Predicting College Basketball Success

Luigi Manieri

Alma Mater Studiorum Bologna

May 6, 2025

Outline

Project Overview

- **Objective:** Predict whether the road team wins a game.
- **Dataset:** Sports match data from Kaggle competition.
  - https://www.kaggle.com/competitions/wfusummer2018
- **Target variable:** `Win` (1 = Road Team Wins, 0 = Loss).

Dataset Overview

- Dataset details:
  - 1893 rows, 11 features, and 1 target column (Win)
  - Numerical features: *win, day, year, roadTeamPoints*
  - Categorical features: *month, weekday, time, roadTeam, locale, homeTeam, conference, OT*

```
Data columns (total 12 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Win             1893 non-null   int64
 1   Month           1893 non-null   object
 2   Day             1893 non-null   int64
 3   Year            1893 non-null   int64
 4   Weekday         1893 non-null   object
 5   Time            1888 non-null   object
 6   RoadTeam        1893 non-null   object
 7   Locale          1893 non-null   object
 8   HomeTeam        1893 non-null   object
 9   Conference      1893 non-null   object
 10  RoadTeamPoints  1893 non-null   int64
 11  OT              122 non-null    object
dtypes: int64(4), object(8)
```

Figure: Dataset info

Data Exploration

	Win	Month	Day	Year	Weekday	Time	RoadTeam	Locale	HomeTeam	Conference	RoadTeamPoints	OT
0	1	Jan	3	2018	Wed	9:00 pm/est	Missouri	@	South Carolina	SEC	79	NaN
1	0	Feb	10	2016	Wed	7:00 pm/est	Providence	@	Marquette	Big East	91	2OT
2	0	Jan	2	2016	Sat	2:00 pm/est	Tennessee	@	Auburn	SEC	77	NaN
3	0	Feb	3	2016	Wed	7:00 pm/est	Arkansas	@	Florida	SEC	83	NaN
4	1	Feb	18	2016	Thu	10:00 pm/est	Utah	@	UCLA	Pac-12	75	NaN

