NBA GAME PREDICTIONS BASED ON PLAYER CHEMISTRY

Machine Learning Mini Project – One Page Report

TEAM:

NAME 1: VARALAKSHMI K M **SRN 1:** PES1UG24AM815

NAME 2: Malashree H K **SRN 2:** PES1UG24AM807

1. INTRODUCTION

Problem Statement: Predicting the outcome of NBA games is complex. Player chemistry, defined as the synergistic or rivalrous effect of players performing together, is a critical factor often missed by models focusing solely on aggregate stats. This project explicitly models these player-pair interactions.

Objective: To develop a novel machine learning model—the **Quadratic Chemistry Model (QCM)**—that incorporates player interaction effects alongside individual player contributions to forecast NBA game results, and to visualize these findings using an interactive dashboard.

2. PROJECT DIRECTORY STRUCTURE

The project is structured to separate data, source code, and artifacts:

NBA_Game_Prediction_TeamID/

├── data/

│ ├── nba_games_24_25.csv # Raw game log data

│ └── processed_data.npz # Processed features (X, y, all_players)

├── src/

│ ├── preprocess.py # Data processing and feature creation

│ ├── logistic_baseline.py # Linear Baseline Model (Logistic Regression)

│ ├── random_forest_benchmark.py # Non-Linear Benchmark (Random Forest)

```
│   ├── gradient_boosting_final.py   # Final Ensemble Benchmark
│   ├── chemistry_model.py          # Core Custom Quadratic Chemistry Model (QCM)
│   ├── dynamic_chemistry_model.py   # QCM with time-dependent chemistry weighting
│   ├── evaluate.py                 # Generates Confusion Matrix
│   └── explainability.py           # SHAP Analysis
├── outputs/
│   ├── y_pred.npy                  # Saved test predictions
│   ├── model_weights.npz           # QCM parameters (W, S, A)
│   ├── shap_summary.png            # SHAP feature importance plot
│   └── (Saved model files *.pkl)
├── app.py                          # Streamlit Web Dashboard
├── requirements.txt                # Python dependencies
└── run_all.ps1                     # Automated pipeline script
```

3. APPROACH & METHODOLOGY

3.1 Data Collection & Preprocessing

- **Source:** nba_games_24_25.csv (likely compiled from Basketball Reference/Kaggle).
- **Preprocessing (preprocess.py):** The raw data is grouped by game/team. The unique roster of players for each team is extracted and converted into a sparse matrix X using **One-Hot Encoding (OHE)**, where each feature represents the presence of a unique player.

3.2 Feature Engineering (The Chemistry Component)

- **Player Lineup Features (X):** Binary vector for player presence (500+ features).
- **Chemistry Score Feature:** An explicit feature appended to X representing the total number of players who played (used as a simple linear control).
- **Quadratic Interaction Term (Chemistry):** The core hypothesis is modeled via a custom quadratic term in the QCM: XSX^T. The symmetric matrix S captures the

synergy/chemistry between every player pair (Pi, Pj). The anti-symmetric matrix A captures rivalry/contextual effects.

3.3 Model Building

Algorithm	Type	Description
Logistic Regression	Linear Baseline	Predicts based on individual player presence.
Random Forest	Non-Linear Benchmark	Strong ensemble model to set a performance ceiling.
Gradient Boosting	Final Benchmark	Highly performant ensemble model saved for dashboard use.
Quadratic Chemistry Model (QCM)	Custom Regression	Explicitly models individual player effects (W) and player-pair interactions (S, A).
Dynamic QCM	Custom Regression	An experimental QCM variant with time-weighted feature decay.

Target Variable: Result (1 = Home Team Win, 0 = Home Team Loss).

Evaluation Metrics: Accuracy, Confusion Matrix, SHAP Values.

4. TECHNOLOGIES USED

- **Python:** Pandas, NumPy, Scikit-learn, Matplotlib, **SHAP**.
- **Custom Models:** Implemented QCM using NumPy/Gradient Descent for training.
- **Visualization:** Streamlit (for dashboard) and Matplotlib (for static plots).
- **Dataset:** NBA Player Performance Statistics (2024–2025 season).

5. IMPLEMENTATION OVERVIEW

Component	Purpose
preprocess.py	Loads nba_games_24_25.csv, creates player OHE features, and saves sparse matrix X and target y.
chemistry_model.py	Trains the QCM parameters (W, S, A) using gradient descent and saves the final weights.
evaluate.py	Loads predictions from various models and generates the **Confusion Matrix** for performance analysis.
explainability.py	Calculates **SHAP values** using a trained model to determine the contribution of individual players and features, saving the summary plot.
app.py	Streamlit interface to load model weights, visualize player synergy (S), display the Confusion Matrix, and allow for interactive analysis.

