due to inconsistencies across books on different platforms.

By implementing an algorithm that more accurately predicts the results of games, the goal would be to standardize books across all platforms. Assuming the algorithm works as intended, it could be sold to platforms who would implement it in place of their current-practice prediction models.

Literature Review

Predicting the winner can be difficult because of the complexity of interactions between 10 people on the court [2]. Previous studies have used logistic and linear regression to predict the outcome of games [1]. SVM and Random Forest models can be used to predict the outcome as demonstrated by [3].

Dataset Description

The dataset(s) utilized by our project involves player statistics collected over a long period of the NBA. Attributes of each data-point include, but are not limited to: Points, rebounds, assists, opponent, and date.

Link: https://www.kaggle.com/datasets/wyattowalsh/basketball

Methods

Preprocessing Methods

- Dimension reduction was done using Principal component analysis (PCA) as our dataset has many features.
- Cleaning the dataset:
    - Data points with missing features or bad values were deleted.

  - Standardization done subtracting the sample mean and dividing by the sample standard deviation.
  - We used **DBScan** to detect outliers.
- 70-20-10 split for training, validation, and testing respectively, accounting for hyperparameters in potential models.

ML Models

- **Logistic Regression:** is a supervised machine learning model that can be used for binary classification making it ideal for predicting win or lose outcomes [5].
- **Random Forest:** is a supervised machine learning model where we can create multiple decision trees to improve prediction accuracy and in case of game variability factors [1].
- **Support Vector Machines:** is a supervised machine learning model that uses an optimal hyperplane to separate winning and losing outcomes [2]. We can use this model to handle linear and non-linear data.

Results and Discussion

Quantitative metrics are F1 score, accuracy, precision, and recall because binary classification is used.

- Accuracy gives insight into the model's prediction performance.
- Precision marks the proportion of true positive predictions.
- Recall measures the proportion of actual positive instances to those identified.
- F1 score balances our precision and recall, giving the most insight into where model accuracy lies and allows model adjustment.

We expect the model to correctly identify wins/losses with an accuracy of at least a baseline of 50% and a stretch goal of outperforming other models (accuracy/prediction upwards of 70%). We plan to have a sustainable model trained once and refined every season via easy to change variables. Ethically, we hope redefining prediction algorithms in the sports betting industry allows consumers better insight into potential bets.

Data Exploration and Preprocessing

Initially, we standardized our dataset to ensure that each feature contributed an equal amount to the PCA transformation.

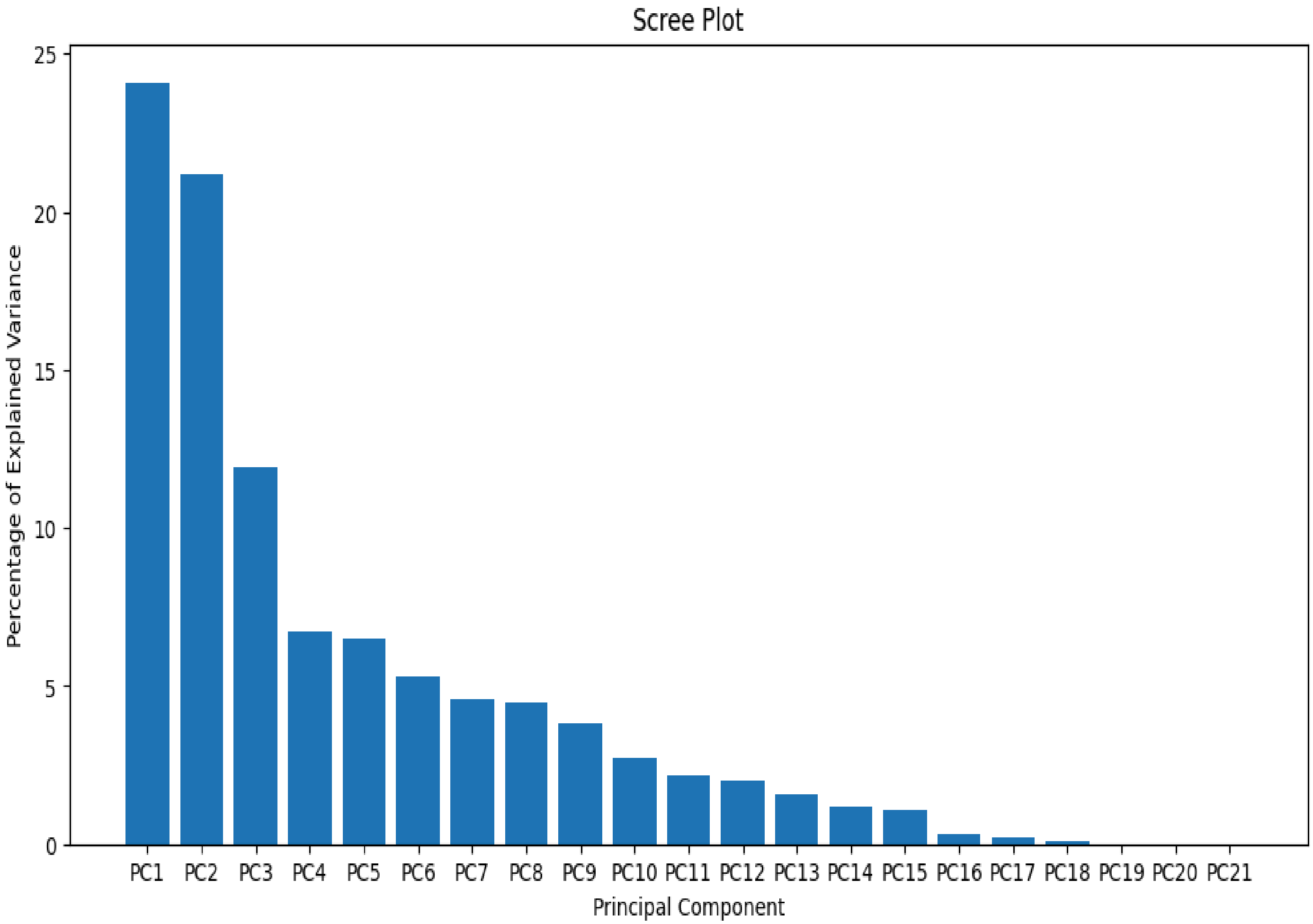


Figure Name: Depicting variation accounted for by found principal components

Observing the scree plot gives us insight towards the principal components that contribute the most to variance in general. It's important to note that this is general variance, not variance directly pertaining towards the win or loss of the home team.

The printed correlations are tied directly to the win or loss of the home team for each principal component. As we can see, the most explanatory principal component is PC1, which is negatively correlated with the win or loss of the home team. Meaning, if there is a high value for PC1 in a given game, that game lends itself to being more likely a loss for the home team. Other components such as PC2 and PC6 are among the higher correlated principal components with wins and losses, but their correlation constant is not nearly as dominant as PC1's.

The dataset is collected after games and since our model is looking to predict the outcome of games, these statistics do not necessarily indicate how features will play out in a given game. There is some merit in observing the cumulative of these values leading up to each game. If a team is consistently putting up high contributions towards the PC1 feature, for example, we hypothesize they are less likely to win the next game.