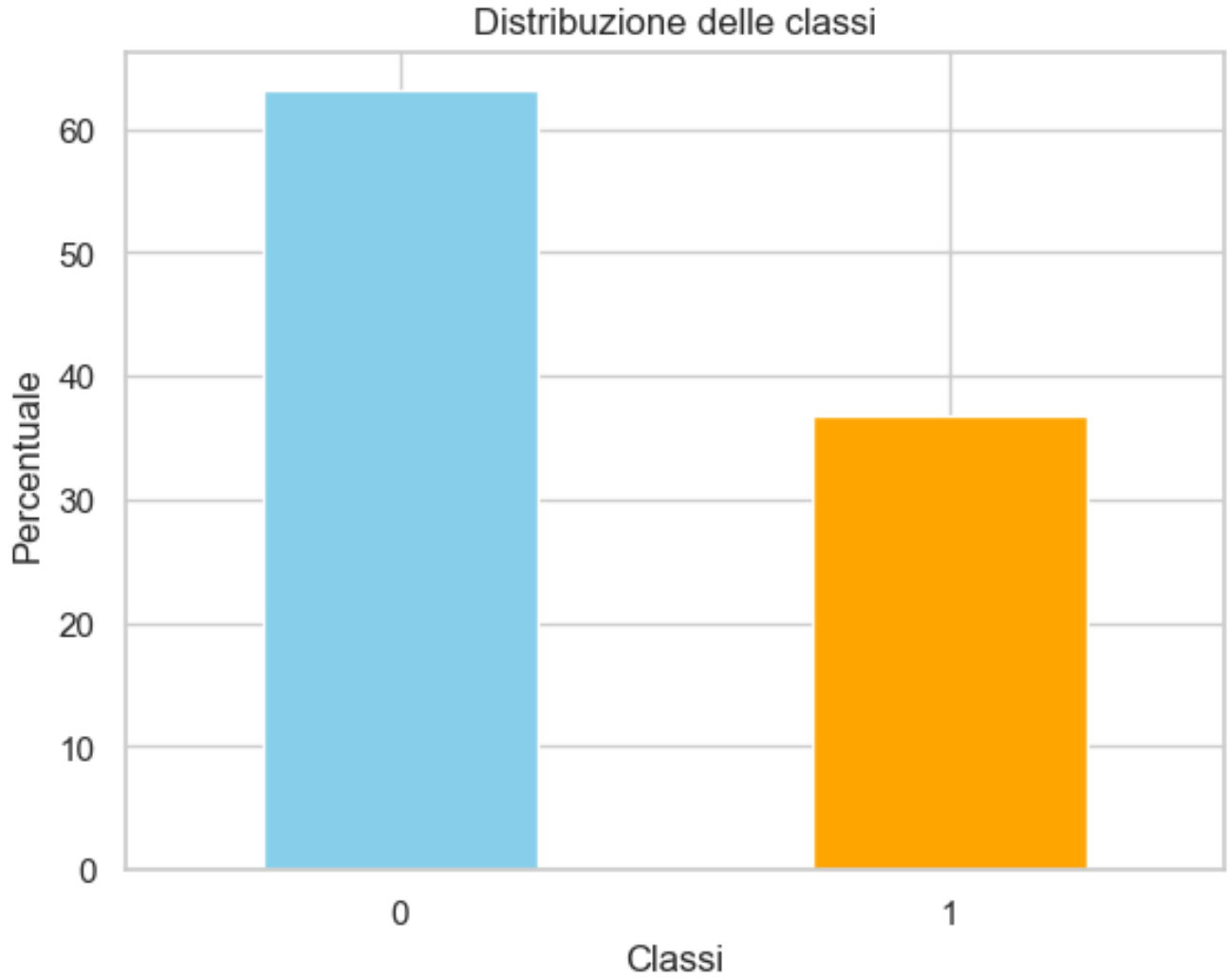
Figure: Head of the dataset

We can already see that:

- most of the features are object and need to be processed before training models
- some feature don't seem very informative (i.e. locale, month, day, weekday)
- there are some missing values (time and OT columns)

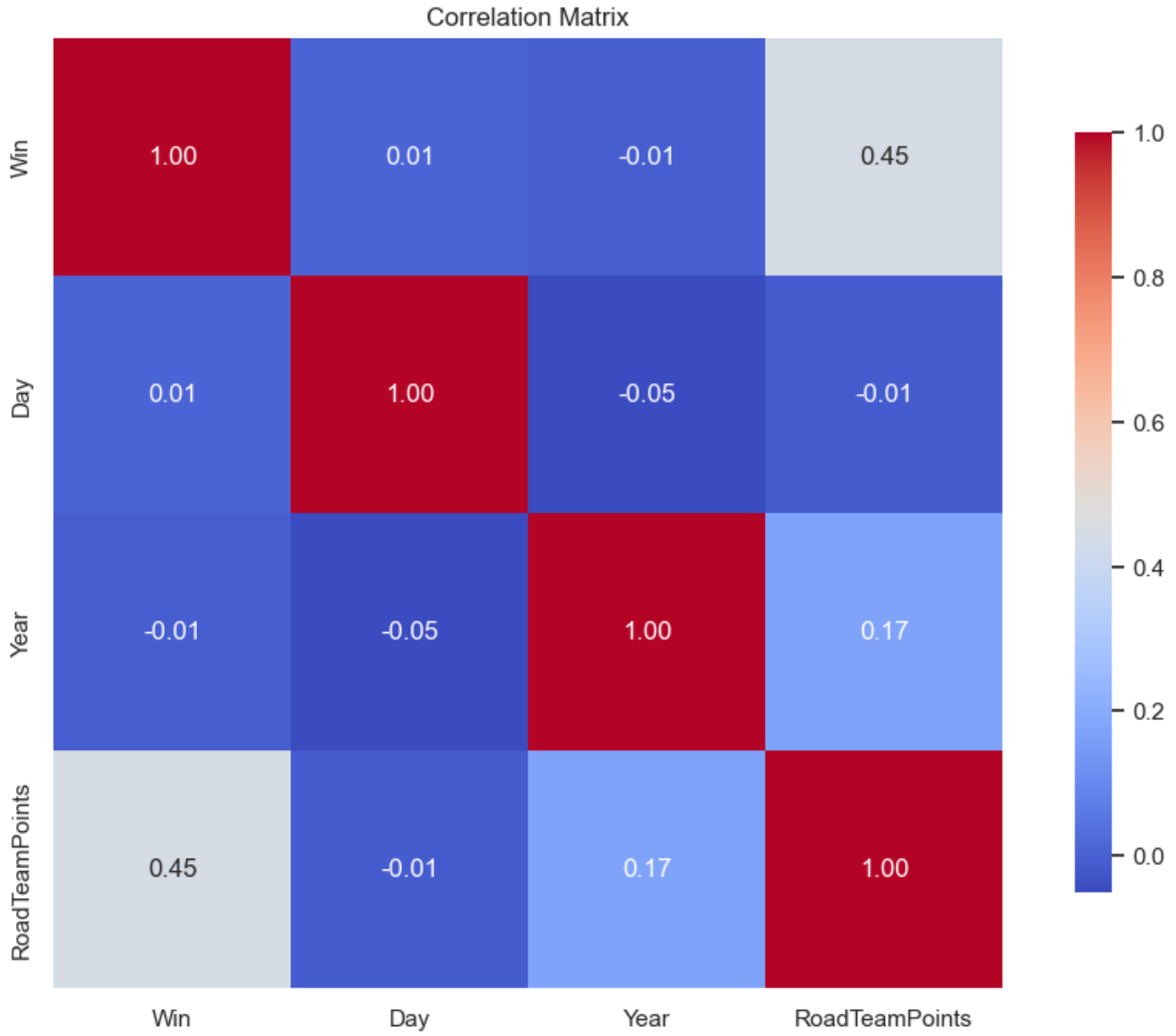
Data Visualization

- Class Balance:



