PREDICTING THE SOCCER TRANSFER MARKET

TOMAS FIURE, MAXIMO JALIFE, AMIN BEN BRAHIM, BLAKE GARBER

Outline

- References
- An overview of soccer, the transfer market
- About us
- Our strategy for innovation within transfer price prediction
- Data collection and cleaning
- Machine learning models
- Implementation
- Findings
- Future research

Key References

Poli, R., Besson, R., & Ravenel, L. (2021, December 23). *Econometric approach to assessing the transfer fees and values of professional football players*. MDPI. https://www.mdpi.com/2227-7099/10/1/4

Transfermarkt. (2022, August 4). *Data Administration, market values and watchlist - become a part of the Transfermarkt Community*. Transfermarkt. https://www.transfermarkt.co.in/transfermarkt-market-value-explained-how-is-it-determined-/view/news/385100

McHale, I., & Holmes, B. (2022, June 20). *Estimating transfer fees of professional footballers using advanced performance metrics and Machine Learning*. European Journal of Operational Research. https://www.sciencedirect.com/science/article/pii/S0377221722005082#fn0004

Cariboo, D. (2023b, October 27). *Football data from Transfermarkt*. Kaggle. https://www.kaggle.com/datasets/davidcariboo/player-scores

Other References

Deloitte Football Money League 2023. Deloitte United Kingdom. (n.d.). https://www2.deloitte.com/uk/en/pages/sports-business-group/articles/deloitte-football-money-league.html

Smith, R. (2021, August 12). *The wisdom of the crowd*. The New York Times. https://www.nytimes.com/2021/08/12/sports/soccer/soccer-football-transfermarkt.html

Coates, D., & Parshakov, P. (2021, October 29). *The wisdom of crowds and transfer market values*. European Journal of Operational Research. https://www.sciencedirect.com/science/article/pii/S037722172100895X

Panja, T. (2021, September 2). *Soccer's new rich leave the old guard looking beleaguered*. The New York Times. https://www.nytimes.com/2021/09/02/sports/soccer/soccer-transfer-market-lionel-messi.html

Kargin, Kerem (2021, April 17). *Ridge Regression Fundamentals and Modeling in Python* https://keremkargin.medium.com/ridge-regression-fundamentals-and-modeling-in-python-bb56f4301f62

Chen, T., & Guestrin, C. (2016). XGBoost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/2939672.2939785

Other References

Tseng, G. (2018, November 29). *Gradient boosting and xgboost*. Medium. https://medium.com/@gabrieltseng/gradient-boosting-and-xgboost-c306c1bcfaf5

YouTube. (2019, April 1). *Gradient boost part 2 (of 4): Regression details*. YouTube. https://www.youtube.com/watch?v=2xudPOBz-vs

TIPPETT, J. (2019). *The expected goals philosophy*. INDEPENDENTLY PUBLISHED

Why do we care about the soccer transfer market?

Soccer and the Transfer Market

- Players can be exchanged between clubs in what is referred to as the "transfer market"
- Functions much like a free market
  - Values increase/decrease based on what clubs are willing to pay (in general)
- Incredibly lucrative sport, with biggest teams generating over 500 million dollars in annual revenue
- Player transfers represent major investments on a team's balance sheet
  - Neymar transferred from Barcelona to PSG for $263 million in 2017
- Transfer prices are determined through negotiations factoring player's skill level, team needs, marketability, market demand, and more

https://www2.deloitte.com/uk/en/pages/sports-business-group/articles/deloitte-football-money-league.html

About Us

Tomi:

- Studying applied math, CS, and Econ
- **Avid soccer and football fan, follows Club Atlético Independiente and Argentina**

Amin:

- Studying applied math and CS
- **Avid soccer fan, follows Manchester United and England**

Blake:

- Studying applied math
- **Avid sports fan in general, follows Football and Baseball**

Maximo:

- Studying applied math and CS
- **Avid soccer fan, follows Club Atlético River Plate and Argentina**

What is TransferMarkt (TM)?

- TransferMarkt.com (2000) is a website that uses crowdsourced opinions to assign 'expected values of players in a free market'
  - Transfers usually happen once a season during the 'transfer window', if they happen
  - There is no live updating 'market-price' of players
- TM values are usually updated twice a season
- TM player values are constantly referenced and highly esteemed in the soccer world
  - Regularly cited as accurate value of a player, often used in salary negotiations
  - Quoted by club presidents and chief executives when asked about decisions
  - Teams have used these values in official financial fillings to investors
  - Values formed basis of legal proceedings

Smith, R. (2021, August 12). *The wisdom of the crowd.* The New York Times. https://www.nytimes.com/2021/08/12/sports/soccer/soccer-football-transfermarkt.html

How are TM assigned?