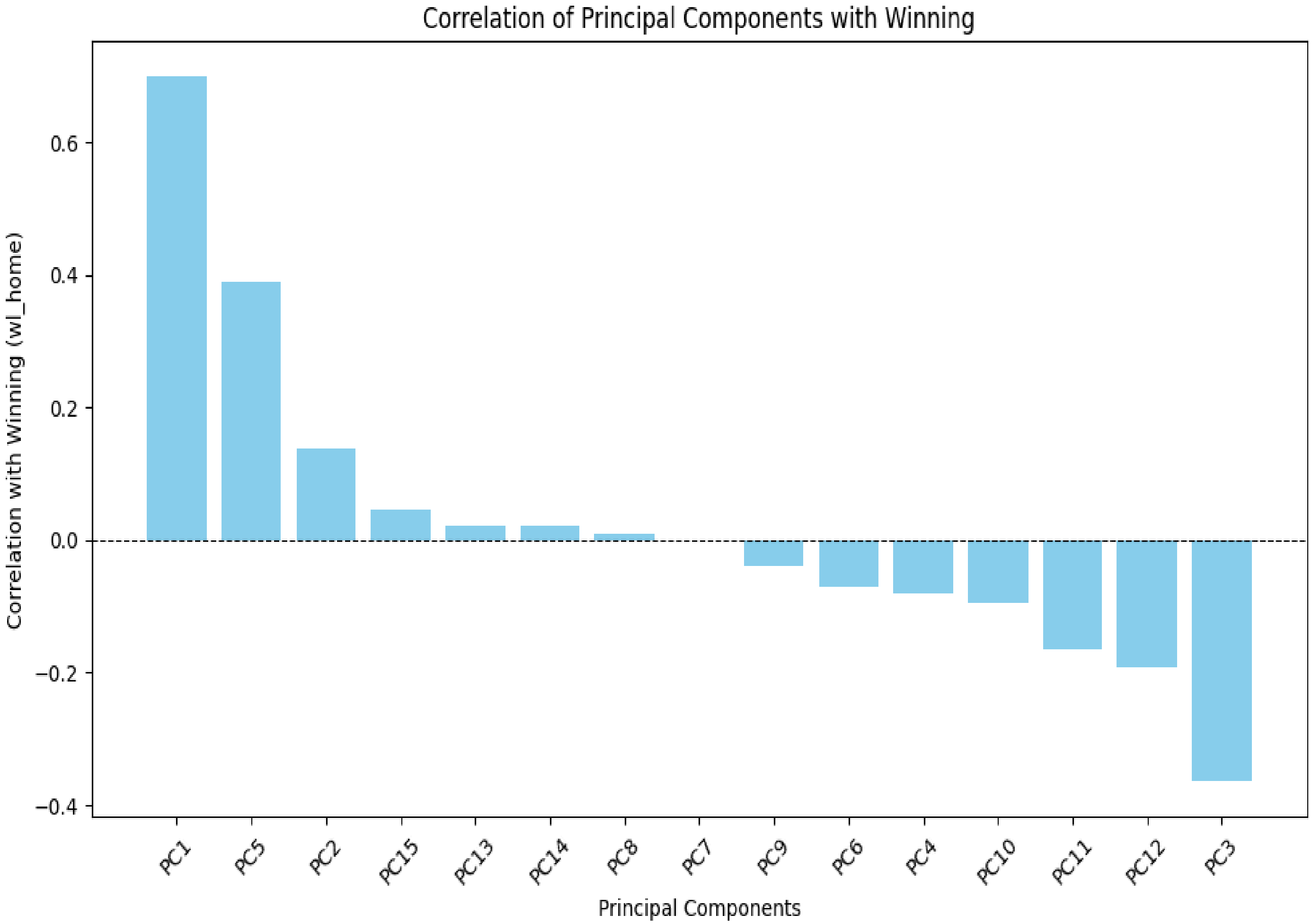


Figure Name: Comparison of principal components by correlation with winning

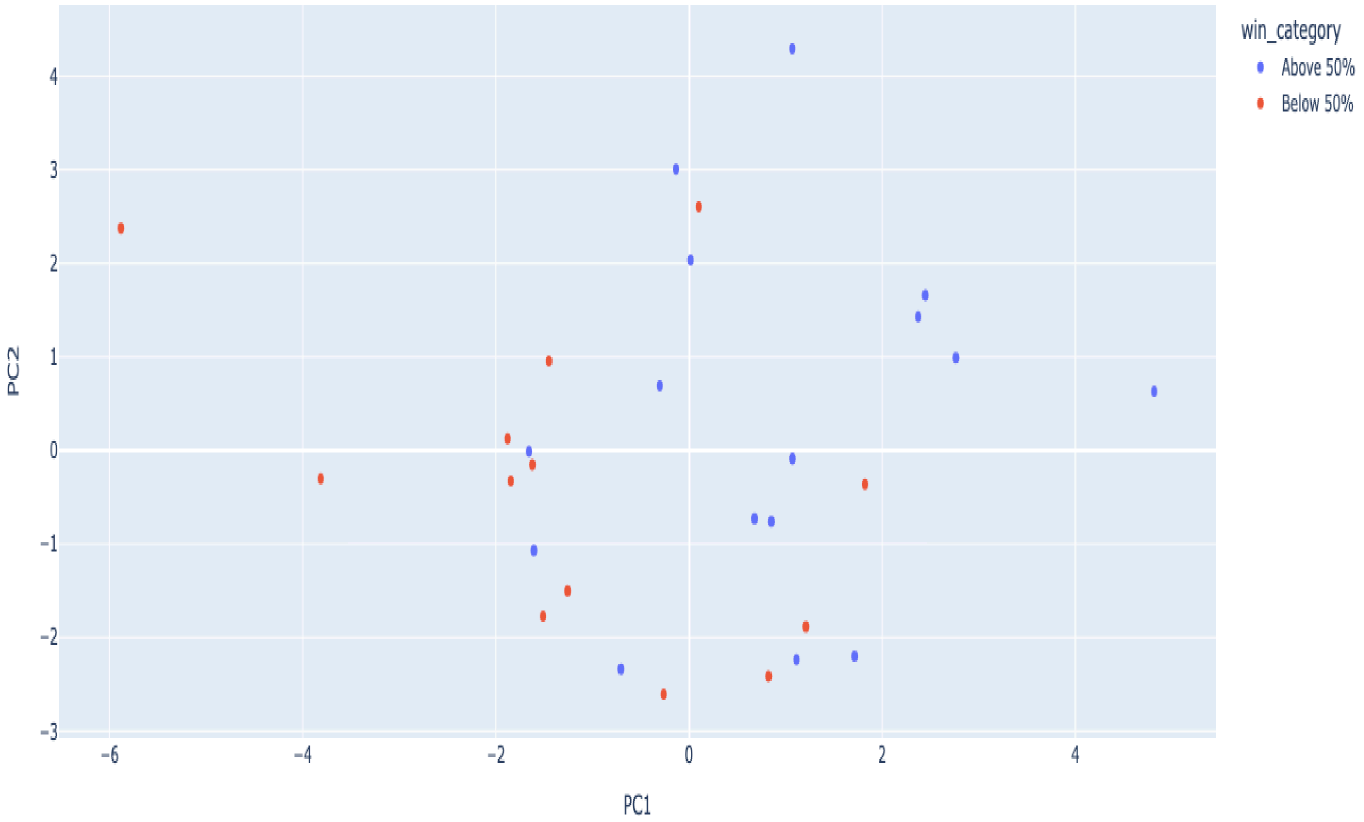


Figure Name: Using PC1 and PC2 to visualize their relationship with winning

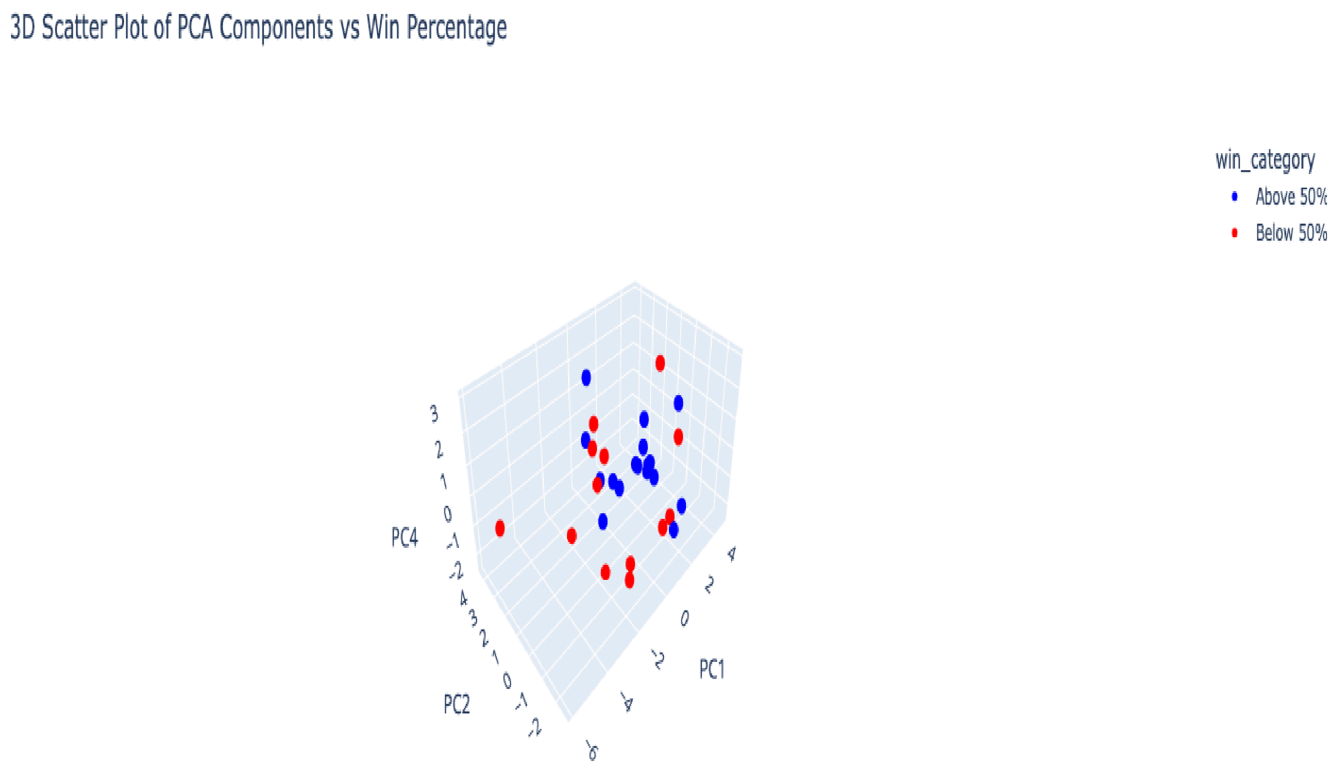


Figure Name: Using PC1, PC2, PC4 to visualize their relationship with winning

This portion of data reduction is much more relevant towards the scope of our project. Focusing on the mean of the features by team from their last 10 games provides a good understanding of how the team is performing in general at the time of their upcoming games, as well as the most important features that are contributing towards their wins or losses.

