

# Kicking It Up a Notch

Predicting Soccer Player Performance with Data Science

*April 28, 2023*





## Project Overview



## Methods



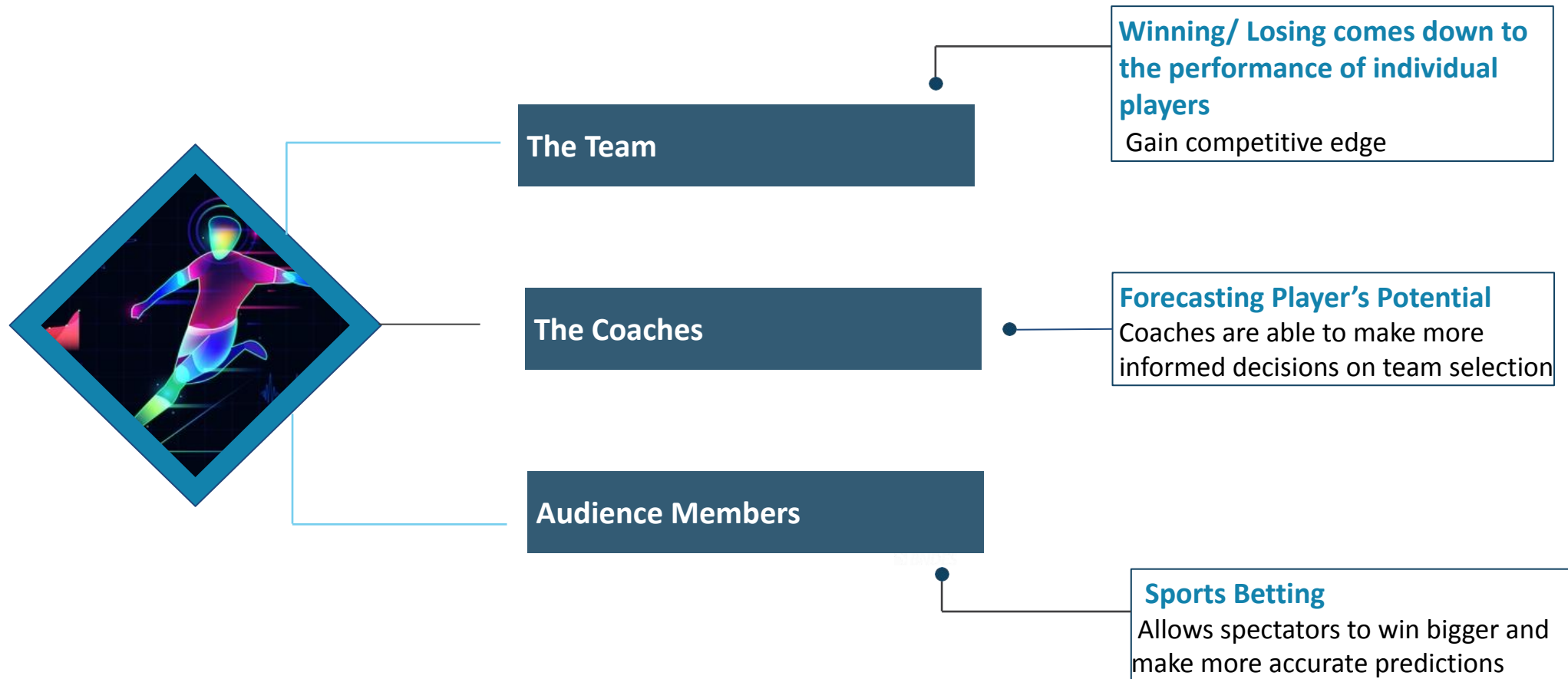
## Conclusion

# Project Overview

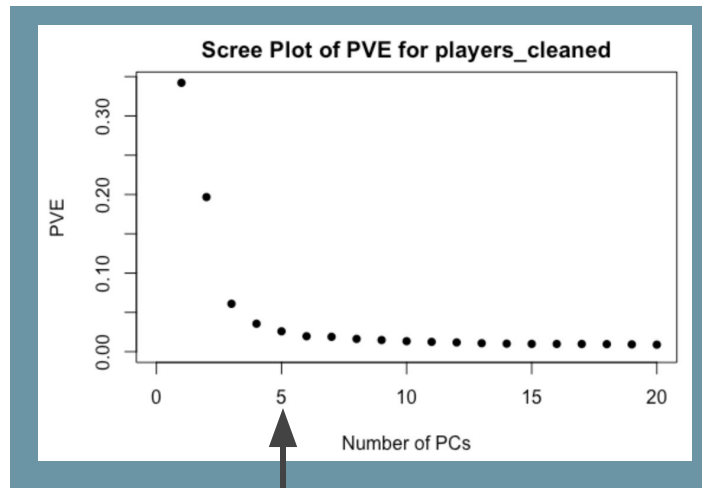


# The Importance of Predicting Soccer Player's Performance

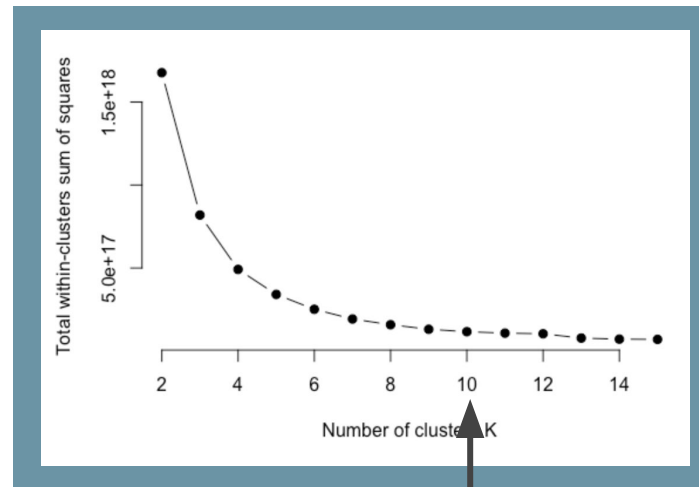
Huge impacts on scouting, trajectory of a team, and forecasting