- Values come from a mix of crowdsourced opinions and moderation from TM employees
  - “Transfermarkt does not use an algorithm but instead relies on the wisdom of the community” - transfermarkt
- Exact methods they employ are not revealed, but TM mentions relevant vague factors
  - Future prospects, performance at club and national team, level and status of the league in sporting and financial terms, injury susceptibility, marketing value, salary, general demand and "trends" on the market
- Some of these features are difficult to quantify

Smith, R. (2021, August 12). *The wisdom of the crowd.* The New York Times. https://www.nytimes.com/2021/08/12/sports/soccer/soccer-football-transfermarkt.html

Predicting TM Values

- Transfermarkt functions much like a 'black box'
  - "There is no algorithm or spreadsheet, It is a qualitative approach. We weigh arguments, meet with our moderators, and find a compromise." - Thomas Lintz, Transfermarkt's managing director
- **Our Goal:** Use ML models to predict TM values using player, league, and club statistics/factors
  - Reveal inner-workings of TM valuations methods
  - Show which features are most important to estimate fair value of a player

Smith, R. (2021, August 12). *The wisdom of the crowd.* The New York Times. https://www.nytimes.com/2021/08/12/sports/soccer/soccer-football-transfermarkt.html

Data Collection

Data Collection

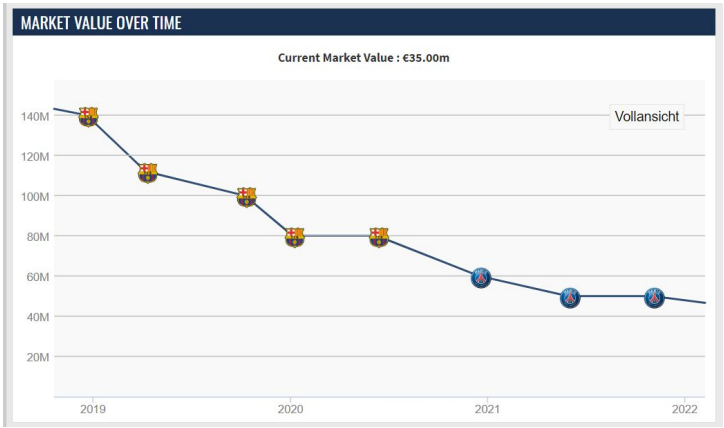
Player Statistics (Features)

- TransferMarkt mentions hard to quantify features in their 'model,' so we sought to replicate the process with quantifiable features
- Useful player stats (team, age, goals scored, passes, minutes, position…) [FBREF website]
- Aggregated data from 2020-2021 and 2021-2022 seasons, since immediately precede the time of the valuations
- Issues with adding additional seasons, balance between robust per player and enough players in sample.
- Stuck to top 5 leagues in Europe (England, France, Spain, Germany and Italy) because we were able to find advanced stats and player wages for those 5 leagues, as well as transfer values in Kaggle data
- Demo:https://colab.research.google.com/drive/1_cnE5eA9gcKwaPZjMNbSW_F_KBUbXvt7?authuser=1#scrollTo=8f3e8558

Data Collection

TM Market Values (Labels)

- Dollar valuations given by TM at the end of 2021-2022 season [kaggle CSV], verified kaggle data with a few randomly selected player values directly from TM website
- Used earliest available transfer value on or after June 2022, with the latest being January 2023
- Transfer values in Euros
- Example TM for player:



Cariboo, D. (2023b, October 27). *Football data from Transfermarkt.* Kaggle. https://www.kaggle.com/datasets/davidcariboo/player-scores

Data Selection

- Selected features which should be correlated with TM value (goals, assists, minutes played…) and that are mentioned in the TM website
- Added player wage, mainly as a proxy for player reputation, and club transfer business, as an indicator of the effect of a player's club on their TM price.
- Feature selection decision based on data available and papers, particularly Poli paper