We plan to predict the outcomes of future games using principal components that are most directly correlated (positive or negative) and analyze how they reflect a team leading up to their future games. Higher values of PC1 and PC2 show the percentage of winning goes down. The opposite happened for PC4. While this portion of our project is not used to draw definitive conclusions or predictions, it can establish the foundation for our machine learning models for predicting the outcome of games.

Logistic Regression

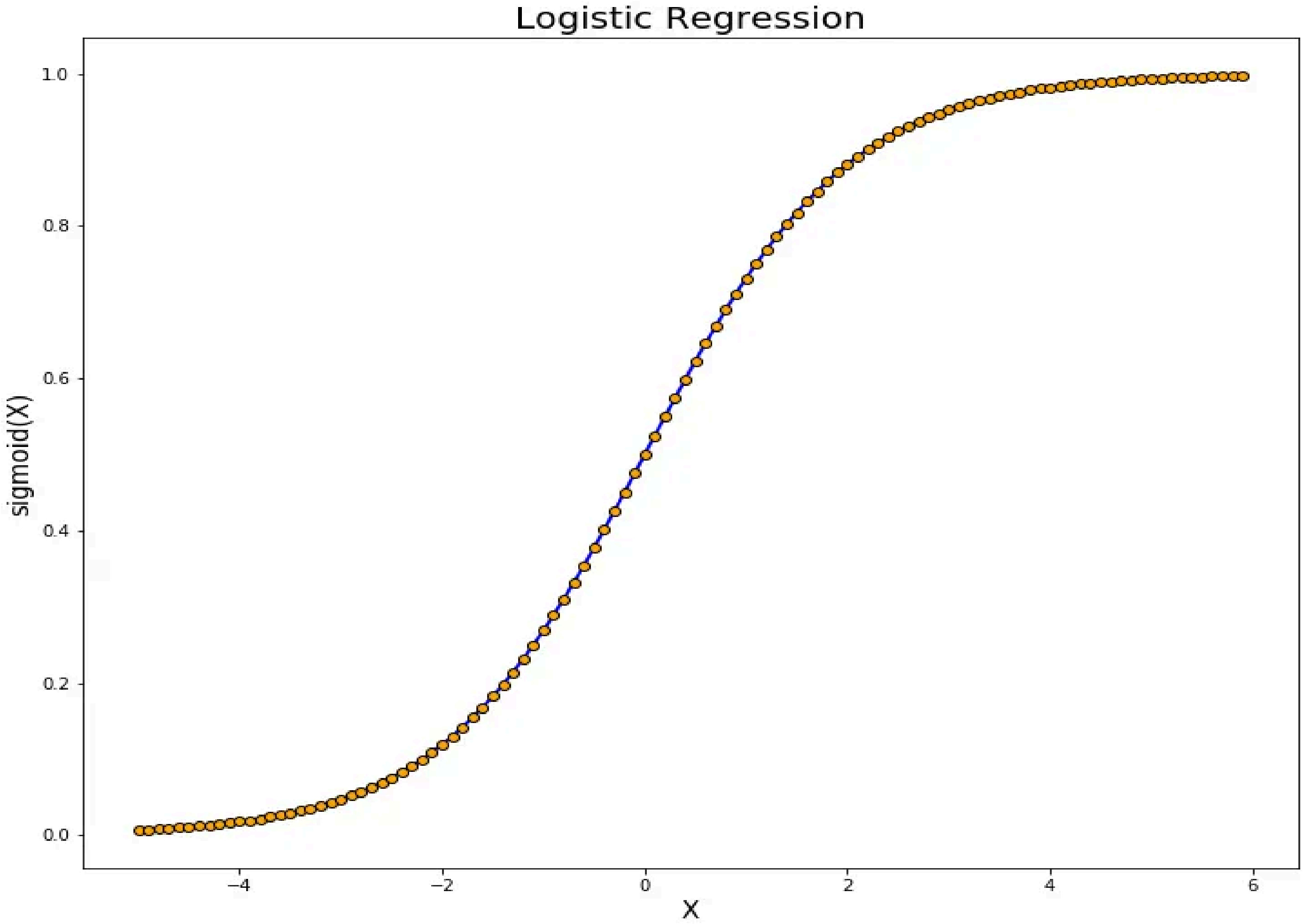
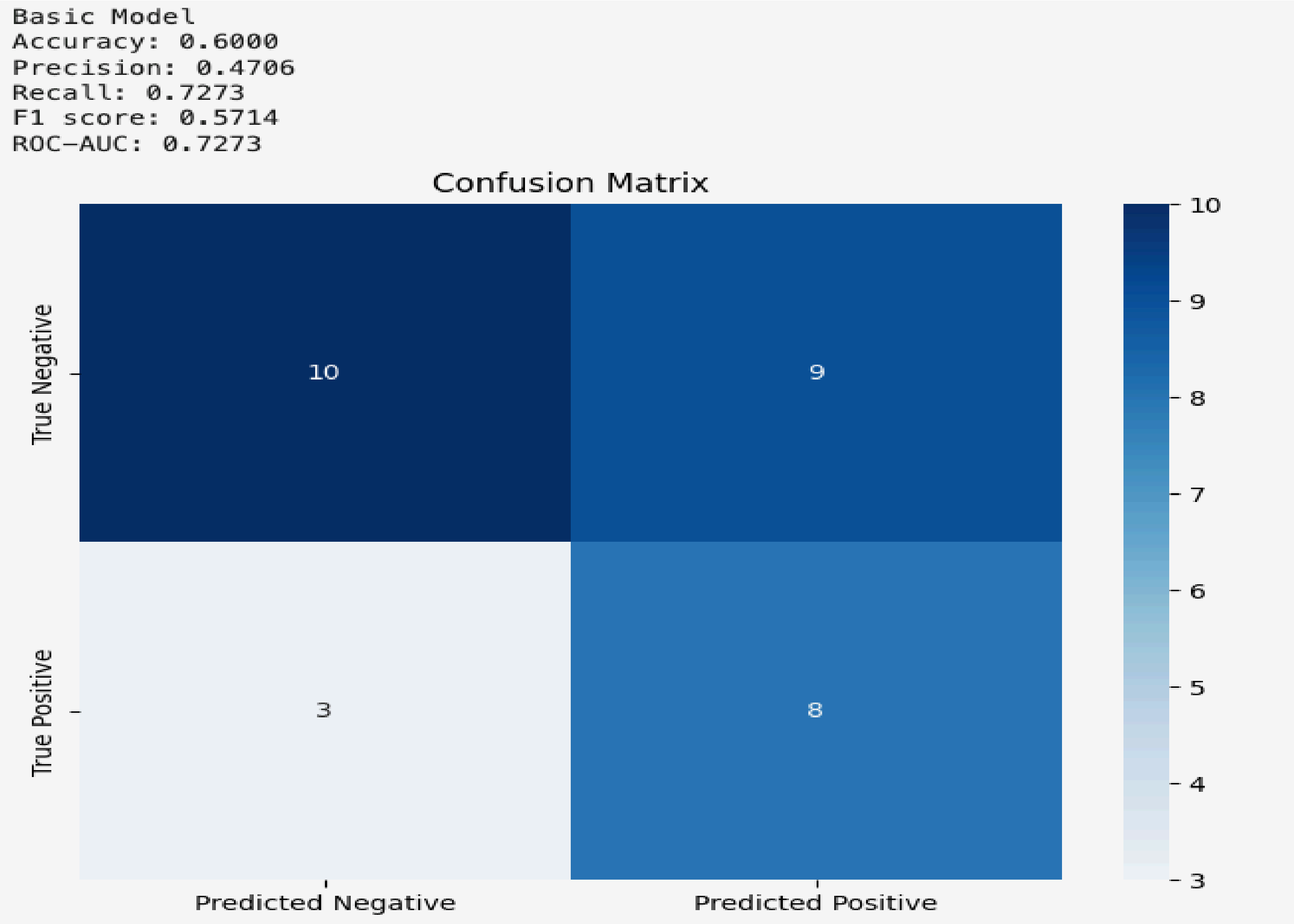


Figure Name: Logistic Regression Visualization [10]