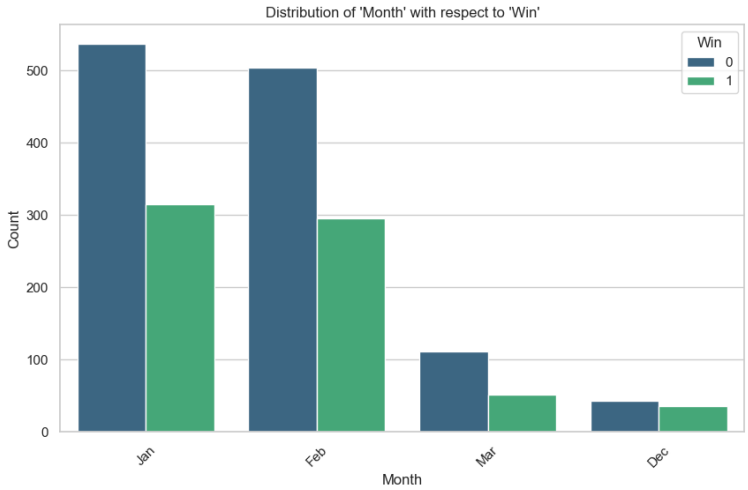
Data Visualization

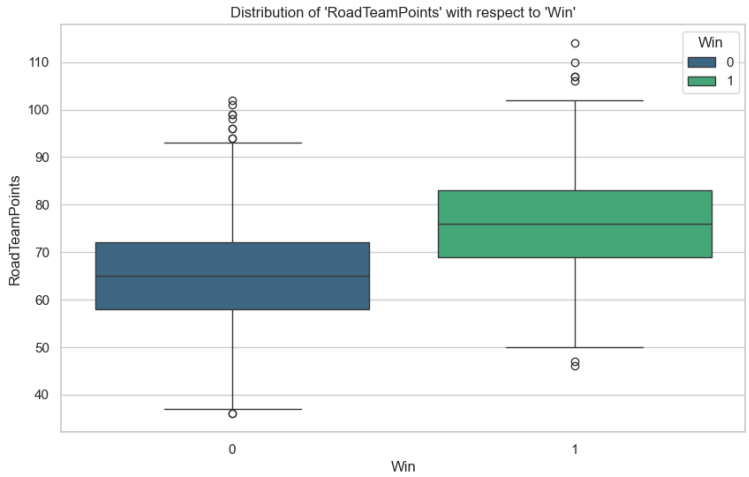
- Correlation matrix (for numerical features):

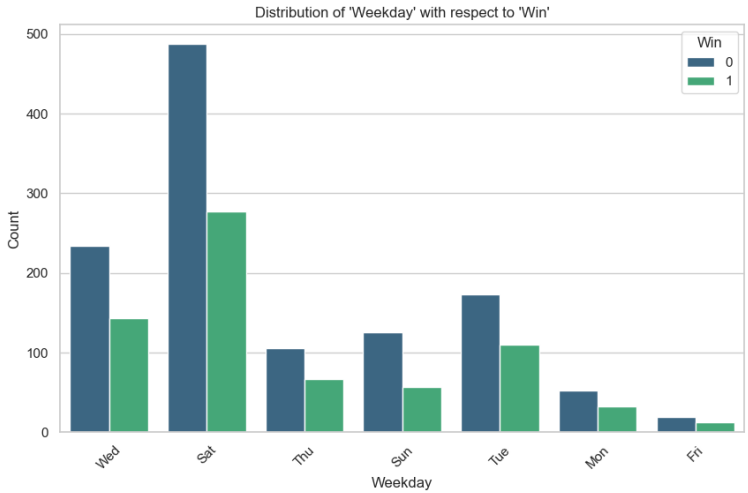


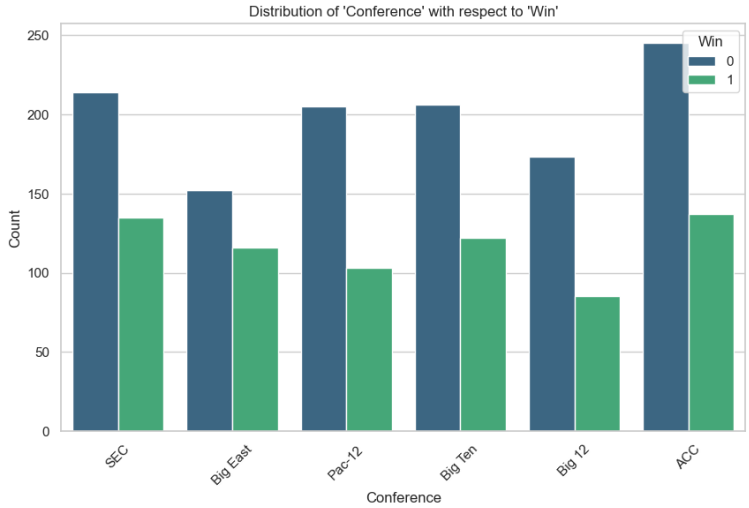
Data Visualization

- Feature vs Target









Data Preprocessing

- Deal with missing values and duplicates
  - in Time column $\rightarrow$ dropped
  - in OT column $\rightarrow$ substitute with placeholder value NOT (No OverTime)
  - duplicates $\rightarrow$ dropped
- Drop unnecessary columns:
  - Locale, Day, Weekday, Month
    - I tried to encode them instead of drop $\rightarrow$ no performance gain

