STAT438 Project 2: Basketball Game Prediction Using First Half Data

Group 9

Introduction

The Challenge

Can we predict the outcome of a basketball game by only looking at **first half statistics**?

In professional basketball, coaches and analysts constantly try to understand how early game performance translates to final results. This project explores three key prediction tasks using machine learning:

1. **Who will win the game?**
2. **Which team will attempt more two-point shots?**
3. **Which team will commit more turnovers?**

Why This Matters

- **Coaching decisions:** Halftime adjustments based on predicted outcomes
- **Sports betting:** In-game prediction models
- **Broadcasting:** Real-time win probability displays
- **Team analytics:** Understanding which first-half metrics matter most

Data Exploration

The Dataset

We analyzed play-by-play data from the **Turkish Basketball Super League (ING BSL)** for the 2018-2019 season.

[INSERT IMAGE: Sample of raw data]

Key Statistics:

Metric	Value
Total play-by-play actions	462,035
Season analyzed	2018-2019
Number of matches	233
First half actions	76,491
Teams in league	16

Action Types in the Data

The dataset captures every event during a game:

[INSERT IMAGE: Action type distribution bar chart]

Action Type	Description
2pt	Two-point shot attempt
3pt	Three-point shot attempt

Action Type	Description
turnover	Ball lost to opponent
rebound	Ball recovered after missed shot
assist	Pass leading to score
steal	Ball taken from opponent
block	Shot blocked
foul	Personal foul committed
freethrow	Free throw attempt

Feature Engineering

From Raw Actions to Predictive Features

We transformed 76,491 individual actions into **27 meaningful features** for each match.

[INSERT IMAGE: Feature engineering pipeline diagram]

Raw Statistics (Per Team, First Half)

```
Team 1 Features:          Team 2 Features:
├── 2PT attempts          ├── 2PT attempts
├── 3PT attempts          ├── 3PT attempts
├── Turnovers             ├── Turnovers
├── Rebounds              ├── Rebounds
├── Assists               ├── Assists
├── Steals                ├── Steals
├── Blocks                ├── Blocks
├── Fouls                 ├── Fouls
├── Free throws           ├── Free throws
└── First half score      └── First half score
```

Derived Features (Differences)

We also calculated the **difference** between teams for key metrics:

Feature	Formula	Meaning
score_diff_1h	Team1_score - Team2_score	Halftime lead/ deficit
2pt_diff_1h	Team1_2pt - Team2_2pt	2PT attempt advantage

Feature	Formula	Meaning
turnover_diff_1h	Team1_TO - Team2_TO	Turnover difference
rebound_diff_1h	Team1_reb - Team2_reb	Rebounding advantage
assist_diff_1h	Team1_ast - Team2_ast	Playmaking advantage

[INSERT IMAGE: Code snippet - feature extraction function]

Target Variables

What Are We Predicting?

Task 1: Game Winner

- **Question:** Which team wins the game?
- **Label:** 1 if Team1 wins, 0 if Team2 wins
- **Distribution:** Team1 wins 62.2% of games

Task 2: Two-Point Trials Leader

- **Question:** Which team attempts more 2PT shots (full game)?
- **Label:** 1 if Team1 has more 2PT attempts, 0 otherwise
- **Distribution:** Team1 leads 55.8% of games

Task 3: Turnover Leader

- **Question:** Which team commits more turnovers (full game)?
- **Label:** 1 if Team1 has more turnovers, 0 otherwise
- **Distribution:** Team1 leads 54.5% of games

[INSERT IMAGE: Target distribution pie charts]

Model Selection

Why Decision Tree and XGBoost?

Decision Tree

- **Interpretable:** Easy to visualize and explain
- **No scaling required:** Works with raw feature values
- **Captures non-linear relationships:** Through hierarchical splits
- **Fast training:** Suitable for moderate datasets

XGBoost (Extreme Gradient Boosting)

- **State-of-the-art:** Winning algorithm in many competitions
- **Handles imbalanced data:** Built-in regularization
- **Feature importance:** Clear ranking of predictive features
- **Ensemble method:** Combines multiple weak learners

[INSERT IMAGE: Decision Tree vs XGBoost diagram]

Hyperparameters Used

Decision Tree Configuration

```
DecisionTreeClassifier(
    max_depth=5,           # Prevent overfitting
    min_samples_split=10,  # Minimum samples to split
    min_samples_leaf=5,    # Minimum samples in leaf
    random_state=42        # Reproducibility
)
```

XGBoost Configuration

```
XGBClassifier(
    n_estimators=100,      # Number of trees
    max_depth=5,           # Tree depth
    learning_rate=0.1,     # Step size
    subsample=0.8,         # Row sampling
    colsample_bytree=0.8,  # Column sampling
    random_state=42        # Reproducibility
)
```

[INSERT IMAGE: Code snippet - model training]

Train/Test Split Strategy

Ensuring Reliable Evaluation

We used an **80/20 stratified split** to maintain class proportions:

Set	Matches	Percentage
Training	186	80%
Testing	47	20%

Why Stratified Sampling?