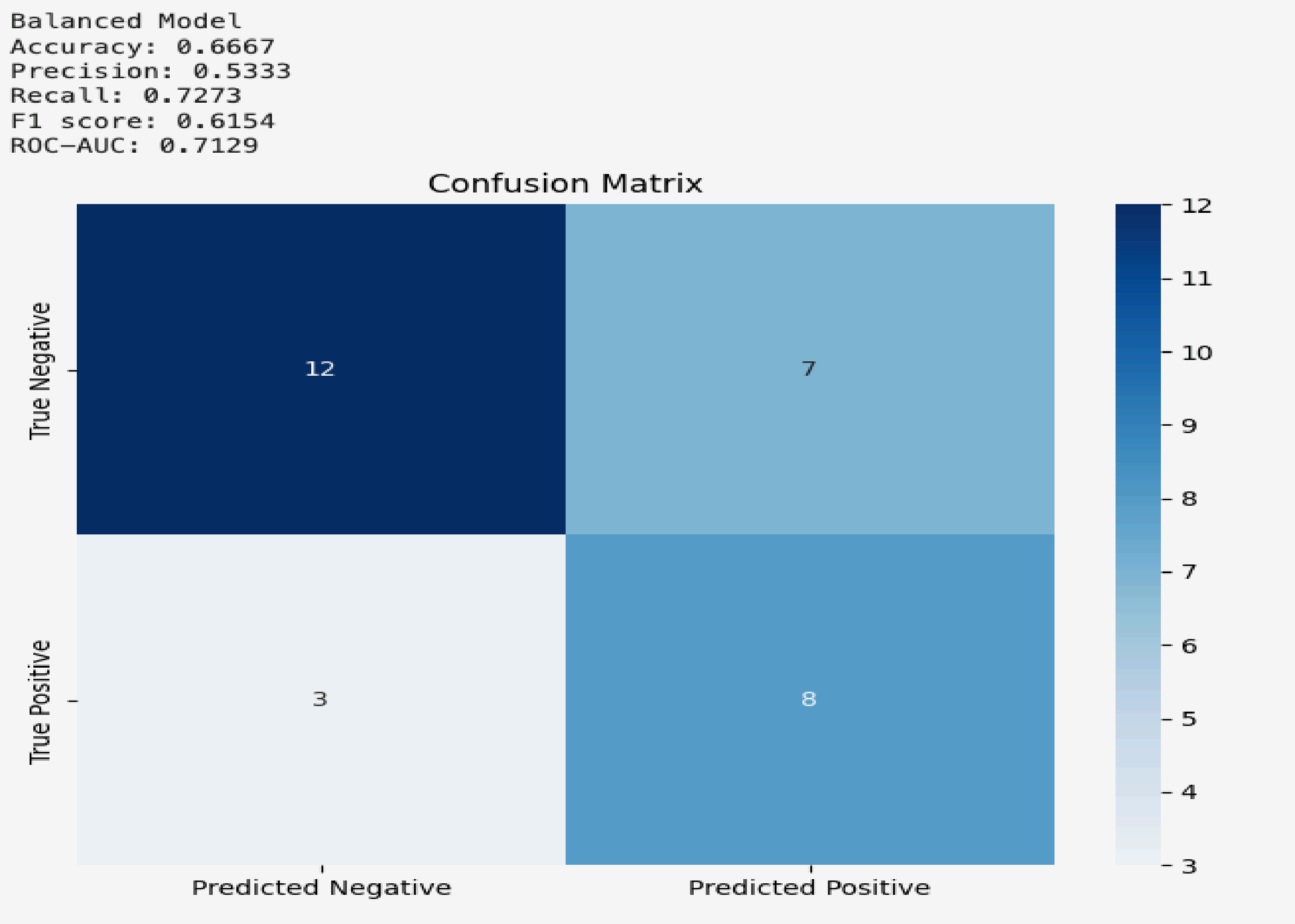
We created two variations of the Logistic Regression model: A basic one using the data as it was fed in and a balanced one with parameter tuning.



```
Basic Model
Accuracy: 0.6000
Precision: 0.4706
Recall: 0.7273
F1 score: 0.5714
ROC-AUC: 0.7273
```

Figure Name: Classification confusion matrix for the basic model

The basic model has an accuracy of 60% and a recall of 72.73% meaning that it is doing well to predict the outcome of a game. The precision rate at 47.06% is low meaning that it incorrectly predicts when the home team actually wins. This balance between recall and precision shows that it is biased towards predicting loss outcomes for a team. In addition, the F1 score was 57.14%. The confusion matrix shows that there were 9 false positives but also it correctly identifies 10 losses and 8 wins. This is reflected in the lower precision score.



```
Balanced Model
Accuracy: 0.6667
Precision: 0.5333
Recall: 0.7273
F1 score: 0.6154
ROC-AUC: 0.7129
```

Figure Name: Classification confusion matrix for the balanced model

The balanced model includes a regularization parameter of 0.1 improving the accuracy to 66.67% and F1 score to 61.54%. A higher F1 score means that there is a better balance of precision and recall. This model compared to the basic one predicts less false positives. This balanced model is also better at predicting losses. The confusion matrix displays that the model correctly identified 12 losses and 8 wins. The recall percentage remains because it is consistent in identifying win outcomes.

Random Forest

We created a variant of the Random Forest classifier to capture nonlinear relationships that would not have been caught by regression. Initially, we had to filter out the NBA teams that were irrelevant to the regular season. The feature engineering consisted of computing home advantage, rolling averages, and momentum metrics for both teams. We also took into consideration recent wins.

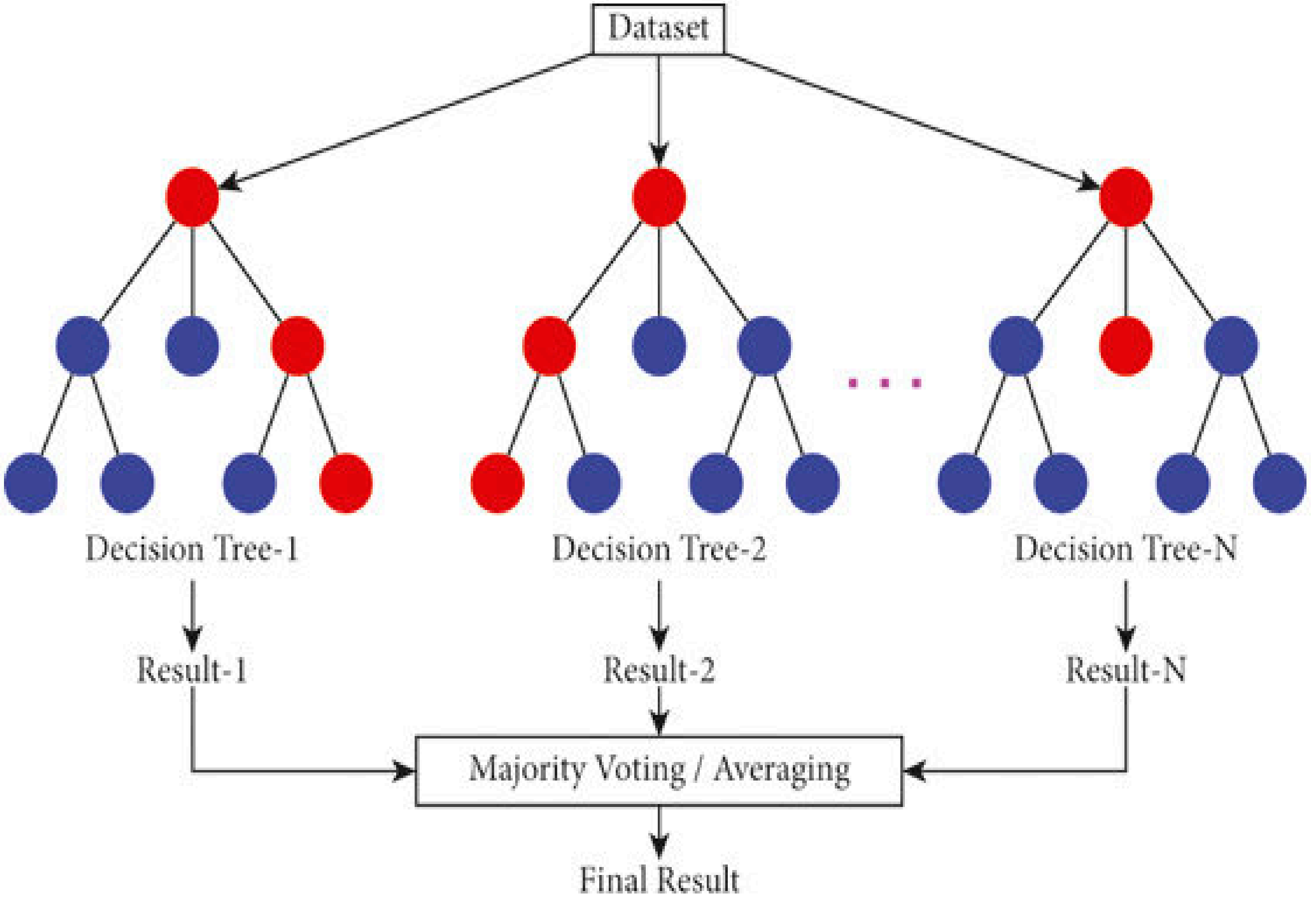
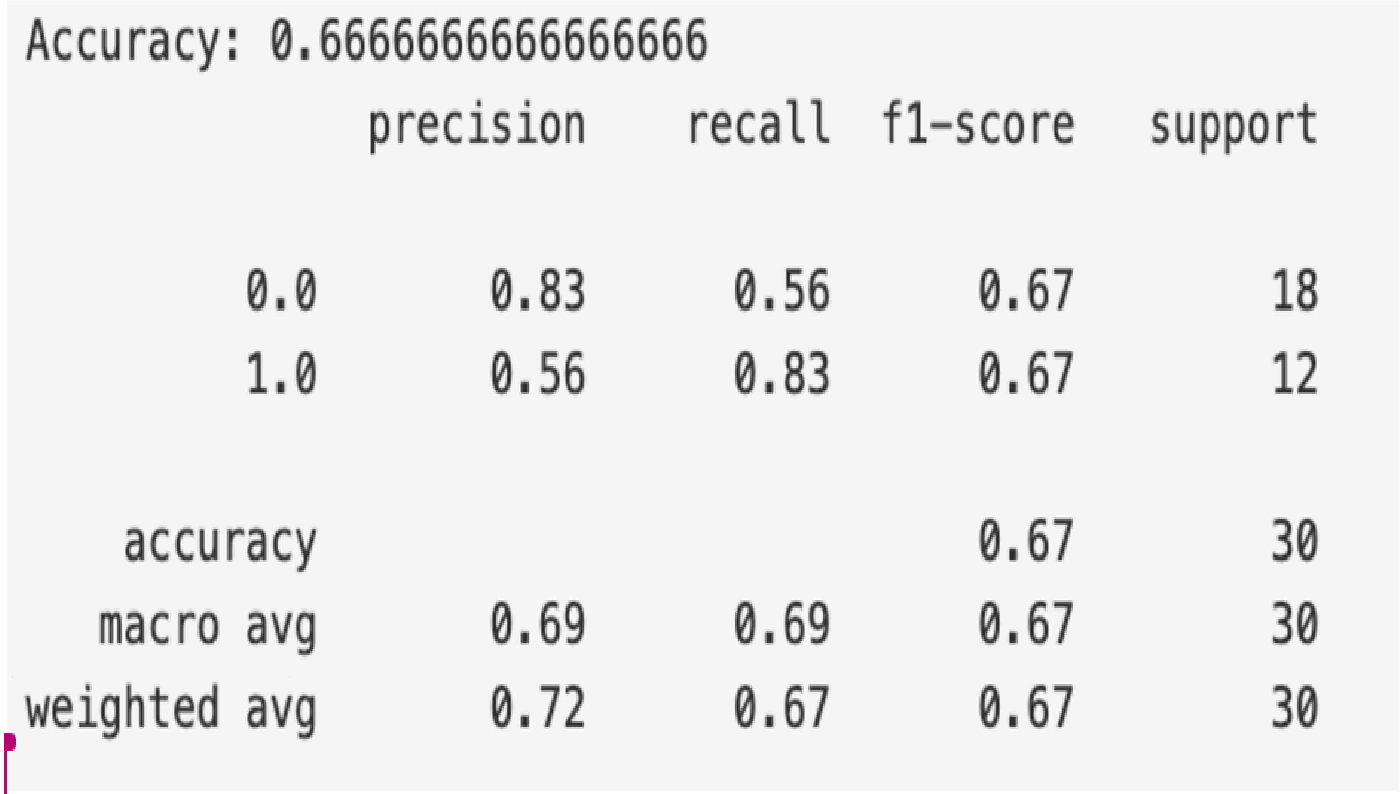