Data Preprocessing

- One hot encoding
  - for HomeTeam and RoadTeam features
- Label encoding
  - for OT and Conference
- MinMaxScaling:
  - Year
- RoadTeamPoints was not scaled in order to give it more importance
  - it's the most informative feature

Data Preprocessing: Processed dataset

Final shape of train_df:
Elements: 1883, Features: 153

	Win	Year	Conference	RoadTeamPoints	OT	RoadTeam_Arizona	RoadTeam_Arizona State	RoadTeam_Arkansas	RoadTeam_Auburn	RoadTeam_Baylor	...	HomeTeam_Vanderbilt	HomeTeam_Villanova
0	1	0.0	5	79	3	0	0	0	0	0	...	0	0
1	0	0.5	2	91	0	0	0	0	0	0	...	0	0
2	0	0.5	5	77	3	0	0	0	0	0	...	0	0
3	0	0.5	5	83	3	0	0	1	0	0	...	0	0
4	1	0.5	4	75	3	0	0	0	0	0	...	0	0

5 rows × 153 columns

Figure: Final structure of the dataset

Data preparation and models

- Define features X and target variable y (Win)
- Applying SMOTE for class balancing
- Dataset division: 80% training set, 20% test set.
- Measures in output: Accuracy, Recall, F1-score, ROC AUC score
- Models considered:
  - Random Forest
  - AdaBoost
  - SVM
  - XGBoost
  - MLP

Model training

- Train the models with default hyperparameters
- Optimize them with GridSearch in order to find the best parameter configuration
  - Does not always result in better performance

	Model	Accuracy	Recall	F1 Score	ROC AUC
0	Random Forest	0.802105	0.803279	0.806584	0.802072
1	AdaBoost	0.812632	0.815574	0.817248	0.812549
2	SVM	0.789474	0.778689	0.791667	0.789777
3	XGBoost	0.783158	0.766393	0.784067	0.783630
4	MLP	0.816842	0.770492	0.812095	0.818146

	Model	Accuracy	Recall	F1 Score	ROC AUC
0	Best Random Forest	0.787368	0.778689	0.790021	0.787613
1	Best AdaBoost	0.797895	0.827869	0.808000	0.797051
2	Best SVM	0.789474	0.778689	0.791667	0.789777
3	Best XGBoost	0.802105	0.795082	0.804979	0.802303
4	Best MLP	0.816842	0.782787	0.814499	0.817800