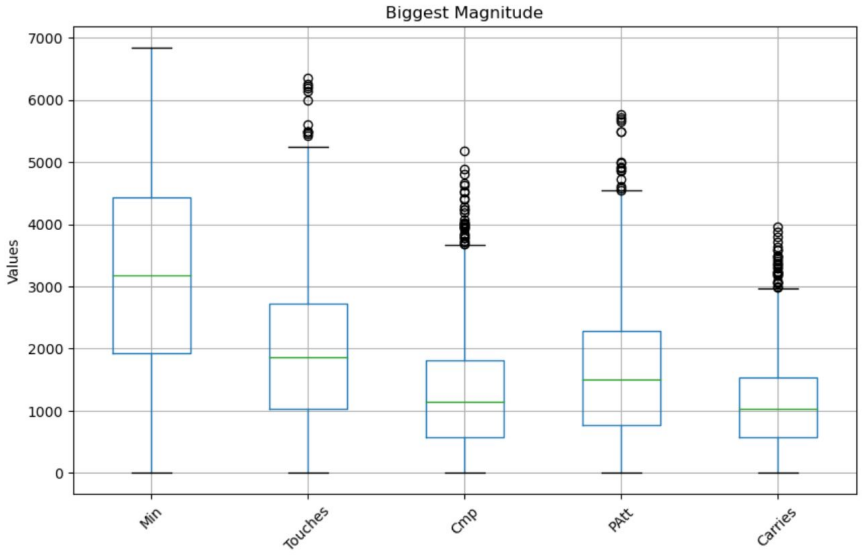
Poli, R., Besson, R., & Ravenel, L. (2021, December 23). *Econometric approach to assessing the transfer fees and values of professional football players*. MDPI. https://www.mdpi.com/2227-7099/10/1/4

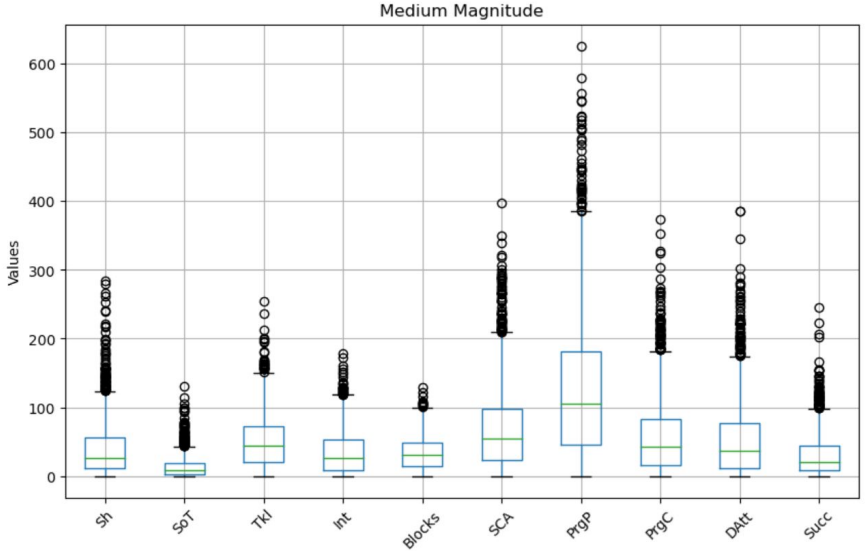
Data Engineering

- Match data and team transfers data had separate name formats, used Levenshtein Distance to match the name columns
- Removed entries with duplicate names in the transfer value data, to avoid conflicts when merging with match data.
- Removed goalkeepers as they are priced differently to outfield players
- Added columns that normalised statistics per minute
- Added modified age column to reflect player age in years as opposed to year and days.
- Generated correlation matrix to examine pairwise correlation among features and removed some features to avoid these correlations.
- Demo:
  https://colab.research.google.com/drive/1F8f5nHajxVpLHM5615xetXooS7e9AtVv?authuser=1#scrollTo=PTu2Nih92YiI

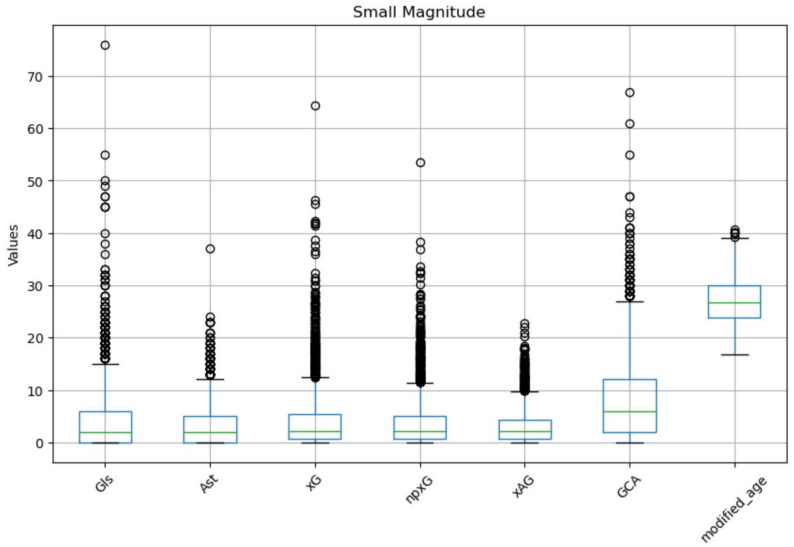
  https://colab.research.google.com/drive/1kyFYcuFGBZS116z07frgWn8F-zpbVr8A#scrollTo=Sd8LnWmrbe25

