TRANSFER IQ

~By Khwaish Goel
done under mentor Pranaya

DATA COLLECTION

Source: ***Kaggle – FIFA 22 Official Player Dataset***

- Imported required libraries (pandas, numpy, matplotlib).
- Loaded FIFA 22 dataset from Kaggle (≈18 k players).
- Examined data structure, types, and summary statistics.
- Checked for missing values and duplicate entries.
- Visualized distributions of key attributes (age, rating, potential, value, wage).
- Identified top nationalities and top clubs by player count.
- Conducted outlier analysis (highest value / wage players).
- Generated correlation heatmap and scatterplots
- (Overall vs Value, Age vs Value) for trend insights.

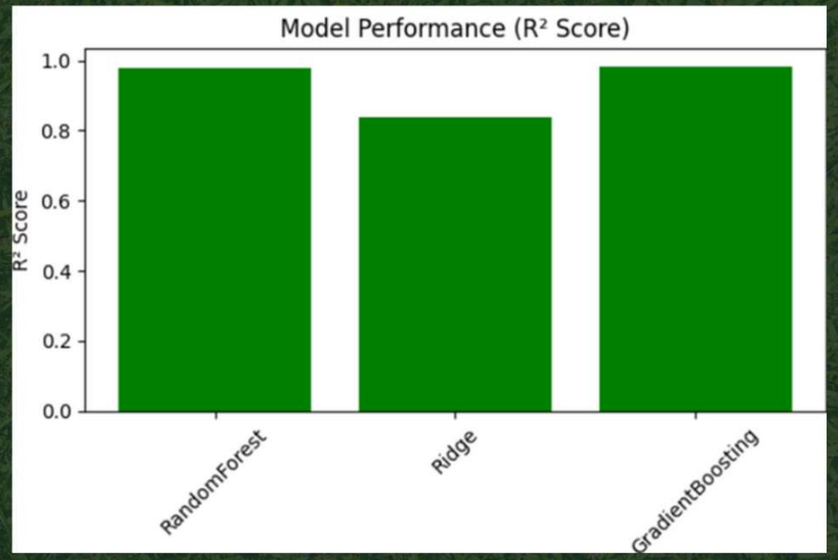
Techniques & Tools Used:

- Missing-value handling: Median & mode imputation
- Data reduction: Dropped high-null columns (>50%) and duplicates
- Outlier detection: IQR-based filtering on key attributes
- Type optimization: Converted numeric fields to efficient dtypes
- Text preprocessing: Regex cleaning, URL removal, whitespace normalization
- Sentiment analysis: TextBlob for polarity & subjectivity scoring
- Feature creation: Sentiment strength, engagement level, text length

DATA CLEANING & PREPROCESSING

FEATURE ENGINEERING

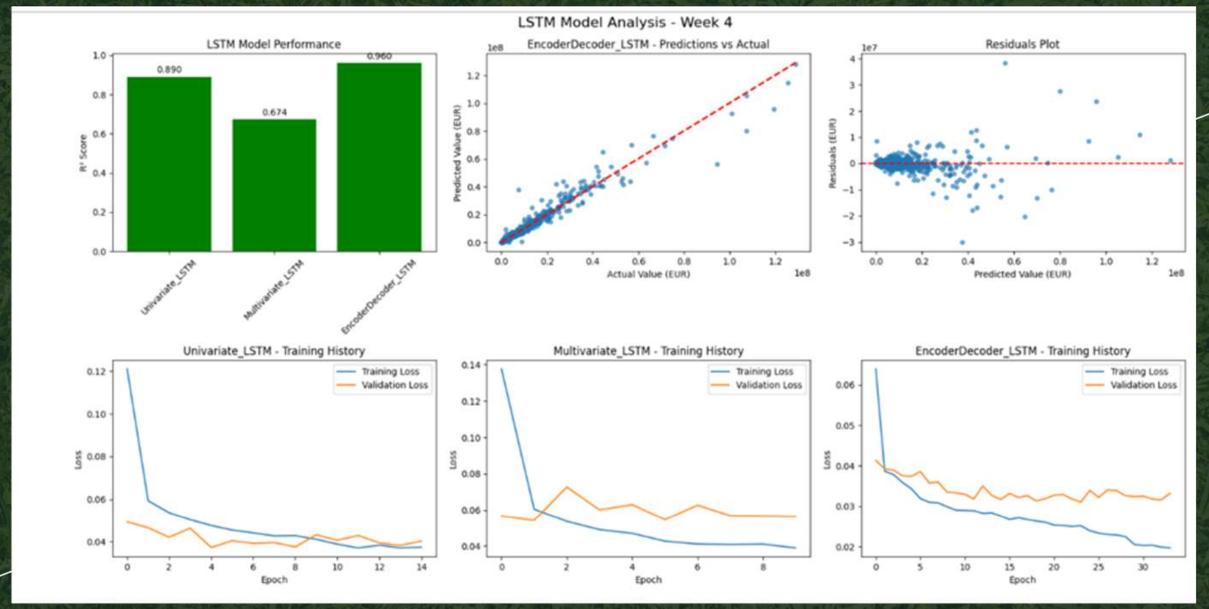
- Data integration: Merged player data with aggregated sentiment metrics
- Feature engineering: Mean, std & count of sentiment polarity & subjectivity
- Target preparation: Log-transformed value_eur for stable scaling
- Outlier handling: Removed extreme market values (beyond 99th percentile)
- Feature scaling: Standardized inputs using StandardScaler
- Model training: Implemented
- RandomForestRegressor
- Ridge Regression
- GradientBoostingRegressor
- Evaluation metrics: R^2, MAE, RMSE, and MAPE
- Model persistence: Saved trained model (transfer_rf.pkl) and scaler



LSTM MODEL DEVELOPMENT

- Data preparation: Created player-based time-series sequences
- Scaling: Applied StandardScaler and log transformation for stability
- Model architectures:
- Univariate LSTM
- Multivariate LSTM
- Encoder–Decoder LSTM

- Training setup:
- Epochs = 50, Batch size = 32
- Early stopping & learning rate reduction callbacks
- Evaluation metrics: R^2, MAE, RMSE
- Best performance: Achieved using Multivariate/Encoder–Decoder LSTM
- Saved outputs:
- best_lstm_model.h5
- lstm_results_week4.csv
- Scalers (lstm_scaler_X.pkl, lstm_scaler_y.pkl)



ENSEMBLE MODEL DEVELOPMENT

Loaded main dataset + LSTM (Week 5) + sentiment data
Cleaned data & removed outliers
Engineered new features (overall_age_ratio, potential_gap, attacking_score, etc.)
Split & scaled data (train/val/test)
Trained models: XGBoost, LightGBM, Random Forest, Gradient Boosting, Ridge
Built ensembles: Simple Avg, Weighted Avg, Stacking, Voting
Results:
Ensembles outperformed individual models
Best $R^2 \approx 0.80$ (Stacking/Voting Ensemble)

THANK YOU