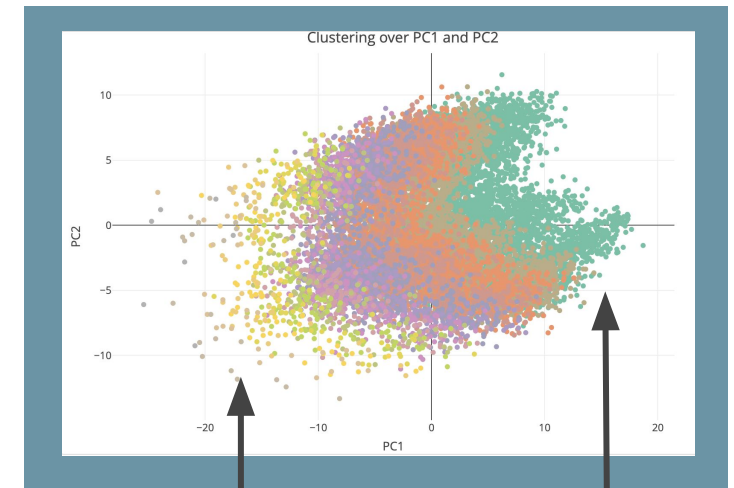
# Initial Findings



Optimal No. of PCs



Optimal No. of Clusters



Highest ranked Players

Lowest ranked Players



About the Project



Methods



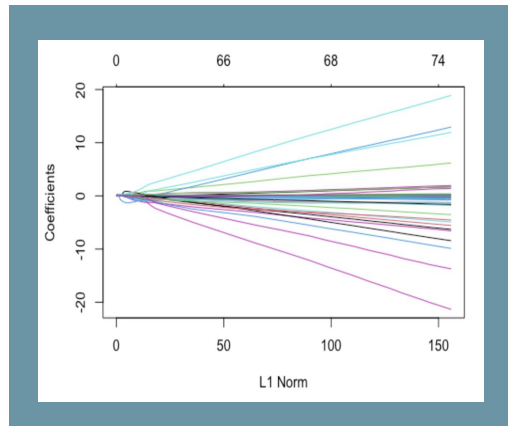
Conclusion

# Methods



# LASSO Regression

Finding Important Predictors



Key Stats

**Mean Squared Error:**

0.0033155

**Root Mean Squared Error:**

0.057581

Call:

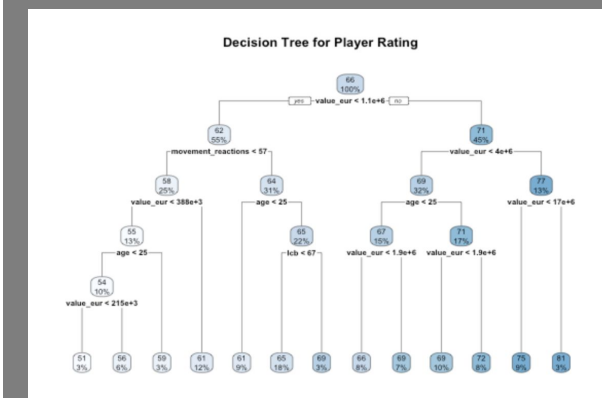
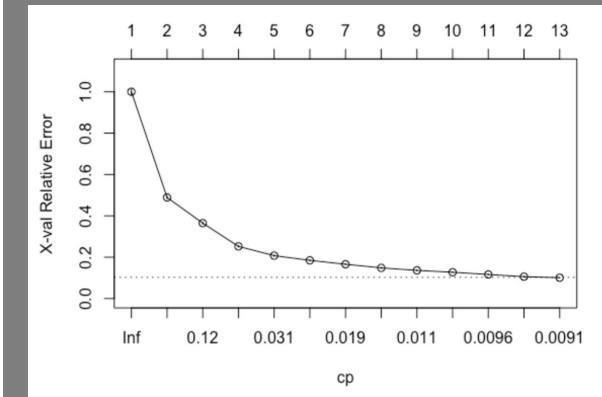
```
lm(formula = overall ~ value_eur + wage_eur + age + height_cm +
  weight_kg + club_team_id + league_level + nationality_id +
  preferred_foot + weak_foot + skill_moves + international_reputation +
  work_rate + release_clause_eur + pace + passing + dribbling +
  physic + attacking_crossing + attacking_finishing + attacking_heading_accuracy +
  attacking_short_passing + attacking_volleys + skill_dribbling +
  skill_curve + skill_fk_accuracy + skill_long_passing + movement_acceleration +
  movement_sprint_speed + movement_agility + movement_reactions +
  movement_balance + power_shot_power + power_jumping + power_stamina +
  power_strength + power_long_shots + mentality_aggression +
  mentality_interceptions + mentality_positioning + mentality_vision +
  mentality_penalties + mentality_composure + defending_marking_awareness +
  defending_standing_tackle + defending_sliding_tackle + goalkeeping_diving +
  goalkeeping_handling + goalkeeping_kicking + goalkeeping_positioning +
  goalkeeping_reflexes + ls + lam + lcm + lwb + ldm + gk +
  `Position:RW` + `Position:ST` + `Position:CF` + `Position:LW` +
  `Position:CAM` + `Position:CM` + `Position:CDM` + `Position:LM` +
  `Position:CB` + `Position:RB` + `Position:RM` + `Position:LB` +
  `Position:RWB` + year_contract_length + PC1 + PC2 + PC3 +
  PC4 + PC5, data = train_data[, c(1, selected_features + 1)])
```

