DS5760 NoSQL for Modern Data Science Applications

Final Project Proposal

- Roshan Sivakumar

Project Title

European Soccer Analytics Platform with MongoDB

Dataset

Source

Kaggle - European Soccer Database

Link: https://www.kaggle.com/datasets/hugomathien/soccer

Description

This comprehensive dataset contains detailed soccer statistics from 11 European leagues spanning the 2008-2016 seasons. The dataset includes:

- 25,000+ match records with complete match details
- 10,000+ player profiles with FIFA ratings and attributes
- Team statistics and attributes sourced from EA Sports FIFA
- Detailed match events (goals, cards, possession, corners, fouls)
- Player lineups with squad formations and positional data
- Betting odds from multiple providers for predictive analysis

Curation Plan

- Download the SQLite database from Kaggle and convert to MongoDB collections
- Denormalize data by embedding player statistics within match documents for optimized queries
- Create separate MongoDB collections: matches, players, teams, and leagues
- Add indexes on frequently queried fields (date, league, team names, player ratings)
- Clean null or missing values in key fields and ensure data consistency

Database Choice

Selected Database: MongoDB

Justification

MongoDB is the optimal choice for this soccer analytics application for the following reasons:

- **Flexible Schema:** Match events vary significantly (different numbers of goals, cards, substitutions, VAR decisions), making MongoDB's schema-less design ideal for handling this variability

- **Nested Documents:** Perfect for embedding player lineups, match events, and team formations within match documents, reducing the need for complex joins
- **Array Support:** Naturally handles multiple goal scorers, assists, yellow/red cards, and substitutions per match
- **Aggregation Pipeline:** Ideal for computing league tables, player statistics aggregations, team performance metrics, and time-series analysis
- **Rich Query Capabilities:** Supports range queries on dates, ratings, and scores; text search on player and team names; and complex filtering
- **Scalability:** Efficiently handles 25,000+ match records and 10,000+ player profiles with horizontal scaling capabilities

Planned Queries (7+ distinct query types)

1. **Top Goal Scorers by League and Season**
   Aggregate all goals scored from match events, group by player name, filter by specific league and season. Return player name, team, total goals scored, and assists. Demonstrates MongoDB's aggregation pipeline with grouping and filtering.

2. **Team Win Rate: Home vs Away Comparison**
   Calculate win percentages for each team when playing at home versus away. Compute total matches, wins, and win rates for both scenarios. Identify teams with the biggest home advantage differential. Showcases conditional aggregation logic.

3. **Head-to-Head Historical Record**
   Given two team names, find all historical matches between them. Return match date, home team, away team, final score, venue, and season. Include aggregate statistics: total wins for each team, draws, and total goals scored. Demonstrates filtered search with sorting.

4. **Player Rating Evolution Over Time**
   Given a player name, query the player attributes collection ordered by date. Return temporal data showing overall rating, potential rating, position, and age across multiple seasons. Illustrates time-series analysis capabilities and temporal data handling.

5. **Most Disciplined vs Most Aggressive Teams**
   Aggregate all yellow and red cards from match events, group by team name. Calculate total yellow cards, total red cards, and card points (yellow=1, red=3). Return top 10 most aggressive and top 10 most disciplined teams. Demonstrates array query capabilities and ranking.

6. **Goals Scored by Match Time Period**
   Extract goal events and bucket them into time periods: 0-15 min (early), 16-30 min, 31-45 min (end of first half), 46-60 min, 61-75 min, and 76-90+ min (late goals). Count goals in each period and calculate percentages. Reveals scoring patterns and demonstrates bucketing/binning aggregation.

7. **Team Attributes Impact on Match Outcomes**
   For each match, compare team overall rating differences, build-up play speed differences, and defense pressure differences. Join with match outcomes. Aggregate to determine: when rating difference exceeds 10 points, what is the win percentage?

Return rating difference buckets, win rates, and sample sizes. Directly supports machine learning feature engineering and demonstrates complex joins and correlation analysis.

Extension Component

Selected Extensions: Web UI + Machine Learning Integration

I will implement both a Flask web application and a machine learning prediction model, combining two extension components for a comprehensive demonstration.

Machine Learning Component

A match outcome prediction model trained on historical match data and team attributes:

- **Features:** Team overall ratings, attack ratings, defense ratings, home field advantage, and recent form statistics
- **Model:** Random Forest Classifier predicting three outcomes (Home Win, Draw, Away Win)
- **Integration:** Users select two teams via the web interface, and the model returns win probabilities for each outcome with percentage confidence scores

Web UI Features

A Flask-based web application with three main interactive pages:

- **Match Predictor Page:** Form with dropdown menus for selecting Team 1, Team 2, and venue (Home/Away). A 'Predict Match Outcome' button triggers ML predictions and displays win probability percentages along with the team attributes used in the prediction
- **Query Dashboard:** Interactive buttons and filters for exploring top scorers by league/season, viewing team head-to-head records, and generating league tables for any season. Results displayed in clean, formatted HTML tables
- **Player Stats Explorer:** Search functionality for player names with visualizations showing rating trends over seasons, career statistics, positions, and goal contributions

Expected Output

The final deliverable will be a fully functional Flask web application that integrates MongoDB queries with machine learning predictions. Users will be able to:

- **Predict match outcomes** between any two teams using the trained ML model with probability distributions
- **Explore historical soccer statistics** through interactive buttons, forms, and dropdown filters
- **View real-time query results** from MongoDB displayed in clean, professional HTML tables
- **Analyze player performance trends** with visual charts showing rating evolution across seasons

During the final presentation, I will demonstrate the live web application by running sample queries, showing the ML prediction interface, and highlighting how MongoDB's features enable efficient data retrieval and analysis for soccer analytics.

Project Repository

All code, documentation, and query demonstrations will be maintained in a GitHub repository with a comprehensive README file detailing setup instructions, database schema design, and usage examples.