Figure: Results before and after GridSearch

Confusion Matrix: Random Forest

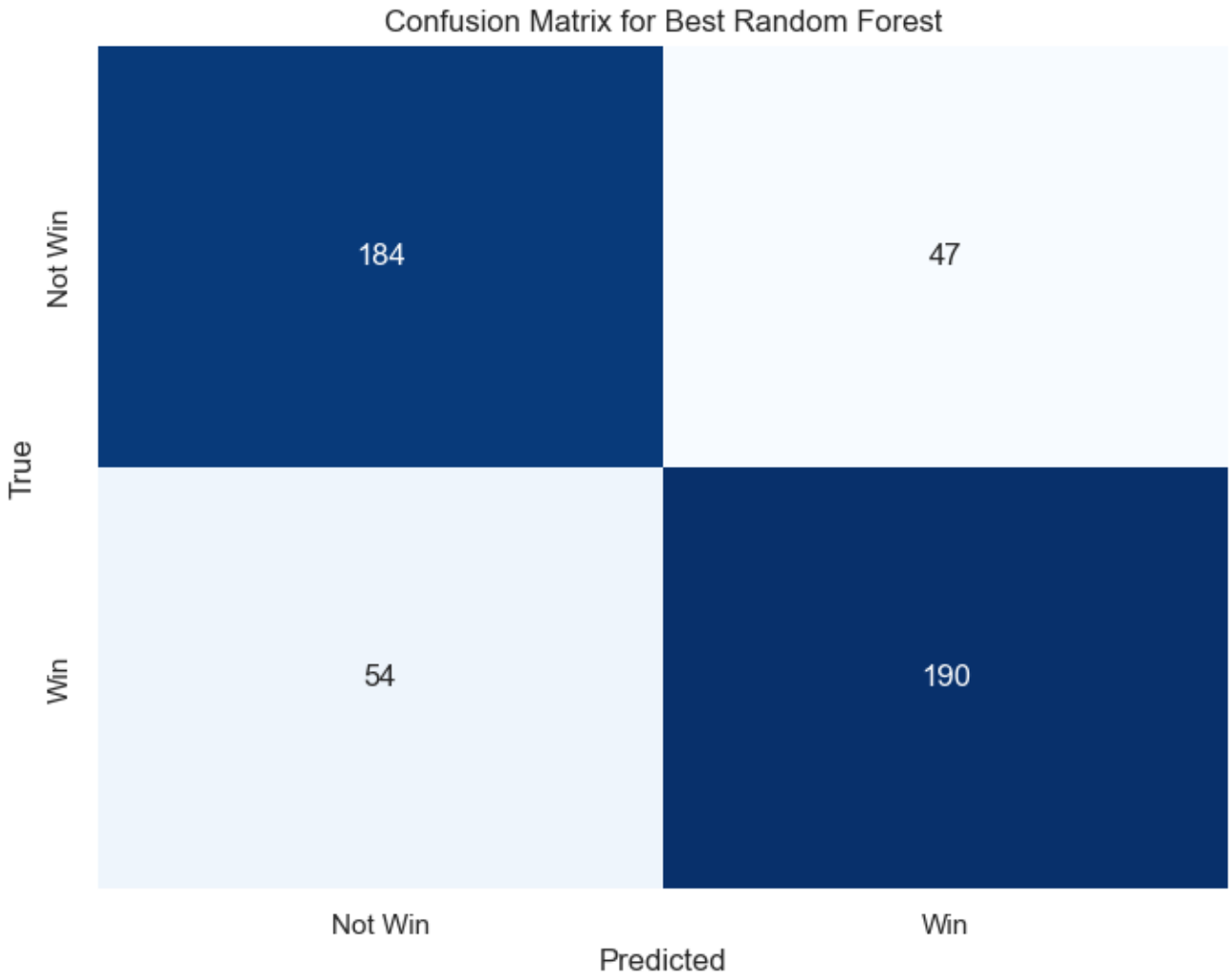


Figure: Confusion matrix for GS-optimized Random Forest

Confusion Matrix: AdaBoost

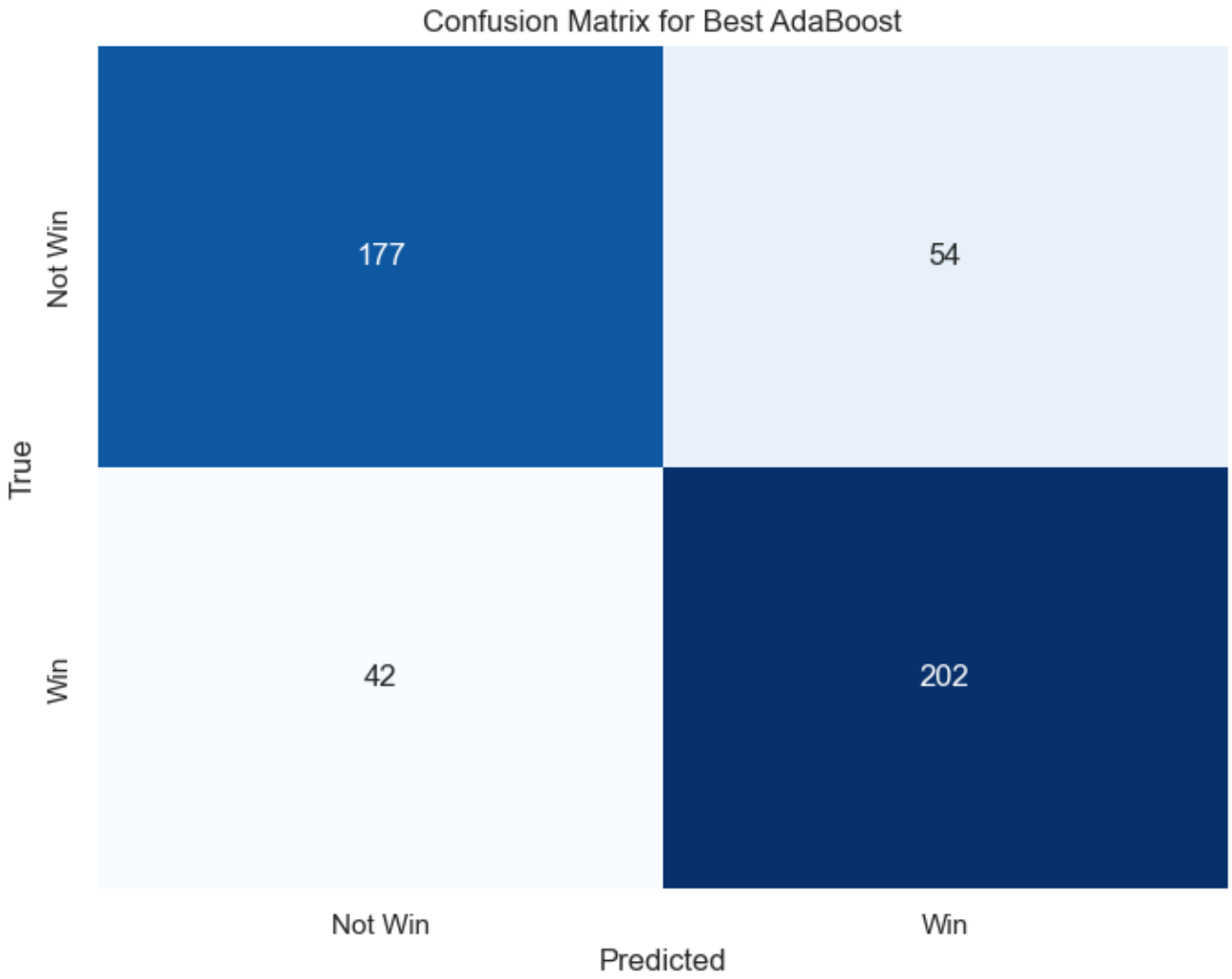


Figure: Confusion matrix for GS-optimized AdaBoost

Confusion Matrix: XGBoost

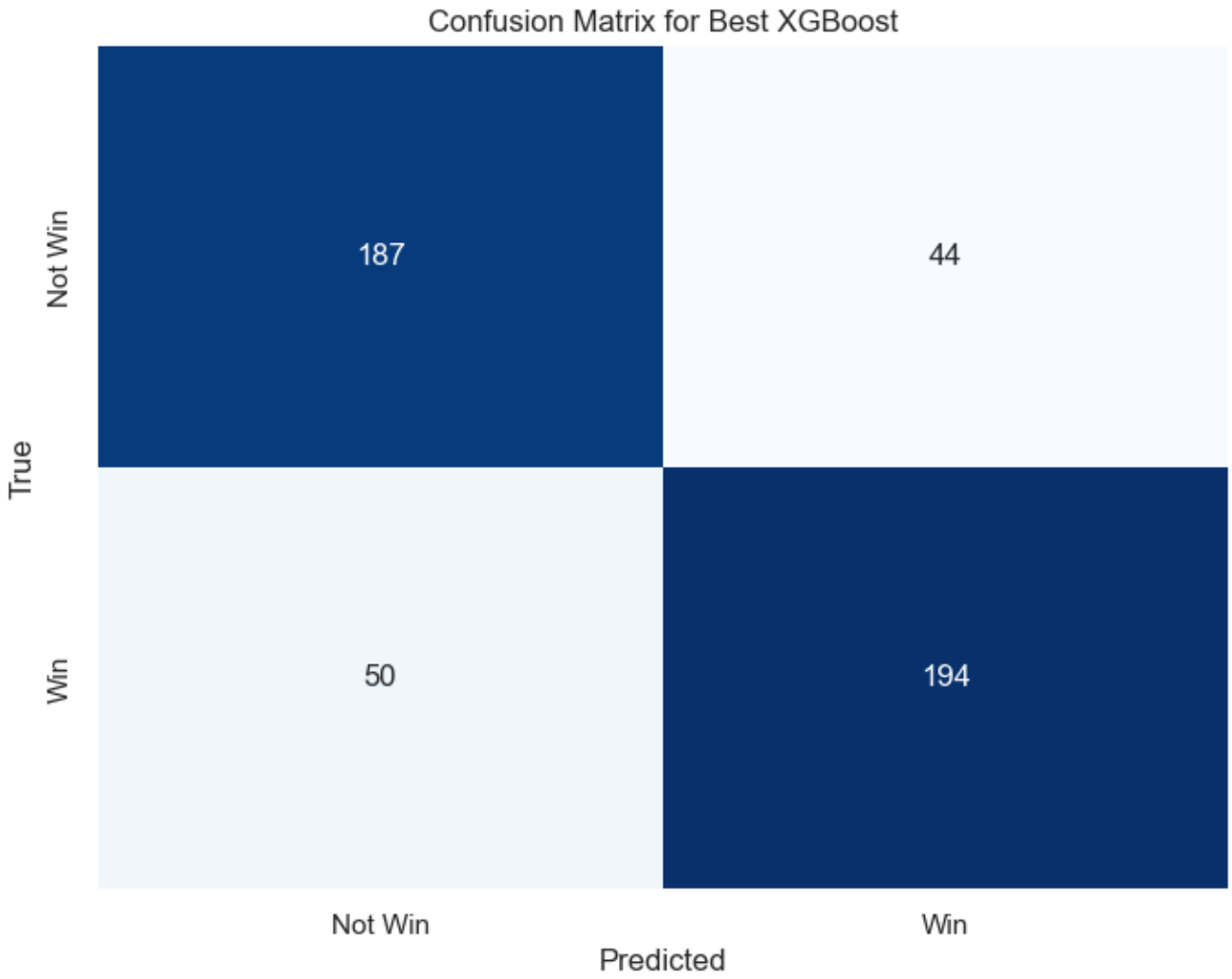


Figure: Confusion matrix for GS-optimized XGBoost

Confusion Matrix: SVM

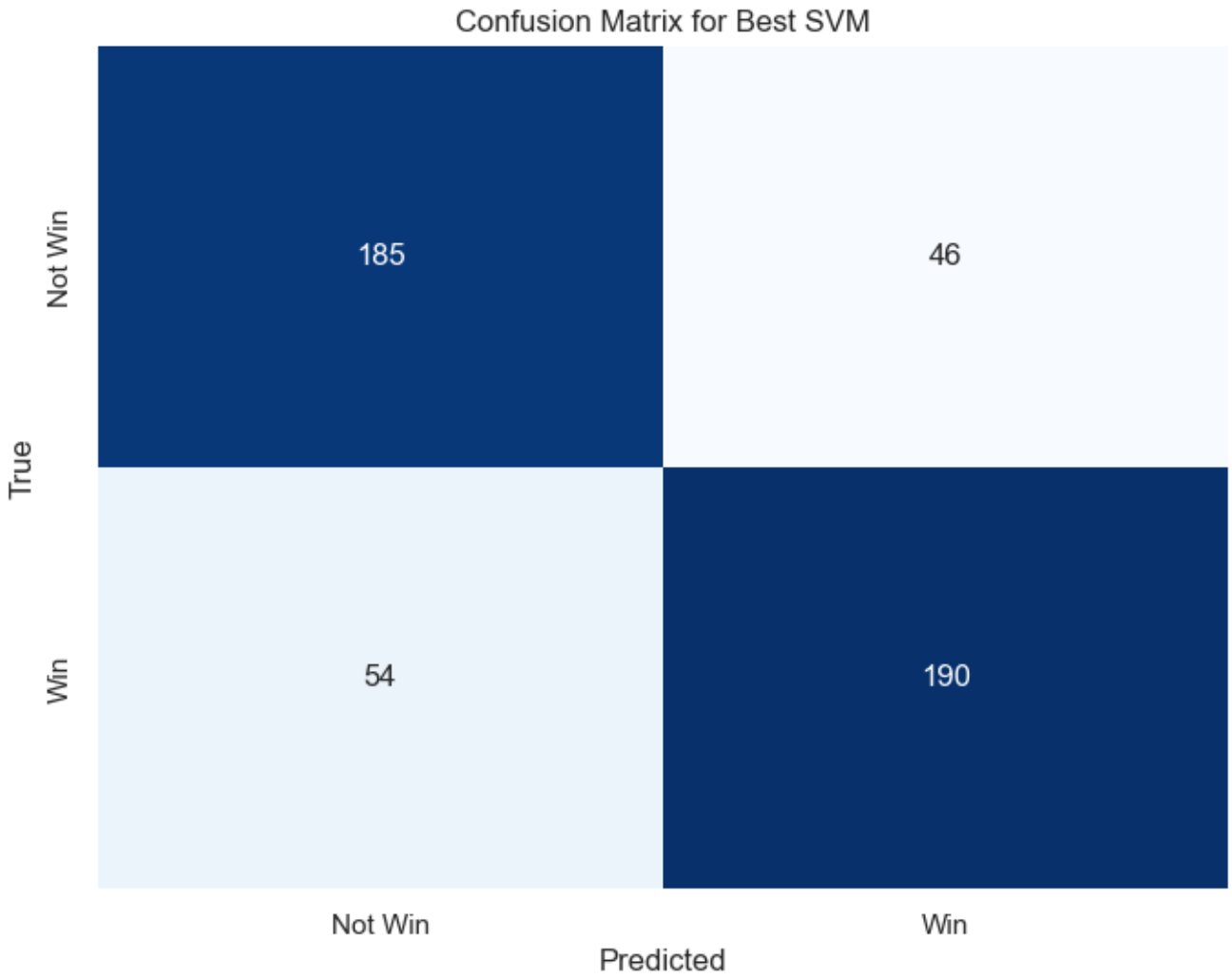


Figure: Confusion matrix for GS-optimized SVM

Confusion Matrix: MLP