Final Model

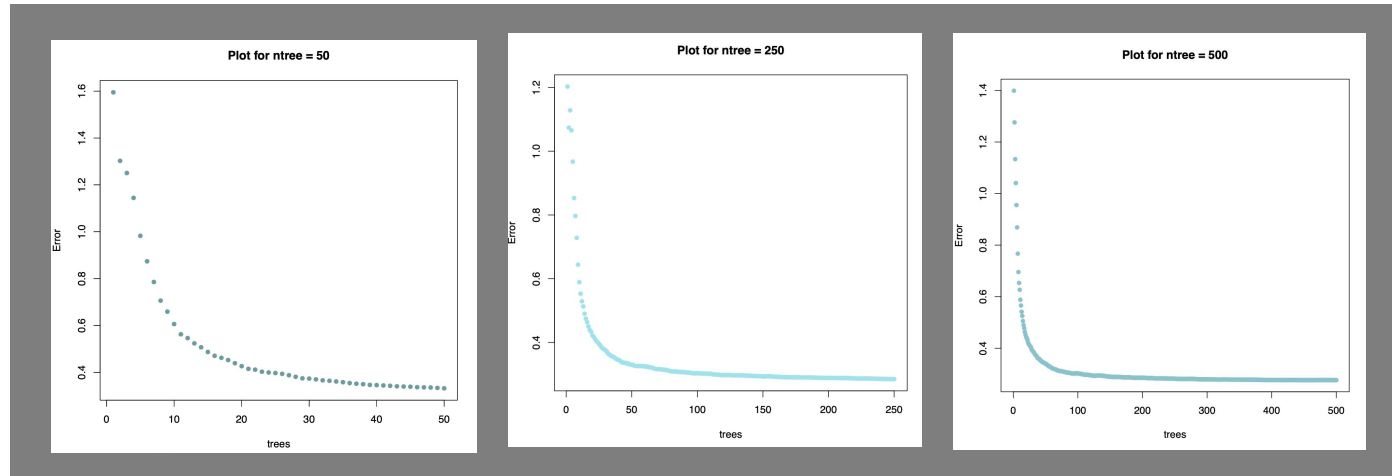
# Trees

Single Tree, Bagged Tree, Random Forest Regression

Single Tree



Random Forest



Single Tree

**Mean Squared Error:**  
0.4.2920

**Root Mean Squared Error:**  
2.0717

**Testing Error:**  
13820

Bagged Tree

**Mean Squared Error:**  
0.09728

**Root Mean Squared Error:**  
0.3119

**Testing Error:**  
371.697

Random Forest

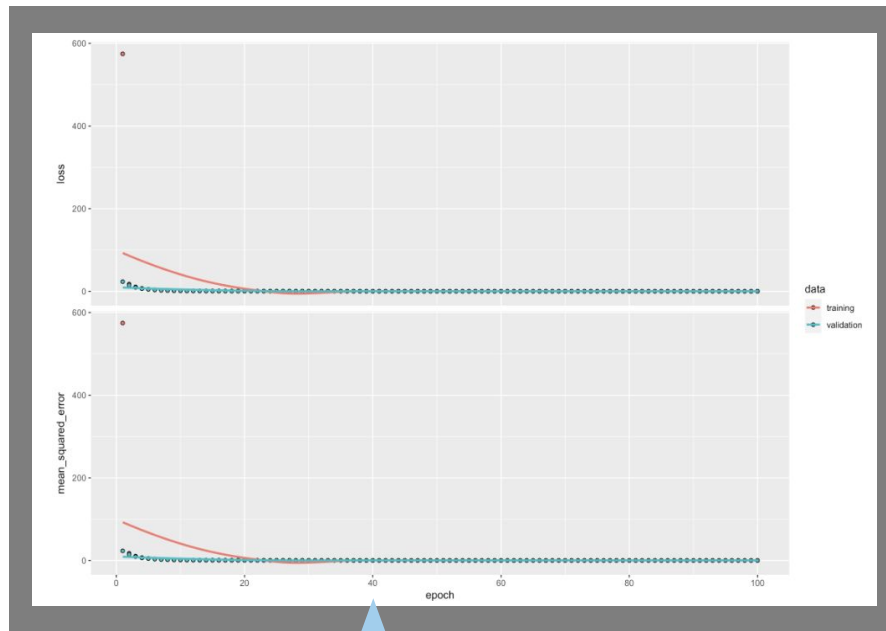
**Mean Squared Error:**  
0.080496

**Root Mean Squared Error:**  
0.283718