Figure Name: Random Forest Visualization [9]

The training split consists of data from earlier games and the test data was from more recent games. Iteratively we move the window backward to simulate multiple training and testing splits. This was to make sure that there was no data leakage. We applied PCA to reduce the feature space and used a parameter grid to tune the hyperparameters.



```
Accuracy: 0.6666666666666666
              precision    recall  f1-score   support

         0.0       0.83      0.56      0.67        18
         1.0       0.56      0.83      0.67        12

    accuracy                           0.67        30
   macro avg       0.69      0.69      0.67        30
weighted avg       0.72      0.67      0.67        30
```

Figure Name: Quantitative Metrics for Random Forest

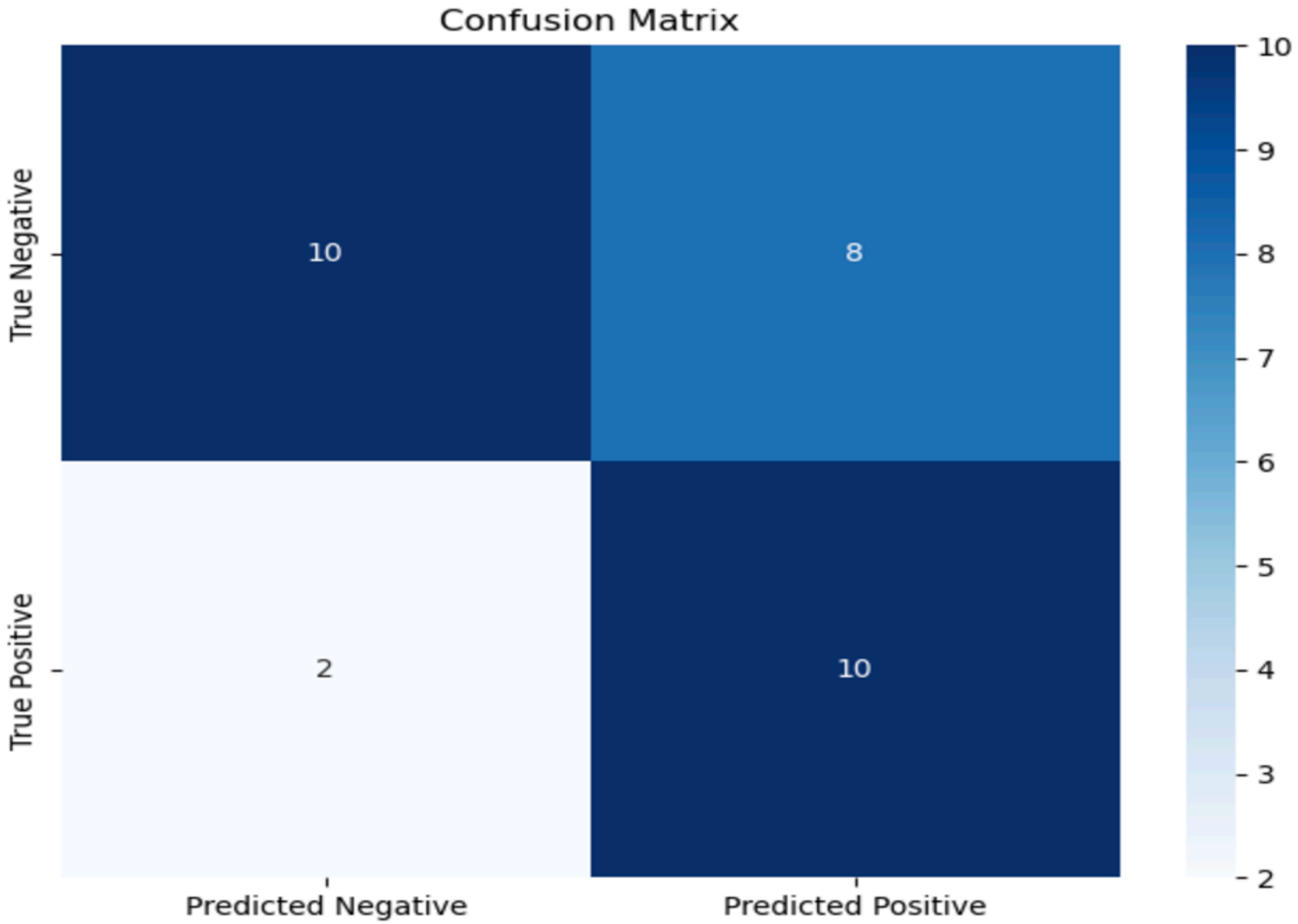


Figure Name: Classification confusion matrix for the Random Forest Model

This model has an accuracy of 66.67%. While this value is on the higher end it may not be the best metric if certain classes are imbalanced. The precision for class 0 is 83% meaning that the model is highly precise in predicting losses. The precision for class 1 is 56% suggesting that the model is over predicting wins while struggling to differentiate based on positive instances. The weighted F1 score of 67% indicates a better balance between precision and recall. Overall this model performed better than the balanced logistic regression model because there is the existence of nonlinear relationships. While the model correctly classified 10 positives and 10 negatives there was a high number of misclassified negative instances as positives.