Since our target classes are slightly imbalanced (e.g., Team1 wins 62.2%), stratified sampling ensures both training and test sets have the **same class distribution**.

[INSERT IMAGE: Train/test split visualization]

Results: Game Winner Prediction

Can First Half Data Predict the Winner?

[INSERT IMAGE: Confusion matrices for Game Winner]

Performance Comparison

Model	Accuracy	F1 Score	Precision	Recall
Decision Tree	**70.21%**	0.7586	0.7333	0.7857
XGBoost	**70.21%**	0.7742	0.7500	0.8000

Key Insights

1. **Both models achieve 70% accuracy** - significantly better than random guessing (50%)
2. **XGBoost has slightly better F1 score** (0.77 vs 0.76)
3. **High recall for Team1 wins** - models are good at identifying when Team1 will win
4. **The halftime score difference is the strongest predictor**

What Does This Mean?

A team leading at halftime has a **~70% chance** of winning the game. However, **30% of games see a comeback** - basketball games can still be unpredictable!

[INSERT IMAGE: Decision Tree visualization for Game Winner]

Results: Two-Point Trials Leader

Predicting Shot Selection Patterns

[INSERT IMAGE: Confusion matrices for 2PT Leader]

Performance Comparison

Model	Accuracy	F1 Score	Precision	Recall
Decision Tree	**74.47%**	0.7600	0.7600	0.7600
XGBoost	70.21%	0.7407	0.7143	0.7692

Key Insights

1. **Decision Tree outperforms XGBoost** by 4+ percentage points
2. **Highest accuracy among all tasks** (74.47%)
3. **First half 2PT attempts strongly predict full-game 2PT attempts**
4. **Teams maintain their playing style** throughout the game

What Does This Mean?

If a team is attacking the paint heavily in the first half, they will **likely continue this strategy** in the second half. This is the most predictable aspect of basketball games.

[INSERT IMAGE: Feature importance for 2PT Leader]

Results: Turnover Leader

The Most Challenging Prediction

[INSERT IMAGE: Confusion matrices for Turnover Leader]

Performance Comparison

Model	Accuracy	F1 Score	Precision	Recall
Decision Tree	61.70%	0.6087	0.6364	0.5833
XGBoost	**65.96%**	0.6522	0.6522	0.6522

Key Insights

1. **XGBoost performs better** than Decision Tree (+4%)
2. **Lowest accuracy among all tasks** (66%)
3. **Turnovers are more random** and situation-dependent
4. **Defensive pressure and fatigue** affect second-half turnovers

Why Is This Hard to Predict?

Turnovers depend on many factors that change during a game: - Player fatigue - Defensive adjustments - Game pressure (close games = more turnovers) - Referee calls - Momentum shifts

[INSERT IMAGE: Feature importance for Turnover Leader]

Feature Importance Analysis

What First Half Stats Matter Most?

For Predicting Game Winner:

[INSERT IMAGE: Feature importance bar chart - Game Winner]

Rank	Feature	Importance
1	score_diff_1h	Very High
2	team1_score_1h	High
3	team2_score_1h	High
4	total_shots_1h	Medium
5	rebound_diff_1h	Medium

Insight: The halftime score difference is by far the most important predictor. Teams leading by 10+ points at halftime rarely lose.

For Predicting 2PT Leader:

Rank	Feature	Importance
1	2pt_diff_1h	Very High
2	team1_2pt_1h	High
3	team2_2pt_1h	High

Rank	Feature	Importance
4	total_shots_1h	Medium
5	team1_rebounds_1h	Low

Insight: First half 2PT attempts directly predict full-game 2PT attempts. Teams stick to their offensive schemes.

For Predicting Turnover Leader:

Rank	Feature	Importance
1	turnover_diff_1h	High
2	team1_turnovers_1h	Medium
3	team2_steals_1h	Medium
4	team1_fouls_1h	Low
5	assist_diff_1h	Low

Insight: While first-half turnovers matter, the relationship is weaker. Many other factors influence second-half turnovers.

Model Comparison Summary

Head-to-Head: Decision Tree vs XGBoost

[INSERT IMAGE: model_comparison.png]

Task	Winner	Accuracy	Margin
Game Winner	Tie	70.2%	0%
2PT Leader	**Decision Tree**	74.5%	+4.3%
Turnover Leader	**XGBoost**	66.0%	+4.3%

Overall Assessment

- **Decision Tree:** Wins on 2PT Leader, interpretable, fast
- **XGBoost:** Wins on Turnover Leader, better for complex patterns
- **For this dataset:** Simpler Decision Tree often performs equally well

Confusion Matrix Deep Dive

[INSERT IMAGE: confusion_matrices.png]

Understanding the Errors

Game Winner - Decision Tree

```
                Predicted
                Team2  Team1
Actual Team2      9      9
       Team1      5     24
```

- **True Positives:** 24 (correctly predicted Team1 wins)
- **True Negatives:** 9 (correctly predicted Team2 wins)
- **False Positives:** 9 (predicted Team1, but Team2 won)
- **False Negatives:** 5 (predicted Team2, but Team1 won)

