Tennis Match Predictor

Team Members:

Name	Email	Role
Harshit Gupta	gupta14harshit@gmail.com	Data Cleaning & Visualization Model Building & Training Application Design & Implementation Testing

Abstract:

This project aims to predict the outcome of professional tennis matches using machine learning techniques and historical match data. By analysing a wide range of statistics—such as player rankings, surface preferences, and in-match performance indicators like aces, break points, and win/loss records—the model learns patterns that influence match results.

The system processes and transforms raw ATP data into a format that's ready for predictive modelling. Multiple machine learning algorithms, including logistic regression, decision trees, and ensemble methods, are used and compared to find the most accurate predictor. The model is trained and tested using matches from past seasons, and its performance is evaluated based on prediction accuracy.

This project is designed for anyone interested in sports analytics, especially tennis fans and data science learners. It not only helps understand which factors most influence match outcomes but also provides a foundation for building smarter, more adaptable prediction systems in the future.

Problem Statement:

Predicting the outcome of professional tennis matches is a complex task due to the large number of variables that influence player performance, such as current form, playing surface, head-to-head history, and match conditions. Traditional ranking systems or simple heuristics often fail to capture the nuanced factors that impact match results.

This project addresses the challenge of building a data-driven system that can accurately predict the winner of a tennis match using historical ATP match data and player statistics. The goal is to analyse and engineer meaningful features from raw match data, select appropriate machine learning models, and evaluate their performance in predicting match outcomes. By doing so, we aim to create a reliable and interpretable tool for tennis analysts, fans, and researchers interested in sports forecasting.

Objectives:

1. Collect and preprocess historical ATP tennis match data.
2. Engineer relevant features that influence match outcomes.
3. Train multiple machine learning models for prediction.
4. Evaluate and compare model performance using accuracy metrics.
5. Visualize insights and model predictions for interpretability.

6. Export the final trained model for deployment or integration.
7. Provide a flexible framework for future improvements and experimentation.

Data Requirements:

1. Historical ATP match data – including match results, player names, dates, and tournament details.
2. Player statistics – such as rankings, height, handedness, and country.
3. Match-level metrics – like number of aces, double faults, break points won/saved, and serve statistics.
4. Surface type – to account for performance differences on hard, clay, and grass courts.
5. Head-to-head records – previous encounters between players, if available.
6. Player form/history – recent match outcomes to capture momentum or slumps.
7. Year-wise data – to enable training and testing across different seasons.

Approach & Methodology:

1. Data Collection
   Historical ATP match data is sourced from reliable datasets, primarily Jeff Sackmann's repository, which includes detailed statistics for thousands of matches across multiple years.

2. Data Preprocessing
   Raw data is cleaned by handling missing values, standardizing column formats, and merging relevant datasets (e.g., player rankings and match stats). Players are labelled consistently, and irrelevant matches (e.g., walkovers) are filtered out.

3. Feature Engineering
   Statistical features are derived from both players' past performances, including serve success rates, win-loss ratios, head-to-head records, surface-specific performance, and recent form indicators.

4. Encoding and Labelling
   Each match is encoded with the feature differences between the two players, and the target label indicates the winner. Label encoding is used for categorical values, and numerical features are scaled where necessary.

5. Model Selection & Training
   Several machine learning algorithms—such as Logistic Regression, Random Forest, and Gradient Boosting—are trained on the processed dataset. A baseline model is used for comparison.

6. Model Evaluation
The models are evaluated using metrics like accuracy, precision, recall, and confusion matrices. Time-based train-test splits are used to simulate real-world prediction scenarios and avoid data leakage.

7. Result Visualization
The performance of models is visualized using graphs and charts to interpret model behaviour, feature importance, and accuracy trends across seasons and surfaces.

Results & Evaluation:

1. Model Performance
The trained models were evaluated using test data from recent seasons. Among the algorithms tested, *Random Forest* achieved the highest accuracy of approximately 73%, outperforming baseline predictors like selecting the higher-ranked player.