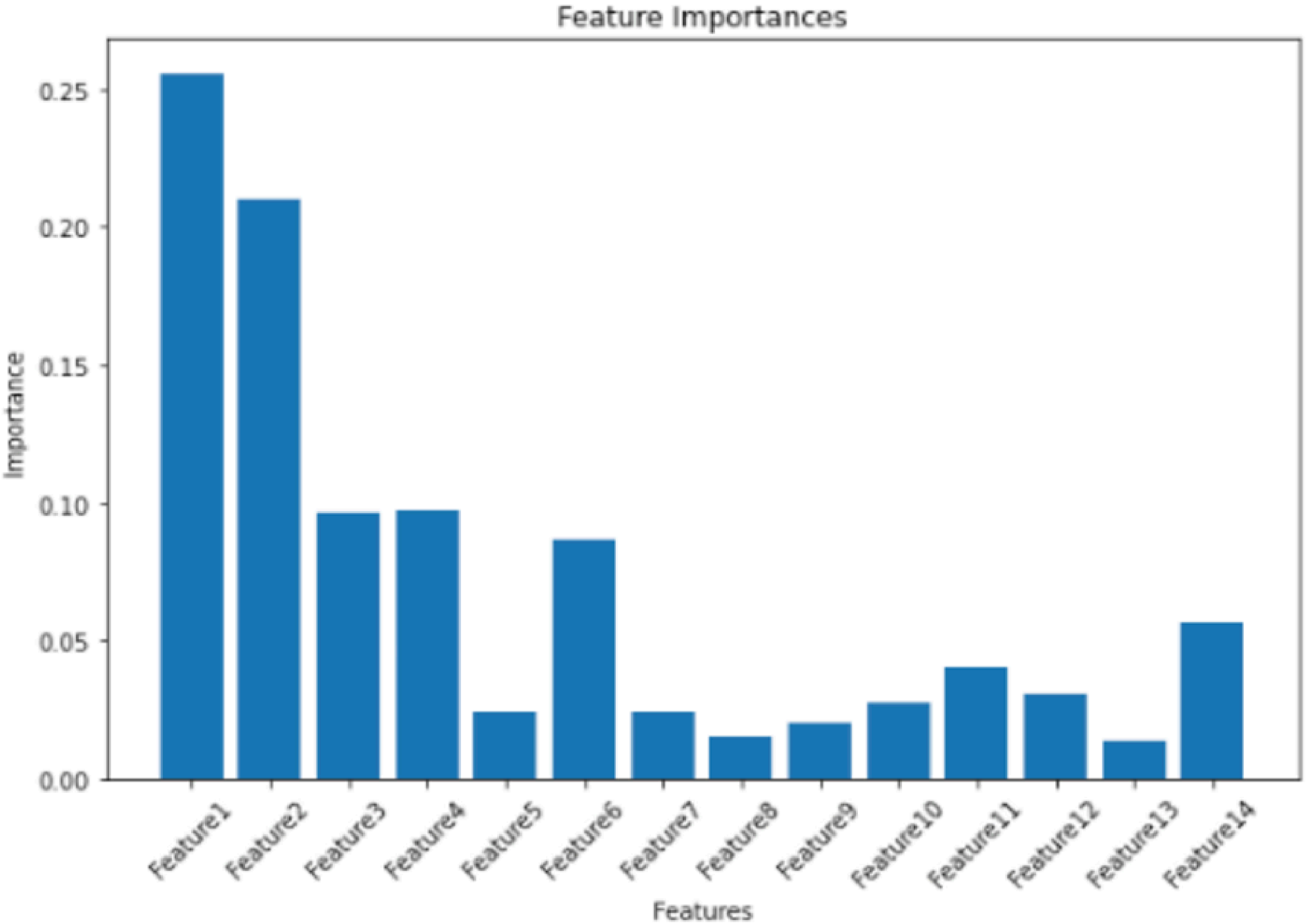


Figure Name: Feature Importances across Random Forest

The most important features to the model predictions are feature 1 and 2. Features 3, 4, 6, and 14 are also of moderate importance.

As seen by the accuracy and other metrics the Random Forest is able to handle non-linear feature relationships. We are also able to see which features are of more importance. An accuracy of 66% is not very high which means that the model is still not capturing all the patterns. The precision score of 56% for the positive class means that there are potential issues with noisy data. The higher recall for the positive class came at the cost of lower precision which increased the number of false positives. To improve this model we need to address the class imbalance by oversampling the 1 (minority) and undersampling the 0 (majority). We can also try to experiment with ensemble models like XGBoost for potentially better performance.

Support Vector Machine

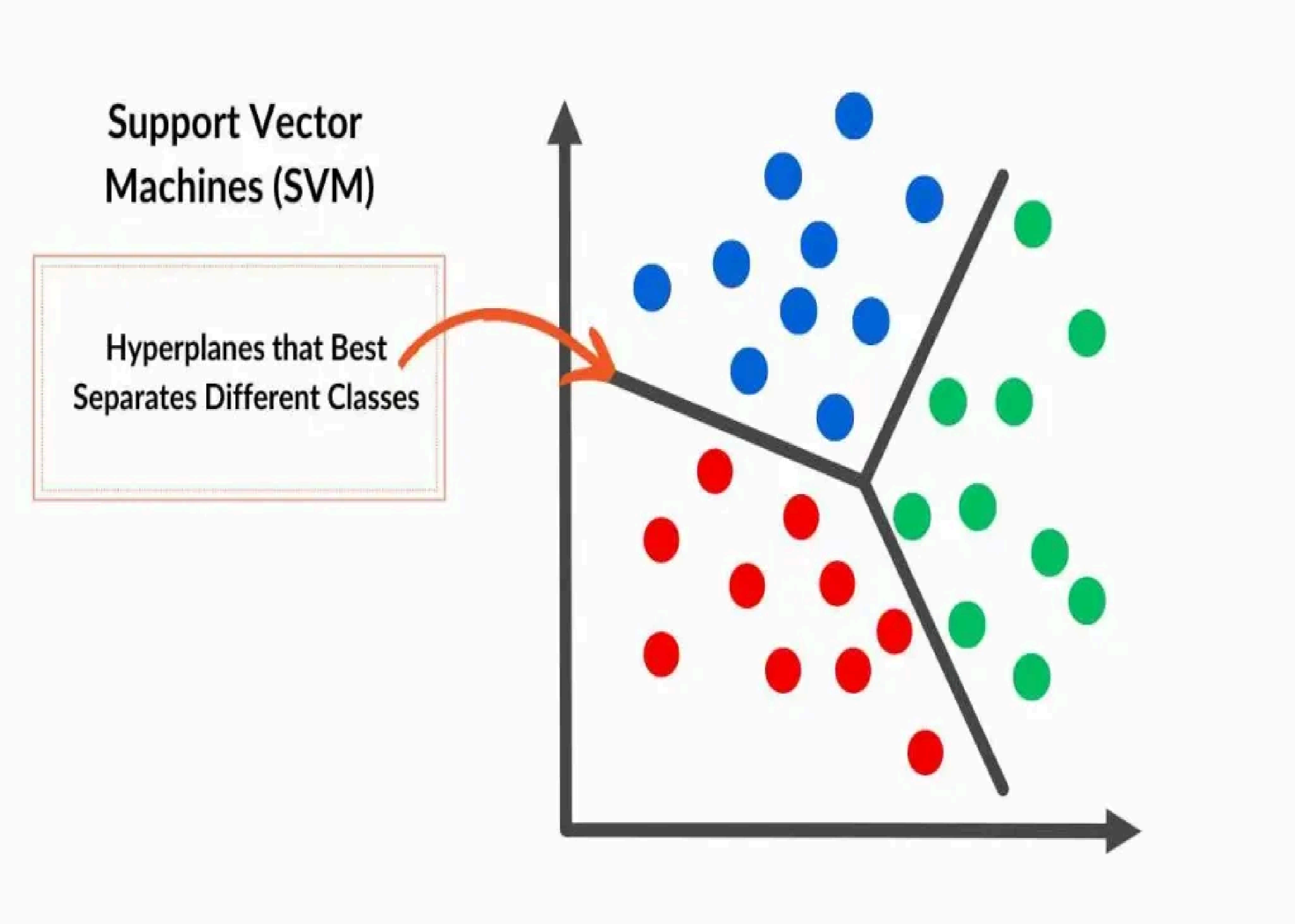


Figure Name: Support Vector Machine Visualization [11]

We have created a variant of a Support Vector Machine that classifies and predicts wins vs losses of NBA games through a decision boundary, called a hyperplane. The SVM attempts to find the best boundary to separate the different classes and maximize the margins between them. The approach to SVM was very similar to that of the random forest. First, we filtered out the NBA teams that did not play in the regular season. Next, we conducted feature engineering by computing advantage, rolling averages, and momentum metrics for the various teams, considering recent wins and losses.