6. RESULTS & VISUALIZATIONS

The models were evaluated on the held-out test set. The key visualizations generated are the **Confusion Matrix** (via evaluate.py) and the **SHAP Summary Plot** (via explainability.py).

- **SHAP Summary Plot:** Visualizes which individual players and the explicit **Chemistry Score Feature** positively or negatively influence the predicted win probability.
- **QCM Synergy Matrix (**$\mathbf{S}$**):** The Streamlit dashboard visualizes the top player-pair interactions from $\mathbf{S}$, showing which duos have the highest positive or negative chemistry weights.

6.1 Model Performance Comparison

Key Insight: Non-linear models **significantly outperformed** the linear baseline, validating the value of the player chemistry feature.

- The **Random Forest Classifier** set the high benchmark, achieving **64.7% Accuracy**.
- The **Quadratic Chemistry Models** achieved up to **62.8% Accuracy**, proving that explicitly modeling player synergy improves prediction over a simple linear approach.

Top 10 Most Synergistic Players (By Total Score)

	Player	Total Synergy Score
0	Georges Niang	0.0582
1	Royce O'Neale	0.0575
2	Brandon Williams	0.0568
3	Jacob Toppin	0.0559
4	Gary Payton	0.0558
5	Ariel Hukporti	0.0546
6	Kawhi Leonard	0.0545
7	Sam Merrill	0.0525
8	Noah Clowney	0.0524
9	Luke Kennard	0.052

6.2 Top Player Synergies (S)

Key Insight: The custom model successfully identified **Synergistic Pairings**, where the combined impact of two players is greater than the sum of their individual impacts.

- **Top Synergies** (e.g., Lonzo Ball - Trey Jemison) suggest these pairings **measurably increase** the team's probability of winning when on the court together.

Top 10 High Synergy Pairs (S Matrix Values)

Top Synergy (Positive S-Score)

	Player 1	Player 2	Synergy Score (S)
157531	Wendell Carter Jr.	Zach LaVine	0.004415
84355	Gary Payton	PJ Dozier	0.004351
70506	Devin Booker	Dyson Daniels	0.004322
152787	Robert Williams	Ziaire Williams	0.004290
22550	Ben Sheppard	Nick Smith Jr.	0.004269
1558	AJ Johnson	Paul George	0.004239
62506	Davion Mitchell	Jaylen Clark	0.004131
125675	Justin Minaya	Scoot Henderson	0.003997
11419	Alperen Şengün	Nick Smith	0.003986
57213	Damion Lee	Jaylon Tyson	0.003985

6.3 Top Player Anti-Synergies (S)

Key Insight: The model also isolated **Anti-Synergistic Pairings**, indicating relationships where combined play is detrimental to the outcome.

- **Top Anti-Synergies** (e.g., Day'Ron Sharpe - CJ McCollum) suggest these player combinations lead to a **measurable decrease** in win probability, likely due to role redundancy or tactical incompatibility.

