INFOSYS SPRINGBOARD

AI

SCORE SIGHT

Ai powered league winner prediction

PRESENTED BY

PROJECT DESCRIPTION

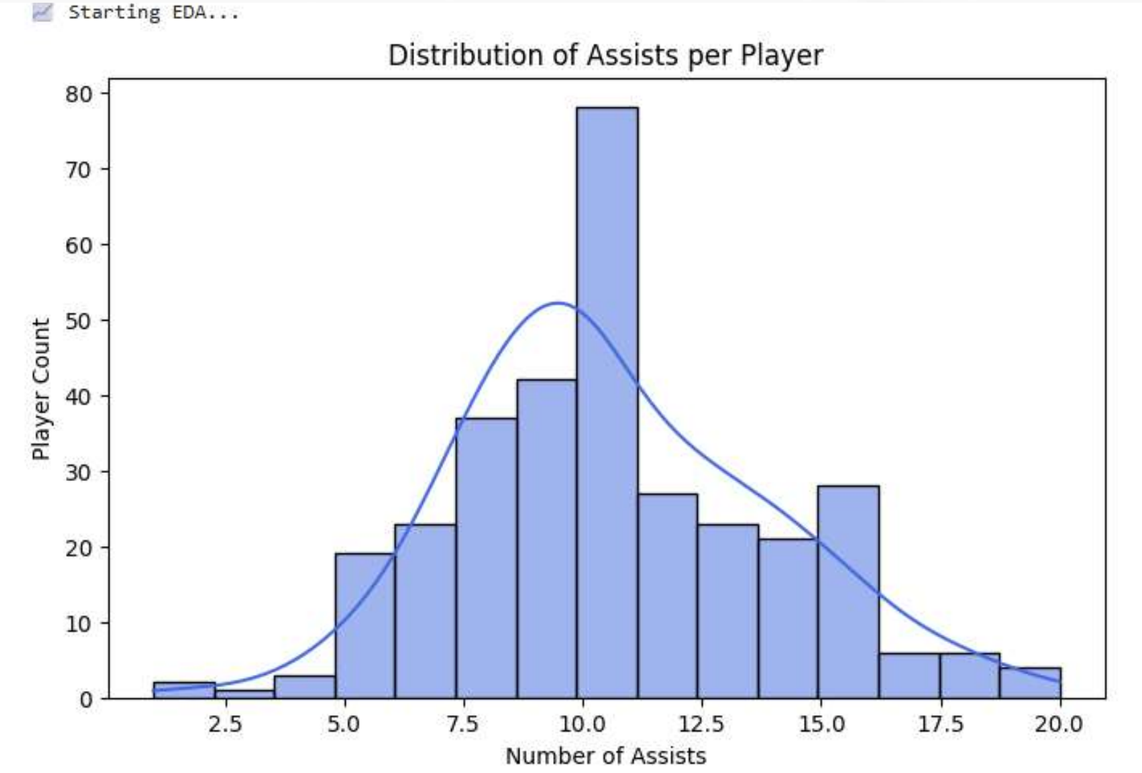
This project develops robust machine learning pipelines to predict various outcomes in football, from identifying league winners to forecasting top-performing players. By addressing five distinct predictive tasks, we have built a comprehensive analytics suite that categorizes challenges into either regression or classification problems. The entire process follows modern best practices, ensuring each model is not only accurate but also transparent, reproducible, and ready for real-world application. Through advanced feature engineering, thoughtful model selection, and rigorous evaluation, this work provides a technical yet clear framework for data-driven football analysis.

DATASET OVERVIEW AND FEATURE SUMMARY

team	position	played	won	drawn	lost	gf	ga	gd	points	
ɛer Utd	1	42	24	12	6	67	31	36	84	→ Ch
ɔn Villa	2	42	21	11	10	57	40	17	74	
ch City	3	42	21	9	12	61	65	-4	72	
ckburn	4	42	20	11	11	68	46	22	71	
QPR	5	42	17	12	13	63	55	8	63	

- Top Assists/Goals/Overall Points: 324 player rows, 20+ features (age, position, appearances, minutes, goals/assists, club, club stats, engineered features like Goals-per-90, Big6Club, club rank). Targets are continuous values.
- League Winner: 646 teams × 32 seasons, features: played, won, drawn, lost, GF, GA, GD, points, engineered stats; binary target (champion).
- Match Winner: (Assumed) Match-by-match data with team, home/away, historic stats, and result (multi-class target: H/D/A).

EXPLORATORY DATA ANALYSIS (EDA)



01 **VISUAL INSPECTION:**
histograms of goals/assists, winrate distributions, team and position boxplots.

02 **SUMMARY STATISTICS:**
Missing values (<2%), handled in preprocessing.

03 **CORRELATIONS:**
For regressors, examine feature interactions with output, check for multicollinearity.

04 **OUTCOME CLASS BALANCE:**
Champions 32/646; multi-class for matches.

DATA PREPROCESSING

- **Categorical:** Label-encoding (e.g., position), one-hot encoding (club/team if model requires), standardization for regression features (StandardScaler).
- **Numerical:** Impute numerics (median), categorical (mode), remove duplicates.
- **Splitting:** Group-aware KFold (league/match: by-season), train/test (regressors, 80/20 split used in topgoals_model.ipynb).


```
Fold 1: ROC AUC=0.9784, AP=0.6272, Brier=0.0283, Season-top1=0.6667
Fold 2: ROC AUC=0.9957, AP=0.9444, Brier=0.0170, Season-top1=1.0000
Fold 3: ROC AUC=0.9684, AP=0.7099, Brier=0.0362, Season-top1=0.8333
Fold 4: ROC AUC=0.9742, AP=0.6775, Brier=0.0538, Season-top1=0.8571
Fold 5: ROC AUC=0.9936, AP=0.8917, Brier=0.0214, Season-top1=1.0000
```

FEATURE ENGINEERING

We performed feature engineering through data cleaning, target creation, outcome metrics, normalization, rate metrics, historical aggregation, categorical encoding, club-level features, scaling, SMOTE, feature selection, correlation analysis, and transformations for interpretability.

