Data-Driven NBA Game Prediction Engine

Group Details: Viraj Ajaykumar – fv115

Project Definition

The primary objective of this project was to determine if a machine learning model could effectively predict the outcomes of National Basketball Association (NBA) games by leveraging historical betting market inefficiencies. The core problem we are solving is the "quantification of edge" in sports betting; specifically, we are investigating whether pre-game Vegas odds, such as Moneyline and Spread, contain sufficient signal to outperform random guessing and simple heuristics in a classification task. Strategically, this project involves the architectural design of a full-stack data pipeline, moving from raw data ingestion to structured storage and finally to predictive analysis. This relates directly to the course themes of Data Management in Data Science, specifically applying the concepts of Database Normalization to reduce redundancy, ETL (Extract, Transform, Load) processes to ensure data quality, and the integration of structured SQL queries with Python-based analytical tools. We aimed to ingest seventeen years of NBA history (2008–2025) into a relational database, testing the hypothesis that a properly engineered data pipeline can expose predictive patterns that are invisible to manual analysis.

Introduction

The "Moneyball" revolution has fundamentally transformed how sports teams operate, shifting decision-making from intuition to empirical evidence. However, this data revolution also presents a significant challenge in data management: handling the volume, variety, and velocity of sports data. The novelty of our approach lies in the system architecture. While many amateur

sports analytics projects rely on static, flat-file CSV analysis, our project engineered a persistent Relational Database Management System (RDBMS) to normalize and store game and player data separately. This mimics real-world enterprise environments where data integrity, consistency, and schema normalization are paramount. This project is important because it addresses existing issues in current data management practices, such as the "update anomalies" and data duplication common in spreadsheet-based tracking. By enforcing a relational schema, we ensure that player attributes are stored once and referenced dynamically, solving the issue of scalability that plagues simpler analysis methods.

Methodology

We implemented the project in three distinct components utilizing Python, SQLite, and Scikit-Learn to create a cohesive data pipeline. First, we moved beyond flat-file storage by implementing a Relational Database Management System using SQLite. We designed a normalized schema consisting of two primary tables to reduce data redundancy. The games table acts as our Fact Table, storing transactional data for every match, including dates, scores, and betting metrics. The players table acts as a Dimension Table, storing static attribute data such as height, weight, and experience. This relational structure ensures data integrity and allows for complex SQL queries to join player attributes with game results, representing a significant improvement over single-spreadsheet analysis.

For the data science component, we developed an automated ETL pipeline using the Pandas library to manage the data lifecycle. In the extraction phase, raw data was ingested from multiple CSV sources, including Kaggle datasets containing nearly two decades of game logs. The transformation phase involved critical data cleaning, specifically the removal of records with

missing betting odds to ensure model stability, and feature engineering, where we calculated a new binary target variable called home_win derived from the raw game scores. Finally, the processed data was persisted into the SQLite database, establishing a "single source of truth" for our downstream analysis. We also employed four distinct visualization techniques—histograms, scatter plots, time-series line plots, and heatmaps—to perform Exploratory Data Analysis and validate the quality of our loaded data.

For the predictive modeling phase, we utilized the Scikit-Learn library. We queried the SQL database to retrieve a training set consisting of games from 2015 to the present. We specifically filtered for these later seasons to account for the "three-point revolution" in the NBA, ensuring our model was training on relevant modern playstyles. We selected home_moneyline, away_moneyline, and spread as our feature vector. We selected a Random Forest Classifier with 100 estimators as our model architecture due to its ability to handle non-linear relationships and its resistance to overfitting compared to linear models. We employed an 80/20 train-test split validation strategy to ensure that our reported accuracy reflected the model's performance on unseen future games rather than memorized historical data.

Results and Contributions

Our analysis yielded several key insights into the nature of NBA competition. The distribution of victory margins, visualized via histogram, followed a normal distribution centered near zero. This finding verified our hypothesis that the vast majority of NBA games are competitively balanced, making prediction difficult. Our scatter plot analysis of the Vegas spread versus the actual margin revealed a strong positive linear correlation, confirming that betting markets are generally efficient. However, the variance around the trend line highlighted the

"upset potential" that our model successfully captured. Additionally, our time-series analysis tracked the average total points per game from 2008 to 2025, revealing a steep upward trend starting around 2015. This effectively visualized the league-wide offensive revolution, validating our decision to filter the training data to recent seasons.

Ultimately, the Random Forest model achieved a predictive accuracy of 66.17% on the test set. This result refutes the null hypothesis that betting odds are purely efficient and cannot be exploited; an accuracy significantly above 50% (random guess) and above the typical home-court advantage baseline (~58%) indicates our method works. The primary advantage of our approach is its modularity; the database can be updated daily without breaking the model. However, a limitation is that our model currently relies solely on pre-game odds and does not account for real-time factors such as player injuries or trades, which could be integrated in future iterations.

Changes After Proposal

Our final report differs slightly from our initial proposal in terms of the database technology used. We initially proposed using PostgreSQL for the database component. However, during the implementation phase, we encountered bottlenecks regarding local server configuration and portability for submission. We pivoted to using SQLite, which is serverless and file-based. This change allowed for faster iteration and easier integration with Python while still strictly adhering to the SQL standards and relational schema design requirements of the course. Additionally, we initially planned to include individual player statistics (like Points Per Game) in the predictive model. We found that this introduced significant noise and

dimensionality issues, so we simplified the feature set to focus on market-driven features (Odds and Spreads), which resulted in higher model stability and accuracy.

Individual Contributions

Viraj Ajaykumar was responsible for the end-to-end Python code implementation. I designed the SQLite database schema and wrote the SQL DDL commands to create the games and players tables. I implemented the ETL pipeline to clean the raw CSV data and engineered the home_win feature. Additionally, I developed the Machine Learning workflow using Scikit-Learn, generated the four final visualizations using Matplotlib and Seaborn, and authored this final project report.

Work Cited

Pedregosa et al., "Scikit-learn: Machine Learning in Python," *JMLR 12*, pp. 2825-2830, 2011.

The Pandas Development Team, "pandas-dev/pandas: Pandas," *Zenodo*, 2020.

Hippolyte, R., "SQLAlchemy Documentation," *SQLAlchemy*, 2024.

Kaggle Dataset, "NBA Games Data with Betting Odds (2008-2025)," accessed 2025.