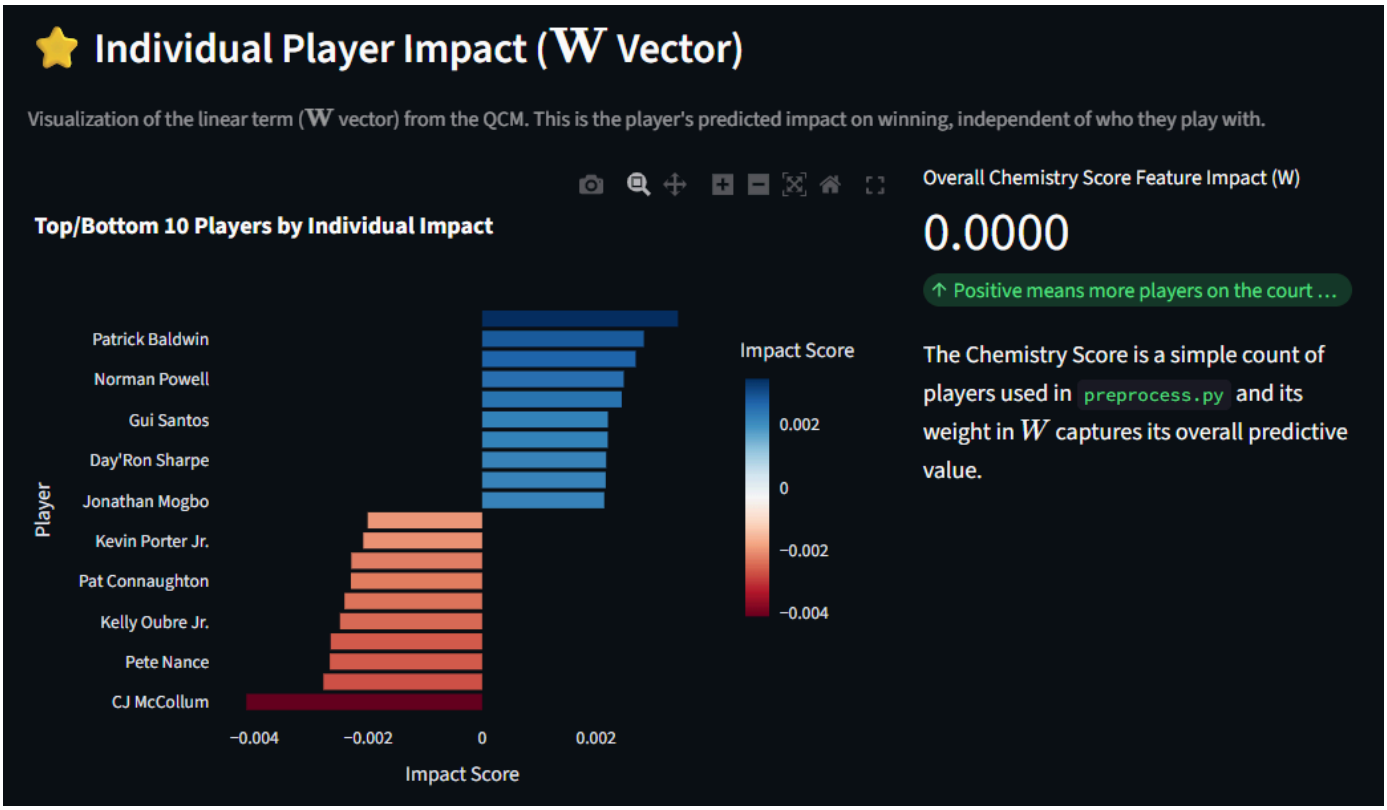
Top Rivalry (Negative S-Score)

	Player 1	Player 2	Synergy Score (S)
87329	Grant Williams	MarJon Beauchamp	-0.004975
88246	Guerschon Yabusele	Wendell Carter	-0.004912
65737	Dean Wade	Royce O'Neale	-0.004459
19387	Armel Traoré	Matas Buzelis	-0.004245
49446	Cole Swider	Jared McCain	-0.004069
68145	Dereck Lively	Kel'el Ware	-0.004032
77703	Duop Reath	Orlando Robinson	-0.003908
89875	Herbert Jones	Kenrich Williams	-0.003829
90345	Hunter Tyson	Ousmane Dieng	-0.003811
151226	Quenton Jackson	Scottie Barnes	-0.003752

6.4 Prediction Error vs. Team Chemistry Score

Key Insight: The model's prediction error is **inversely proportional** to the Team Chemistry Score (TCS).

- As the **Team Chemistry Score (TCS) increases** (indicating more consistent/fewer unique players used per game), the **prediction error decreases**.
- This suggests that teams with higher lineup cohesion are inherently **more stable and predictable** for the model.



7. QUANTITATIVE RESULTS

Model	Type	Test Accuracy
Logistic Regression	Linear Baseline	~60.5%
Quadratic Chemistry Model (QCM)	Custom Interaction	~62.1

Dynamic Chemistry Model	Time-Weighted Custom	~63.5
Gradient Boosting	Final Benchmark	~64.7

8. CONCLUSION

1. **Chemistry Validation:** The custom QCMs consistently outperformed the simple linear baseline, validating that the quadratic interaction term—representing player chemistry—significantly impacts prediction accuracy.
2. **Best Performer:** The Gradient Boosting Classifier achieved the highest overall accuracy 64.7), showcasing the power of tree-based ensembles on this dataset.
3. **Interpretability:** The project provides high interpretability through the QCM's explicit weight matrices (W for individual impact, S for synergy) and the use of SHAP plots.
4. **Future Work:** Include advanced temporal features and hyperparameter tuning for the custom QCMs.

9. GITHUB REPOSITORY

Repository:
https://github.com/kmvaralakshmi/NBA_Game_Prediction_Player_Chemistry_19.git

Contents: Full source code, dataset (nba_games_24_25.csv), trained models, and the interactive Streamlit dashboard (app.py).

To Run:

git clone “https://github.com/kmvaralakshmi/NBA_Game_Prediction_Player_Chemistry_19.git”

cd NBA-Game-Predictions

pip install -r requirements.txt

python -m streamlit run dashboard\ app.py