PROJECT REPORT

Title: Transfer IQ – Dynamic Player Transfer Value Prediction using AI and Multi-Source Data

Duration: 8 Weeks
Objective: To develop an AI-driven model that predicts football player transfer values by combining multi-source data such as player performance, injury history, social-media sentiment, and market trends.

WEEK 1 – DATA COLLECTION AND INITIAL EXPLORATION

Day 1: The project began with identification of all data sources—StatsBomb Open Data for player performance, Transfermarkt for market values, Twitter API for sentiment, and public sports databases for injury records. Python libraries such as pandas, numpy, BeautifulSoup, tweepy, and matplotlib were installed.

Day 2: Player performance data were collected from StatsBomb, including goals, assists, passes, and minutes played. These were stored in CSV format.

Day 3: A scraping script was written to gather player market values from Transfermarkt. More than 400 players' names, clubs, and current values were extracted successfully.

Day 4: Tweets mentioning players were collected through the Twitter API. Data were cleaned and analyzed using the VADER sentiment analyzer to generate sentiment scores.

Day 5: Historical injury records were obtained and merged with the main player dataset using player IDs, containing injury duration and type.

Day 6: Exploratory Data Analysis (EDA) was conducted to identify trends, missing values, and outliers. Boxplots and histograms were created for visual inspection.

Day 7: A weekly summary documented all datasets, their structure, and initial insights.

Deliverables: Raw datasets from all sources and an initial EDA report.

WEEK 2 – DATA CLEANING AND PREPROCESSING

Day 1: Datasets were checked for duplicates and missing values; inconsistencies were removed. Day 2: Missing values were imputed, and data were standardized into a uniform schema.

Day 3: A feature-engineering plan was prepared to include trend metrics, injury risk, and contract duration.

Day 4: New performance and injury-related features were created such as form index and injury frequency.

Day 5: Numerical features were normalized and categorical variables were one-hot encoded.

Day 6: Sentiment features were cleaned and integrated with player data.

Day 7: All processed datasets were validated and documented for model readiness.

Deliverables: Cleaned, pre-processed, and feature-engineered datasets with an initial sentiment analysis report

WEEK 3 – ADVANCED FEATURE ENGINEERING AND SENTIMENT ANALYSIS (PART 1)

Day 1: Advanced trend features like rolling averages were developed.

Day 2: Injury recovery rate and severity index were computed.

Day 3: Sentiment analysis was enhanced by combining VADER and TextBlob outputs.

Day 4: Sentiment and performance data were merged to observe their correlation.

Day 5: Redundant and low-impact features were removed.

Day 6: Correlation heatmaps were prepared for visualization.

Day 7: A progress report was written summarizing intermediate feature-engineering results.

Deliverables: Enhanced features capturing player form, injury impact, and sentiment.

WEEK 4 – ADVANCED FEATURE ENGINEERING AND SENTIMENT ANALYSIS (PART 2)

Day 1: Final refinement of engineered features was carried out to improve accuracy.

Day 2: Sentiment trends were analyzed over time to see perception shifts.

Day 3: Dataset balancing and normalization were performed.

Day 4: The dataset was split into training and testing sets.

Day 5: Sentiment integration was verified through ID matching.

Day 6: Descriptive statistics were produced for all variables.

Day 7: The finalized dataset was archived and prepared for LSTM model training.

Deliverables: Final feature-rich dataset, sentiment-trend insights, and complete data documentation.

WEEK 5 – LSTM MODEL DEVELOPMENT

Day 1: A univariate LSTM architecture was created using TensorFlow to predict transfer values from performance data.

Day 2: The model was trained and evaluated; loss curves were analyzed.

Day 3: The model was extended to a multivariate LSTM including injury and sentiment data, which improved accuracy.

Day 4: An encoder-decoder LSTM was implemented for multi-step forecasting.

Day 5: Evaluation metrics such as RMSE and MAE were computed.

Day 6: Hyperparameters like epochs and batch size were tuned.

Day 7: Model weights were saved, and a report summarizing LSTM training results was produced.

Deliverables: Trained univariate and multivariate LSTM models with performance evaluations.

WEEK 6 – ENSEMBLE MODEL DEVELOPMENT AND INTEGRATION

Day 1: An XGBoost baseline model was trained using engineered features.

Day 2: LSTM outputs were integrated into XGBoost to form an ensemble.

Day 3: LightGBM was tested as an alternative boosting method.

Day 4: A final hybrid model combining LSTM and XGBoost was finalized.

Day 5: Validation on unseen data confirmed accuracy improvements.

Day 6: Comparative performance charts were generated.

Day 7: Documentation of ensemble integration and results was completed.

Deliverables: Hybrid LSTM + XGBoost model with superior performance and validation report.

WEEK 7 – MODEL EVALUATION, TUNING, AND TESTING

Day 1: Hyperparameter tuning began using grid search for XGBoost.

Day 2: LSTM layer and dropout adjustments were made.

Day 3: Models were evaluated using RMSE, MAE, and R^2 metrics.

Day 4: Ensemble models were tested on new validation data.

Day 5: Cross-validation verified model consistency.

Day 6: Comparative improvement graphs were prepared.

Day 7: A comprehensive evaluation summary was compiled.

Deliverables: Optimized models and detailed performance-comparison report.

WEEK 8 – FINAL EVALUATION, VISUALIZATION, AND REPORTING

Day 1: Final predictions for top players were generated.

Day 2: Interactive visualizations were developed using Plotly and Dash.

Day 3: Model outputs and charts were compiled into a formal report.

Day 4: Full documentation of methodology, models, and findings was prepared.

Day 5: A concise presentation deck was designed summarizing the project.

Day 6: The entire project workflow was reviewed for completeness.

Day 7: The final project report and presentation were submitted successfully.

Deliverables: Final predictions, visualization dashboards, comprehensive documentation, and presentation slides.

FINAL SUMMARY

Across eight weeks, the TransferIQ project achieved the development of a reliable AI-based system to predict football player transfer values. Through continuous data collection, cleaning, advanced feature engineering, and integration of LSTM and ensemble methods, the final model reached high predictive accuracy. Incorporating social-media sentiment and injury data provided additional insights into player valuation dynamics. The resulting system demonstrates the power of combining machine-learning and sentiment analysis in modern sports analytics.