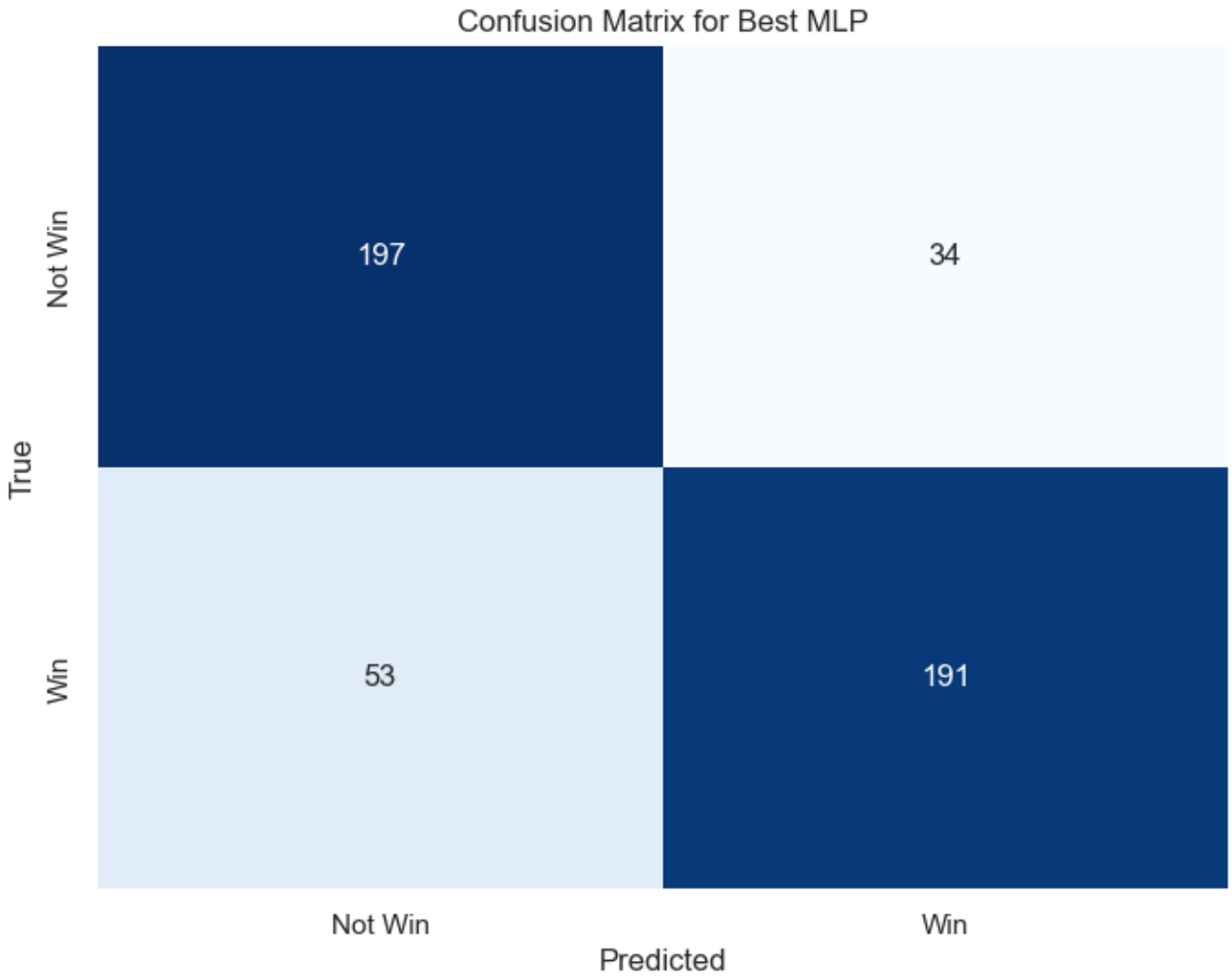


Figure: Confusion matrix for GS-optimized MLP

Results Conclusions

- Model performance was similar across different classifiers.
- Grid search and hyperparameter tuning led to only marginal improvements.
- The most significant performance gain came from addressing class imbalance using SMOTE.
- This highlights that data quality and balance are more critical than model complexity.
- With low-quality or poorly structured data, model performance quickly reaches a ceiling - better data is the only way to raise that limit.

Results Conclusions