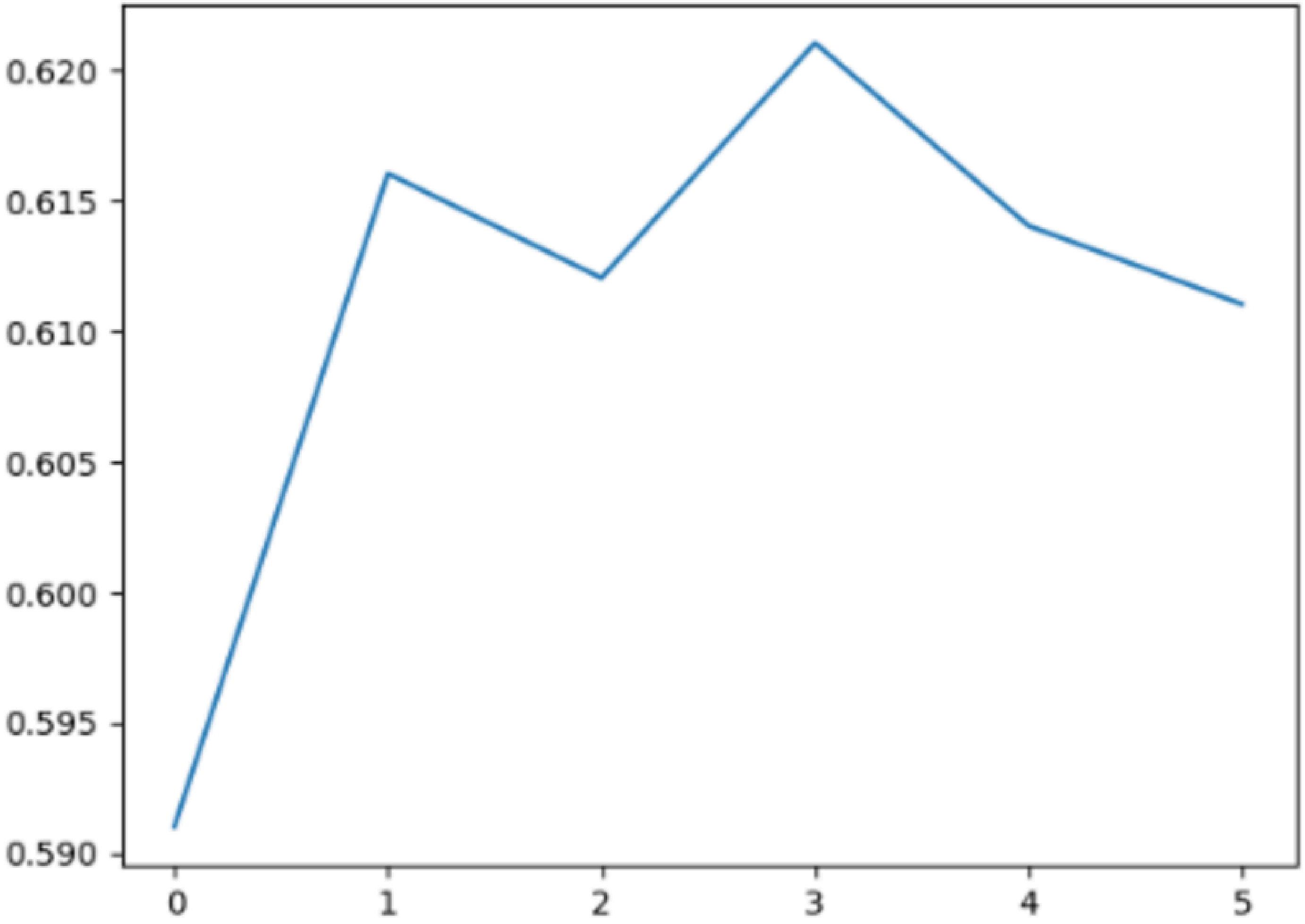


Figure Name: Line Graph for Regularization Parameter Tuning for SVM

The model specifically used a linear kernel with a regularization parameter of 1, which we found to provide the optimal balance between fitting the data while maintaining generalization. We tuned this parameter by testing a variety of regularization parameters from 0.001 to 100 and graphing and interpreting the classification accuracy and confusion matrix for each. As for the selection of features, we used a threshold of 0.55 to identify the most predictive components and applied these.

```
Cumulative Accuracy: 0.621
```

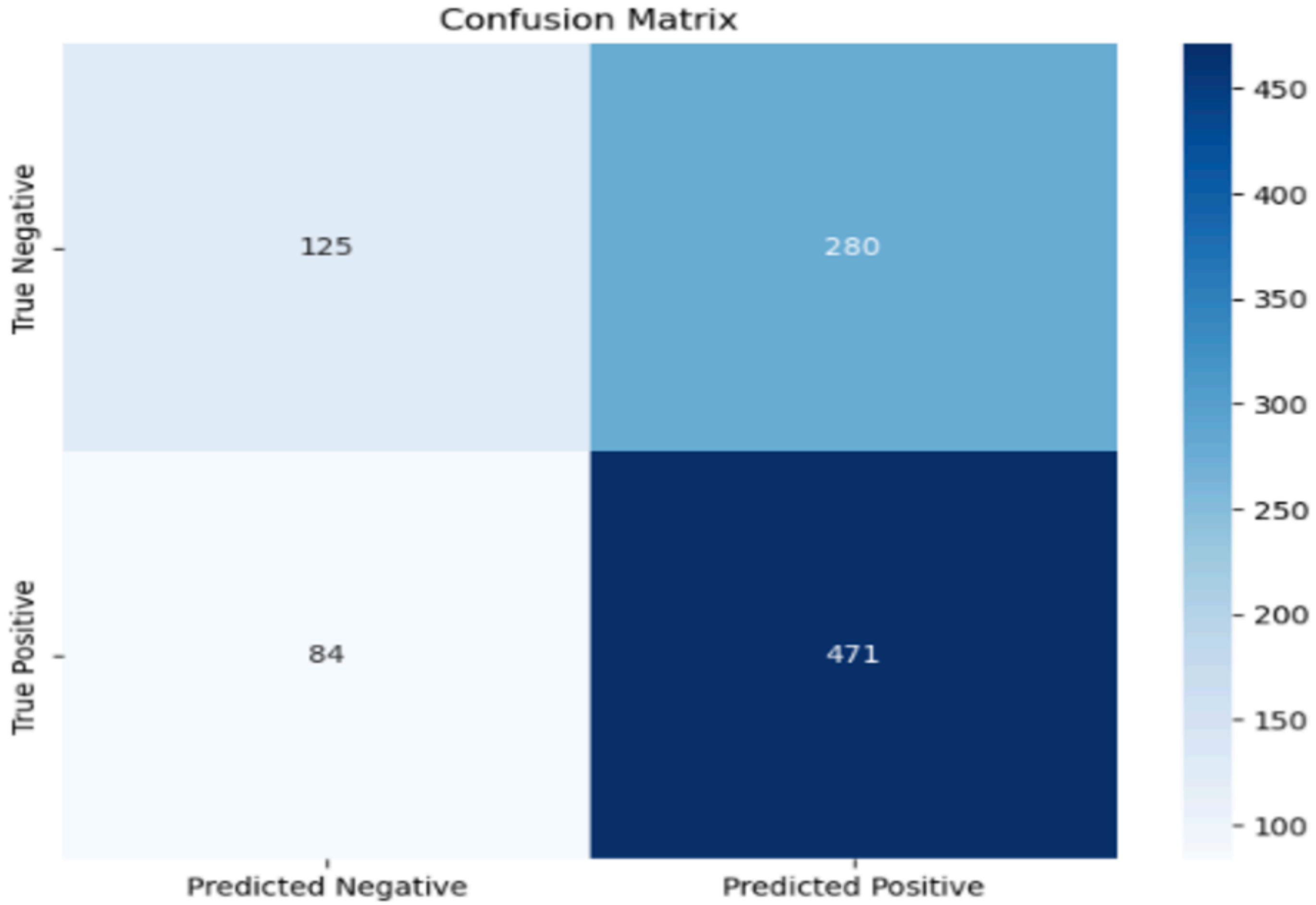
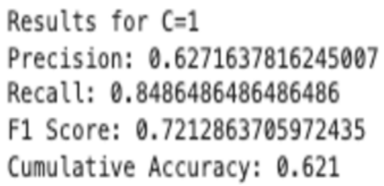


Figure Name: Classification Confusion Matrix for SVM



```
Results for C=1
Precision: 0.6271637816245007
Recall: 0.8486486486486486
F1 Score: 0.7212863705972435
Cumulative Accuracy: 0.621
```

Figure Name: Quantitative Results for SVM

Ultimately, the model achieved approximately 62% accuracy in predicting game outcomes correctly. Our precision was somewhat low at approximately 63% due to the model's predictability to choose win, perhaps due to a class imbalance. However, the C value chosen allows the best generalizability, and using completely balanced classes provided lower results. The recall was high at approximately 85% which was very high, detailing the opposite to our low precision, meaning our model was good at predicting wins by being generous. The decision boundary can be said to be more biased to wins. The F1 score remains in a fair place at 72%. This is very notable due to the vast number of variables that could affect each outcome. We worked to beat a 50% baseline accuracy, and we were able to do this to a good extent.