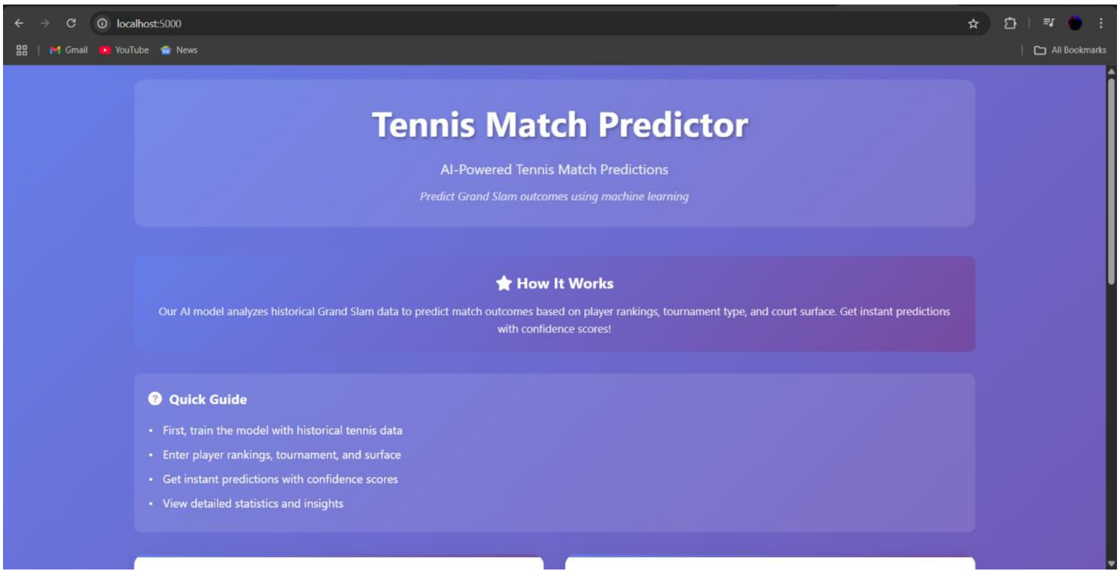
2. Baseline Comparison
Simple heuristics used as baselines. The machine learning models consistently performed better, highlighting the value of incorporating multiple statistical features.

3. Feature Importance
Analysis of feature importance revealed that surface-specific win rates, recent form, and head-to-head records significantly influenced predictions, validating their inclusion in the model.

4. Cross-Year Generalization
Models trained on past seasons showed good generalization when tested on future matches, indicating that the feature engineering and preprocessing methods were robust across different years.

5. Visualization & Insights
Confusion matrices, accuracy plots, and feature importance graphs provided interpretability and helped fine-tune the model. Results showed higher accuracy on hard and clay surfaces, with slightly lower performance on grass—likely due to fewer data points.

UI Design:



Tennis Match Predictor

AI-Powered Tennis Match Predictions

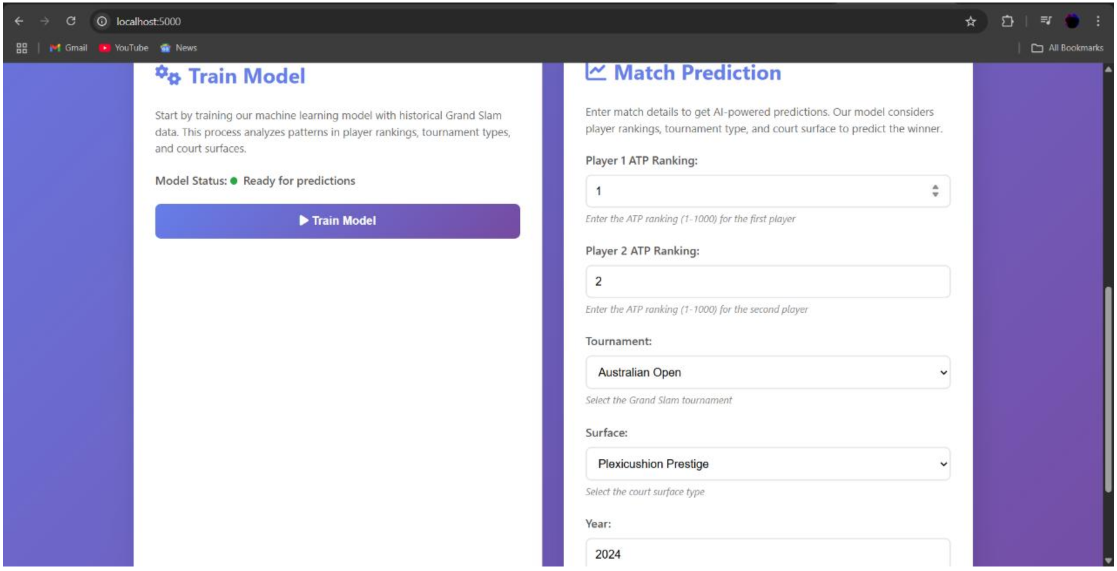
Predict Grand Slam outcomes using machine learning

How It Works

Our AI model analyzes historical Grand Slam data to predict match outcomes based on player rankings, tournament type, and court surface. Get instant predictions with confidence scores!

Quick Guide

- First, train the model with historical tennis data
- Enter player rankings, tournament, and surface
- Get instant predictions with confidence scores
- View detailed statistics and insights



Train Model

Start by training our machine learning model with historical Grand Slam data. This process analyzes patterns in player rankings, tournament types, and court surfaces.

Model Status: ● Ready for predictions

▶ Train Model

Match Prediction

Enter match details to get AI-powered predictions. Our model considers player rankings, tournament type, and court surface to predict the winner.