	Model	Accuracy	Recall	F1 Score	ROC AUC
0	Random Forest	0.743363	0.603865	0.632911	0.713944
1	AdaBoost	0.745133	0.623188	0.641791	0.719415
2	SVM	0.725664	0.570048	0.603581	0.692845
3	XGBoost	0.736283	0.599034	0.624685	0.707338
4	MLP	0.688496	0.550725	0.564356	0.659441

	Model	Accuracy	Recall	F1 Score	ROC AUC
0	Best Random Forest	0.748673	0.594203	0.634021	0.716096
1	Best AdaBoost	0.753982	0.584541	0.635171	0.718248
2	Best XGBoost	0.759292	0.584541	0.640212	0.722438
3	Best SVM	0.753982	0.589372	0.637076	0.719267
4	Best MLP	0.686726	0.545894	0.560794	0.657025

(a) No SMOTE

	Model	Accuracy	Recall	F1 Score	ROC AUC
0	Random Forest	0.802105	0.803279	0.806584	0.802072
1	AdaBoost	0.812632	0.815574	0.817248	0.812549
2	SVM	0.789474	0.778689	0.791667	0.789777
3	XGBoost	0.783158	0.766393	0.784067	0.783630
4	MLP	0.816842	0.770492	0.812095	0.818146

	Model	Accuracy	Recall	F1 Score	ROC AUC
0	Best Random Forest	0.787368	0.778689	0.790021	0.787613
1	Best AdaBoost	0.797895	0.827869	0.808000	0.797051
2	Best SVM	0.789474	0.778689	0.791667	0.789777
3	Best XGBoost	0.802105	0.795082	0.804979	0.802303
4	Best MLP	0.816842	0.782787	0.814499	0.817800

(b) With SMOTE