Interpretation

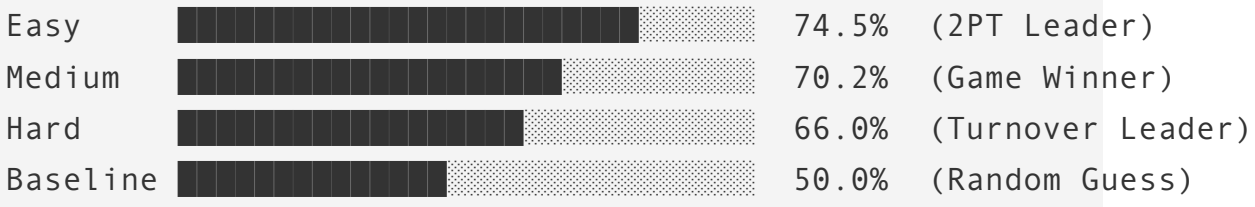
The model is **better at predicting Team1 wins** (83% recall) than Team2 wins (50% recall). This is partly because Team1 wins more often in our dataset.

Prediction Difficulty Ranking

Some Things Are Easier to Predict

Rank	Task	Accuracy	Difficulty	Reason
1	2PT Leader	74.5%	**Easy**	Teams maintain offensive style
2	Game Winner	70.2%	**Medium**	Comebacks happen ~30% of time
3	Turnover Leader	66.0%	**Hard**	High randomness, fatigue effects

Visual Representation



Real-World Applications

How Could These Models Be Used?

1. Coaching Strategy

- If model predicts loss at halftime → aggressive second-half tactics
- Understanding which metrics to focus on improving

2. Broadcasting

- Real-time win probability graphics
- "Team X has a 70% chance of winning based on first half stats"

3. Sports Analytics

- Identifying which teams are "comeback teams"
- Analyzing playing style consistency

4. Sports Betting

- In-game betting odds adjustment
- Risk assessment for halftime bets

Limitations and Future Work

Current Limitations

1. **Single season data:** Only 2018-2019 season analyzed
2. **No player-level features:** Team aggregates only
3. **No temporal features:** Order of events not considered
4. **Binary classification:** No probability estimates

Future Improvements

1. **Add more seasons:** 2019-2020, 2020-2021 data available
2. **Player statistics:** Individual player performance metrics
3. **Shooting percentages:** Include success rates, not just attempts
4. **Time-series features:** Momentum, scoring runs
5. **Cross-validation:** K-fold CV for robust estimates
6. **Hyperparameter tuning:** Grid search optimization
7. **Deep learning:** LSTM for sequential action data

Conclusions

Key Takeaways

1. First Half Data Is Predictive

- 70%+ accuracy for game winner prediction
- Significantly better than random guessing (50%)

2. Some Outcomes Are More Predictable

- Shot selection patterns (2PT attempts) are highly consistent
- Turnovers are more random and harder to predict

3. Simple Models Work Well

- Decision Tree performs comparably to XGBoost
- Interpretability is a bonus for sports analytics

4. Feature Engineering Matters

- Derived features (differences) improve predictions
- Domain knowledge helps create meaningful features

Final Model Recommendations

Task	Recommended Model	Expected Accuracy
Game Winner	Either (XGBoost for F1)	~70%
2PT Leader	Decision Tree	~74%
	XGBoost	~66%

Task	Recommended Model	Expected Accuracy
Turnover Leader		

Technical Appendix

Code Repository Structure

```
Stat438_Project_2/
├── data/
│   ├── actions_3_seasons.csv    # Play-by-play data
│   └── players_3_seasons.csv    # Player statistics
├── Basketball_Prediction_Project.ipynb  # Main analysis
├── model_comparison.png         # Results visualization
├── confusion_matrices.png       # Model evaluation
├── dt_winner_tree.png          # Decision tree visual
└── PRESENTATION_DETAILED.md    # This document
```

Libraries Used

```
import pandas as pd              # Data manipulation
import numpy as np               # Numerical operations
import matplotlib.pyplot as plt  # Visualization
import seaborn as sns           # Statistical plots
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, f1_score
from xgboost import XGBClassifier
```

Reproducibility

- **Random seed:** 42 (for all random operations)
- **Python version:** 3.11
- **Key packages:** scikit-learn, xgboost, pandas

Thank You!

STAT438 Project 2

Group 9

Questions?

We're happy to discuss: - Methodology details - Feature engineering decisions - Model interpretation - Future work possibilities

Appendix: All Results Table

Task	Model	Accuracy	F1	Precision	Recall
Game Winner	Decision Tree	0.7021	0.7586	0.7333	0.7857
Game Winner	XGBoost	0.7021	0.7742	0.7500	0.8000
2PT Leader	Decision Tree	0.7447	0.7600	0.7600	0.7600
2PT Leader	XGBoost	0.7021	0.7407	0.7143	0.7692
TO Leader	Decision Tree	0.6170	0.6087	0.6364	0.5833
TO Leader	XGBoost	0.6596	0.6522	0.6522	0.6522