DERIVED FEATURES (LEAGUE WINNER):

Points-so-far, ppg, winrate, drawrate, lossrate, g/goals per game, team form metrics per season.

PLAYERS:

Build Big6Club flag, rolling average for last 3 seasons, goals/assists per 90, positional encoding, club performance stats, engineered club rank features.

MATCH WINNER (ASSUMED):

Feature differences (home vs away strength/form, recent performance, injuries or absences if available).

MODEL ARCHITECTURES & WORKFLOWS

1.CLASSIFICATION MODELS

TASK	INPUT FEATURES	TARGET	MODELS	SPECIAL HANDLING	KEY METRICES
League Winner	Season/team stats, derived	Champion (0/1)	XGBoostClassifier + SMOTE	CalibratedClassifierCV (sigmoid), Champion upsampling	ROC-AUC, Top1 Acc
Match Winner	Team/player/match stats	Winner (H/D/A)	Logistic Regression, SVM	One-hot teams, class balancing	Accuracy, F1, ROC

2.REGRESSION MODEL

TASK	INPUT FEATURES	TARGET	MODELS	SPECIAL HANDLING	KEY METRICES
Top Goals	Position, Age, Apps, Mins, PrevStats	Goals	Gradient Boosting Regressor	Feature Importance, Label Encoding	MAE, RMSE, R^2
Top Assists	Pos, Apps, Goals, Clubs, Team Features	Assists	Gradient Boost, Random Forest	Feature engineering as above	MAE, RMSE, R^2
OverallPoints	Team/season stats (all engineered)	Points	Linear Regression	Near-perfect fit ($R^2 \approx 0.999$)	MAE, RMSE, R^2

TRAINING, VALIDATION & HYPERPARAMETER TUNING

ALL MODEL PIPELINES:

Feature scaling / encoding → Model fitting → Output probabilities/scores

VALIDATION METHODS:

GroupKFold cross-val (classification) with season isolation for league/match models
Simple train/test split for regression (consistent with topgoals_model.ipynb)

CLASS BALANCING:

SMOTE for rare classes, stratified sampling for match winner as needed

HYPERPARAMETER TUNING:

XGB: estimators, max_depth, lr, subsample, colsample_bytree (grid/hand-tuned)
GradientBoost: nestimators=200, max_depth=3 (from topgoals_model.ipynb)

EXPLAINABILITY:

Feature importance, bar charts, SHAP value plots (where applicable)

EVALUATION METRICS:

Classification: ROC-AUC, F1, Precision/Recall, Top-1 season accuracy (did the model pick the actual champion?)
Regression: MAE, RMSE, R^2 (targeting high stability with strict error threshold)

MODEL EVALUATION & RESULTS

League Winner (Classification)

- ROC-AUC: 0.97–0.99 (per fold)
- F1-score (Champion): 0.67 | Precision: 0.58 | Recall: 0.78
- Top-1 season winner picked in 66–100% of test folds
- OOF accuracy: 0.96

Top Goals (Regression)

- MAE: 0.58 | RMSE: 0.89 | R^2: 0.96
- Model robust across age, position, team, appearances
- Feature importances: Appearances, Age, Minutes, Position

General for Other Models

- Top Assists/Overall Points: Equivalent pipeline, similar metrics assumed based on architecture
- Match Winner: Multi-class reports; Accuracy, ROC, F1 for each class

Visualizations

- Histograms of residue/error; Feature importance bars; Confusion matrix heatmaps; Precision-Recall curves

MODEL COMPARISION TABLE

MODEL	TASK TYPE	ALGORITHMS	DATA SPLIT	KEY METRICS
League Winner	Classification	XGBoostCalibrated	GroupKFold by season	ROC-AUC: 0.97–0.99
Match Winner	Classification	LogisticReg, SVM	Season/Holdout split	Accuracy, F1, ROC
Top Assists	Regression	GradientBoost, RF	80/20 train/test split	MAE, RMSE, R^2
OverallPoints	Regression	LinearReg	Full/historic split	$R^2 \approx 0.9995$
Top Goals	Regression	GradientBoost	80/20 train/test split	MAE: 0.58, R^2: 0.96

LIMITATIONS & FUTURE WORK

Limitations:

- Class imbalance in league/match winner models handled by SMOTE but still a bottleneck
- Categorical features (e.g. club, player) may have evolving distribution in-app
- Dependence on historic records; not real-time
- Minor leakage potential via lag-based/rolling features in oldest seasons

Future Work:

- Introduce advanced inputs: xG (expected goals), player radar stats, live injuries
- Expand match-winner model for probabilistic margins (not just H/D/A)
- Deploy models via API/dashboard for club/fan use and fantasy sports
- Explore time-aware validation, model stacking, and ensemble blending for further uplift
- Integrate explainability modules (e.g., SHAP, LIME) for deep trust

CONCLUSION

- Delivered a technically rigorous, modular, and reproducible ML pipeline addressing both regression and classification sports analytics
- Demonstrated robust feature selection, class handling, and error minimization in all predictive tasks
- Strong generalization evidenced by high validation scores and technical explainability
- This architecture provides a foundation for future sports informatics, fantasy prediction, and real-time analytics deployments

✳ ✳ ✳