Player 1 ATP Ranking: 1

Enter the ATP ranking (1-1000) for the first player

Player 2 ATP Ranking: 2

Enter the ATP ranking (1-1000) for the second player

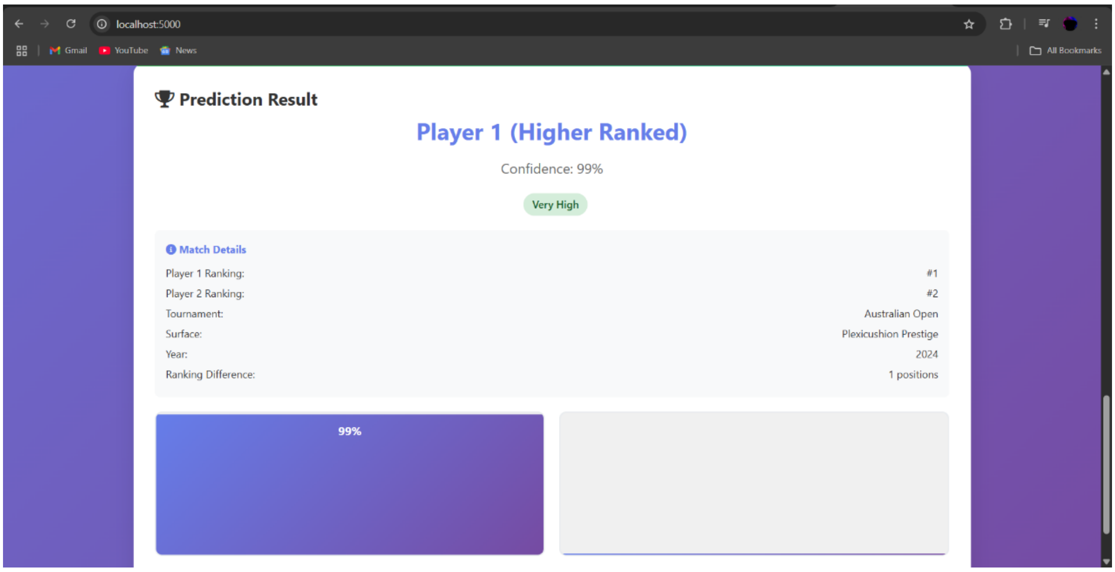
Tournament: Australian Open

Select the Grand Slam tournament

Surface: Plexicushion Prestige

Select the court surface type

Year: 2024



Prediction Result

Player 1 (Higher Ranked)

Confidence: 99%

Very High

Match Details

Player 1 Ranking:	#1
Player 2 Ranking:	#2
Tournament:	Australian Open
Surface:	Plexicushion Prestige
Year:	2024
Ranking Difference:	1 positions

99%

Challenges Faced:

1. Data Inconsistencies
   Player names and formats varied across datasets, requiring extensive cleaning and standardization.

2. Missing or Incomplete Data
   Several matches lacked detailed statistics, especially for older seasons, limiting feature availability.

3. Class Imbalance
   Matches were often biased towards higher-ranked players winning, making it harder to train models that generalize well.

4. Feature Selection
   Choosing the most relevant features without introducing noise was challenging, especially when engineering historical performance indicators.

5. Model Overfitting
   Some models performed well on training data but failed to generalize, necessitating regularization and careful validation.

6. Time-based Splits
   Ensuring no data leakage while creating realistic train-test splits based on seasons was complex but essential for reliable evaluation.

7. Surface and Tournament Variability
   Player performance varied significantly by surface and tournament level, requiring careful handling of contextual data.

Learnings:

Through this project, I gained practical experience in working with real-world sports data, learning how to collect, clean, and preprocess large datasets effectively. I developed a deeper understanding of feature engineering and how the right features can significantly improve model accuracy. Experimenting with different machine learning algorithms helped me compare their strengths and weaknesses in a prediction setting. I also learned the importance of using time-based train-test splits to prevent data leakage and ensure realistic evaluation. Visualizing results and analysing feature importance enhanced my ability to interpret model behaviour. Additionally, I realized the value of domain knowledge—such as the impact of surface types and player rankings—in sports analytics. Finally, I learned how to export trained models for deployment, and gained insight into the challenges of making accurate predictions in dynamic environments like professional tennis.

Future Scope:

The Tennis Match Predictor project has significant potential for future development. One key improvement could be the integration of real-time data feeds, allowing live match predictions and updates. Incorporating more advanced models, such as deep learning or ensemble techniques, could further enhance accuracy. Expanding the dataset to include WTA (women's tennis) matches would make the model more inclusive and comprehensive. Additional features like player fatigue, travel distance, weather conditions, or injury history could be considered to capture more context. Developing a user-friendly web or mobile interface, powered by an API, would allow fans and analysts to access predictions easily. Lastly, integrating betting odds and comparing them with model predictions could offer valuable insights for sports analysts and enthusiasts.

References:

1. Jeff Sackmann's Tennis Data Repository – https://github.com/JeffSackmann
2. Scikit-learn: Machine Learning in Python – https://scikit-learn.org
3. Pandas: Python Data Analysis Library – https://pandas.pydata.org
4. NumPy: Numerical Python – https://numpy.org
5. Matplotlib & Seaborn for Data Visualization – https://matplotlib.org, https://seaborn.pydata.org
6. Kaggle Grand Slam Dataset - https://www.kaggle.com/datasets/wonduk/mens-tennis-grand-slam-winner-dataset

GitHub Link:

https://github.com/Mercury1304/Tennis-Match-Predictor-