**Football Tracking Data Analysis: Ball Position Prediction**

**Context and Purpose**

In modern football analysis, accurately predicting ball-related information is crucial for understanding game dynamics. I've developed two complementary approaches to predict ball locations from player tracking data, providing coaches and analysts with valuable tools for strategy formulation, performance evaluation, and tactical adjustments.

**Methodology Overview**

I implemented two distinct methodologies to address the challenge of predicting ball positions using only player tracking data:

1. **Random Forest Regression Model**: A feature-engineered approach that leverages team formations and player movement patterns
2. **Graph Neural Network (GNN)**: A spatial relationship-based approach that models players as nodes in a dynamic graph

Both approaches were trained on matches with known ball coordinates and then used to predict ball positions for match 4, where only player tracking data was available.

**Data Analysis and Feature Engineering**

**Data Understanding**

* The tracking data provided was time-series data with each row representing 1/10 seconds of match play
* Each file contained player coordinates for home and away teams (player\_id\_x, player\_id\_y)
* Two matches had ball position data which served as training data
* Match 4 required ball position predictions

**Feature Engineering - Random Forest Approach**

* **Team Formations**: Calculated team centroids, spreads, and speeds
* **Player Dynamics**: Extracted velocity vectors and acceleration data
* **Spatial Relationships**: Calculated distances between players and proximity metrics
* **Temporal Features**: Incorporated sequence information through windowed features

**Feature Engineering - GNN Approach**

* **Node Features**: Represented each player as a node with position, velocity, and team features
* **Edge Creation**: Connected players based on spatial proximity
* **Graph Structure**: Captured team formations and tactical relationships
* **Message Passing**: Allowed information flow between spatially connected players

**Model Development**

**Random Forest Implementation**

def \_train\_models(self, features, ball\_x, ball\_y):

"""Train machine learning models to predict ball x and y coordinates."""

# Convert to numpy arrays

X = np.array(features)

y\_x = np.array(ball\_x)

y\_y = np.array(ball\_y)

# Scale features

self.scaler = StandardScaler()

X\_scaled = self.scaler.fit\_transform(X)

# Train Random Forest models

self.x\_model = RandomForestRegressor(n\_estimators=100, random\_state=42)

self.x\_model.fit(X\_scaled, y\_x)

self.y\_model = RandomForestRegressor(n\_estimators=100, random\_state=42)

self.y\_model.fit(X\_scaled, y\_y)

**Graph Neural Network Implementation**

class FootballGNN(nn.Module):

def \_\_init\_\_(self, node\_features, hidden\_channels=64):

super(FootballGNN, self).\_\_init\_\_()

# GNN layers

self.conv1 = GCNConv(node\_features, hidden\_channels)

self.conv2 = GCNConv(hidden\_channels, hidden\_channels)

# Output regression layers for predicting ball position

self.fc1 = nn.Linear(hidden\_channels, hidden\_channels)

self.fc2 = nn.Linear(hidden\_channels, 2) # 2 outputs: ball\_x and ball\_y

def forward(self, data):

# Node features

x, edge\_index, batch = data.x, data.edge\_index, data.batch

# Graph convolution

x = self.conv1(x, edge\_index)

x = F.relu(x)

x = F.dropout(x, p=0.1, training=self.training)

x = self.conv2(x, edge\_index)

x = F.relu(x)

# Global pooling

x = global\_mean\_pool(x, batch)

# Fully connected layers

x = self.fc1(x)

x = F.relu(x)

x = F.dropout(x, p=0.1, training=self.training)

# Output ball coordinates

ball\_pos = self.fc2(x)

return ball\_pos

**Challenges and Learnings**

**Challenges Encountered**

1. **Limited Training Data**: Only two matches with ball data were available
2. **Missing Player Positions**: Not all players were visible in every frame
3. **Complex Movement Patterns**: Ball movements follow various physical and tactical patterns
4. **Multiple Prediction Paradigms**: Deciding between physics-based vs. data-driven approaches

**Successful Approaches**

1. **Feature Engineering**: Creating meaningful representations of team formations and player relationships
2. **Graph-Based Modeling**: Capturing spatial relationships to understand tactical contexts
3. **Ensemble Methods**: Combining predictions for more robust results
4. **Temporal Smoothing**: Applying smoothing for more natural ball trajectories

**Unsuccessful Attempts**

1. **Simple Linear Regression**: Too basic for the complex spatial relationships
2. **Pure Physical Modeling**: Lacked ability to capture tactical patterns
3. **Using Raw Player Coordinates**: Required more feature engineering to be effective

**Evaluation and Visualization**

To evaluate prediction quality, I developed a visualization tool that:

1. Creates animations showing player and predicted ball movements
2. Compares predicted trajectories against ground truth (when available)
3. Generates heat maps of prediction errors across the pitch
4. Provides statistical analysis of prediction accuracy

**Key Insights**

1. **Team Formation Impact**: Player distributions strongly indicate likely ball positions
2. **Velocity Information**: Player speed and direction are crucial predictors
3. **Spatial Context Matters**: Relationships between players contain valuable information
4. **Model Complementarity**: RF excels at specific patterns while GNN captures spatial relationships

**Final Results**

The final prediction for match 4 combines the strengths of both approaches, providing a comprehensive prediction of ball positions that accounts for:

1. Player positions and velocities
2. Team formations and tactics
3. Spatial relationships between players
4. Typical ball movement patterns in football

The provided CSV output contains the predicted ball positions for each 1/10 second frame of match 4, which can be used for further analysis or combined with the animation tool for visualization.

**Future Improvements**

1. **More Training Data**: Additional matches would improve prediction accuracy
2. **Player Role Information**: Incorporating player roles could enhance predictions
3. **Physics-Informed Learning**: Combining machine learning with physical constraints
4. **Tactical Context**: Adding game state information like score and time remaining

This methodology demonstrates how machine learning approaches can effectively predict ball positions in football, providing valuable insights for coaches, analysts, and researchers.