**Testing Error:**  
259.1974

# Keras Model

Optimizing the Model



**epochs = 40:** loss and mean squared error stabilize around 40.

## Key Stats

**Mean Squared Error:**  
0.008929

**Root Mean Squared Error:**  
0.094496





About the Project



Methods



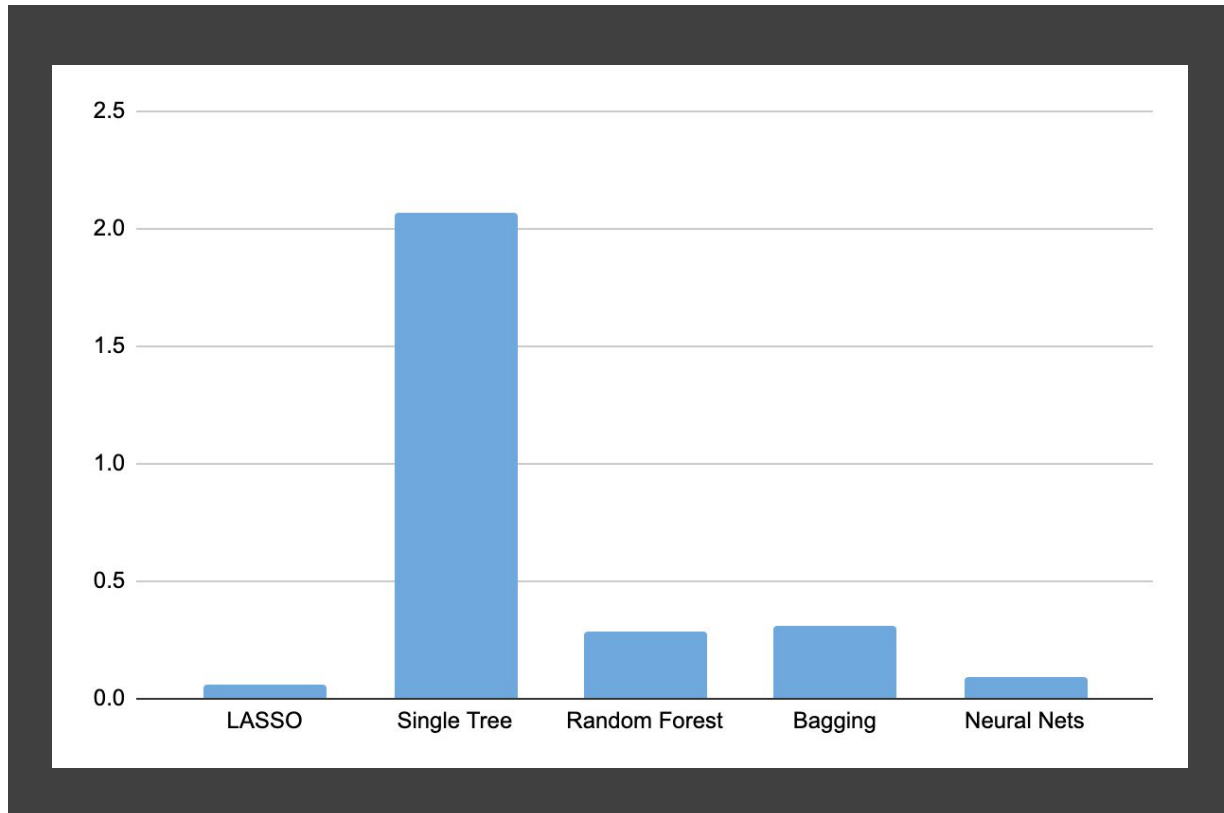
Conclusion

# Conclusion



# Compiled RMSE

## Success of Different Methods by RMSE



### Results: Model RMSE Values

LASSO 0.057581

Single Tree 2.0717

Random Forest 0.283718

Bagging 0.3119

Neural Nets 0.094496

# Future Goals