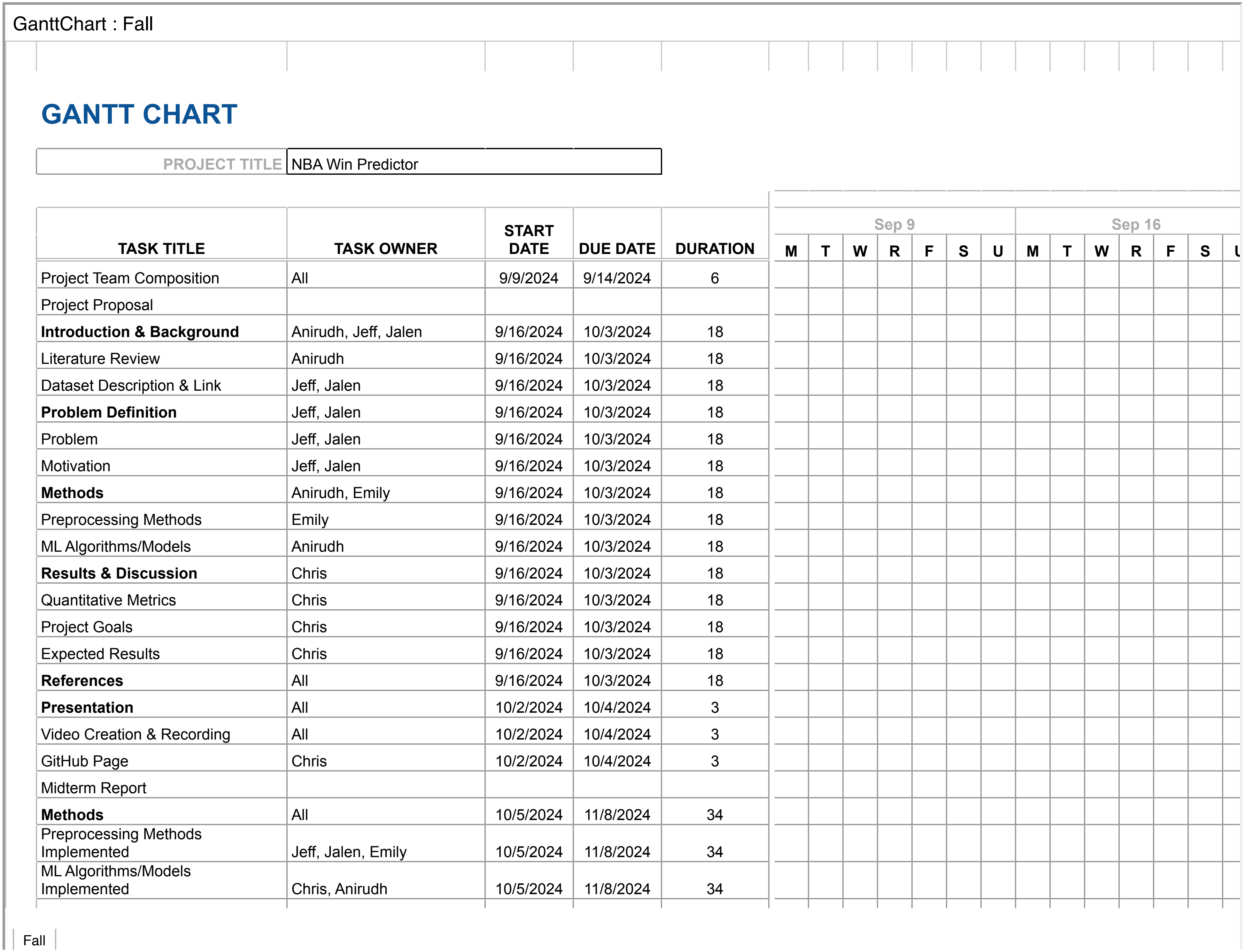
Model Comparisons

Each of these models had their own strengths, weaknesses, and trade-offs. The logistic regression model served as a baseline of how we could predict game outcomes. This model struggled to capture non-linear relationships while showing a high recall percentage (72.73%) with low precision due to a bias in predicting losses. The Random Forest however was much better at handling non-linear feature interactions but surprisingly had the same accuracy as the balanced logistic model (66.67%). However the higher F1 score of (67%) on the Random Forest makes it better at capturing complex patterns. This is why accuracy is not the always the full story in terms of model metrics. The feature importance analysis and utilizing PCA allowed us to achieve good results but it struggled with class imbalance leading to lower precision for predicting wins. The SVM did not have as high accuracy as either model (62%) and had a high recall (85%). SVM, while computationally more intensive, was highly successful in identifying actual wins. It also had the highest F1 score (72%) because the linear kernel captures patterns that align well with the decision boundary for games. The SVM predicts wins more effectively but this also may lead to more false positives.

Next Steps

The baseline accuracy range that we were targeting was 50% upwards to 70%. While the results of all of the models models fall within that range there is more that can be done to increase accuracy. The biggest place for exploration is within our dataset. Although our dataset was quite comprehensive with teams, games, matchups, and statistics, it does not have much data on the individual players or other factors. Adding more features will make the model more complex and add ambiguity, but we could conduct more dimensionality reduction and linear combinations to potentially find better components to use to predict game outcomes. Having access to seemingly crucial knowledge of players such as injuries, individual performance, transfers, and history could prove to bring further understanding of a team’s performance. For example, knowing LeBron was traded to another team would give a properly trained model better classification knowledge on how the team would perform with a newly added superstar. There could even be complex relationships that cannot be predicted by the naked eye such as the travel time, weather patterns, and other entities that would originally be classified as “noise”, such as what is seen in models in fields such as the stock market. Also, our data analysis is focused on the home teams and statistics for the home team. There could potentially be some “alpha” found within looking into the away team statistics and analyzing and comparing those statistics in depth. Ultimately, our models performed quite well in regards to our original plan and goals, but there is a lot more to explore and refine in the field of predicting NBA wins.

Gantt Chart



GanttChart : Fall

GANTT CHART

PROJECT TITLE	NBA Win Predictor

					Sep 9							Sep 16						
TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	M	T	W	R	F	S	U	M	T	W	R	F	S	U
Project Team Composition	All	9/9/2024	9/14/2024	6														
Project Proposal																		
Introduction & Background	Anirudh, Jeff, Jalen	9/16/2024	10/3/2024	18														
Literature Review	Anirudh	9/16/2024	10/3/2024	18														
Dataset Description & Link	Jeff, Jalen	9/16/2024	10/3/2024	18														
Problem Definition	Jeff, Jalen	9/16/2024	10/3/2024	18														
Problem	Jeff, Jalen	9/16/2024	10/3/2024	18														
Motivation	Jeff, Jalen	9/16/2024	10/3/2024	18														
Methods	Anirudh, Emily	9/16/2024	10/3/2024	18														
Preprocessing Methods	Emily	9/16/2024	10/3/2024	18														
ML Algorithms/Models	Anirudh	9/16/2024	10/3/2024	18														
Results & Discussion	Chris	9/16/2024	10/3/2024	18														
Quantitative Metrics	Chris	9/16/2024	10/3/2024	18														
Project Goals	Chris	9/16/2024	10/3/2024	18														
Expected Results	Chris	9/16/2024	10/3/2024	18														
References	All	9/16/2024	10/3/2024	18														
Presentation	All	10/2/2024	10/4/2024	3														
Video Creation & Recording	All	10/2/2024	10/4/2024	3														
GitHub Page	Chris	10/2/2024	10/4/2024	3														
Midterm Report																		
Methods	All	10/5/2024	11/8/2024	34														
Preprocessing Methods Implemented	Jeff, Jalen, Emily	10/5/2024	11/8/2024	34														
ML Algorithms/Models Implemented	Chris, Anirudh	10/5/2024	11/8/2024	34														

Link: https://docs.google.com/spreadsheets/d/1XQ0EQoz8NIh5HJAoZL4YnIaZN_QbHBAhaE8E8BESF-k/

Contribution Chart

Name	Contributions
Anirudh	Literature Review, ML Algorithms/Models, Video Creation Midterm contributions: Midterm report, ML model, Analysis Final contributions: Random forest model tuning, Final report, Final slides, Analysis, Video
Chris	Results & Discussion, Video Creation, GitHub Page Midterm contributions: Model Optimization, Model Visualization, Quantitative Measures, GitHub Page Final contributions: SVM code refactorign, Final report/slides, Analysis, GitHub Pages, Video
Emily	Preprocessing Methods, Video Creation Midterm contributions: Data preprocessing, Midterm report, Analysis, Visualizations Final contributions: Data preprocessing, SVM model creation, Analysis, Visualizations
Jalen	Dataset Description, Problem Definition, Video Creation, Gantt Chart Midterm contributions: Midterm report, Data preprocessing, Analysis Final Contributions: Random forest model creation, Code refactoring, Analysis, Visualizations
Jeff	Dataset Description, Problem Definition, Video Creation Midterm contributions: Data preprocessing, ML Modeling, Visualizations, Quantitative Measures Final Contributions: Data preprocessing, Random forest model creation, Code refactoring, Analysis, Visualizations

References

1. M. Beckler, H. Wang, and M. Papamichael, "NBA Oracle." Accessed: Oct. 04, 2024. [Online]. Available: http://www.mbeckler.org/coursework/2008-2009/10701_report.pdf
2. C. Osken and C. Onay, "Predicting the winning team in basketball: A novel approach," Heliyon, vol. 8, no. 12, p. e12189, Dec. 2022, doi: https://doi.org/10.1016/j.heliyon.2022.e12189.
3. J. Wang, "Predictive Analysis of NBA Game Outcomes through Machine Learning," Oct. 2023, doi: https://doi.org/10.1145/3635638.3635646.
4. H. Ju and H. Zhang, "Application of Multiple Linear Regression Model in the Sustainable Development of National Traditional Sports," Applied Mathematics and Nonlinear Sciences, vol. 8, no. 2, pp. 3033–3042, Jul. 2023, doi: https://doi.org/10.2478/amns.2023.2.00019.
5. Zheng Songling and M. Xi, "An Improved Logistic Regression Method for Assessing the Performance of Track and Field Sports," Computational Intelligence and Neuroscience, vol. 2022, pp. 1–10, Aug. 2022, doi: https://doi.org/10.1155/2022/6341495.
6. C. Walsh and A. Joshi, "Machine learning for sports betting: Should model selection be based on accuracy or calibration?," Machine Learning with Applications, vol. 16, p. 100539, Jun. 2024, doi: https://doi.org/10.1016/j.mlwa.2024.100539.

7. J. P. Dmochowski, "A statistical theory of optimal decision-making in sports betting," vol. 18, no. 6, pp. e0287601–e0287601, Jun. 2023, doi: https://doi.org/10.1371/journal.pone.0287601.
8. U. Matej, Š. Gustav, H. Ondřej, and Ž. Filip, "Optimal sports betting strategies in practice: an experimental review," IMA Journal of Management Mathematics, vol. 32, no. 4, pp. 465–489, Feb. 2021, doi: https://doi.org/10.1093/imaman/dpaa029.
9. https://www.researchgate.net/journal/Complexity-1099-0526
10. https://towardsdatascience.com/logistic-regression-explained-9ee73cede081
11. https://spotintelligence.com/2024/05/06/support-vector-machines-svm/