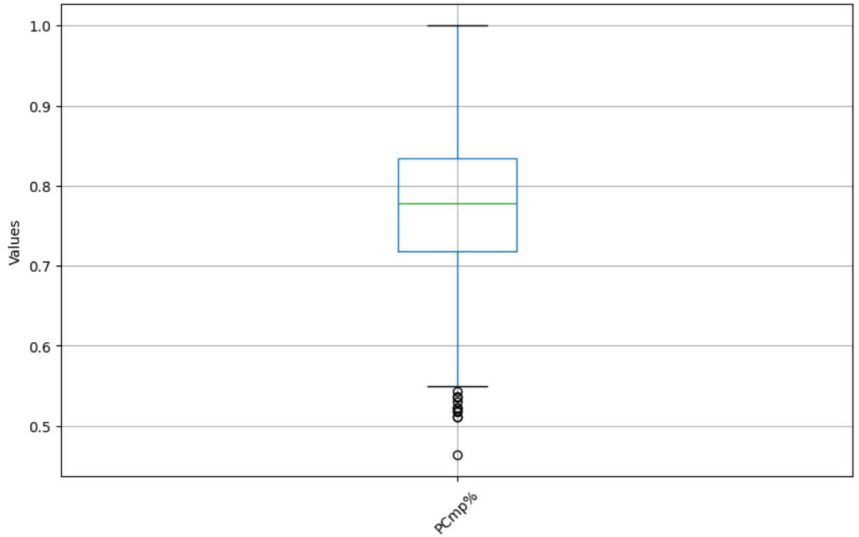
Data Exploration





Data Exploration





Data Exploration

	annual_wages	total_transfers	market_value_in_eur
count	1,678.00	1,678.00	1,678.00
mean	2,786,140.64	486.73	12,268,921.33
std	4,246,763.35	404.88	16,622,494.15
min	20,000.00	40.00	50,000.00
25%	640,000.00	165.70	2,300,000.00
50%	1,500,000.00	346.37	5,750,000.00
75%	3,210,000.00	721.91	16,000,000.00
max	63,640,000.00	1,752.00	160,000,000.00

Note: total_transfers is represented in millions of Euros, rest are in Euros

Models:

1. Linear Regression
2. Gradient Boosted Trees for Regression

Linear Regression

Linear Regression

- Linear Regression is used for predicting a continuous label (TM value) based on features (player stats).
- The goal is to model a linear relationship between the features and the target variable that minimizes some notion of error

We assume y is of the form: $$y = x\beta + \epsilon$$

y = true label

x = data point vector

β = weight vector

ϵ = error term

Linear Regression Assumptions

- Linearity: The relationship between the dependent and independent variables is linear
- Independence: Observations are independent of each other
- Homoscedasticity: The variance of the errors is constant across all levels of the independent variables
- Normality: For valid hypothesis testing, the errors should be normally distributed, although this is more of a concern when the sample size is small

Minimising Error (Ordinary Least Squares)

- We want to find a weight vector (β) that minimises the mean squared error MSE (ϵ^2)

$$\mathbf{X} = \begin{pmatrix} x_{00} & x_{01} & \dots & x_{0p} \\ x_{10} & x_{11} & \dots & x_{1p} \\ x_{20} & x_{21} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n0} & x_{n1} & \dots & x_{np} \end{pmatrix} \qquad \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix} \qquad \mathbf{y} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

- Let our prediction for each data point be: $F(x) = x\beta$
- We want to minimize the following over β, and we can call our solution β_{ols}

$$\text{Minimize: } \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}))^2 = ||X\beta - y||_2^2$$

Ridge Regression

- This is a variant of the Ordinary Least Squares Method that includes a regularization term λ
- λ controls a penalty on the weight coefficients to avoid overfitting and handle feature collinearity, and it always gives us a unique solution

$$\text{Minimize: } \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}))^2 + \lambda \sum_{i=1}^{p} \beta_i^2$$

$$\min_{\beta} \|X\beta - y\|_2^2 + \lambda \|\beta\|_2^2$$

- Note: solving this optimization problem is equivalent to solving the following constrained optimization problem:

$$\text{minimize } ||X\beta - y||^2$$

$$\text{such that } ||\beta||^2 \leq B$$

Ridge Regression Solution

$$\text{Minimize: } J(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left(y_i - (\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip})\right)^2 + \lambda \sum_{i=1}^{p} \beta_i^2$$

$$J(\beta) = (y - X\beta)^T (y - X\beta) + \lambda \beta^T \beta$$

$$y^T y - 2y^T X\beta + \beta^T X^T X\beta + \lambda \beta^T \beta$$

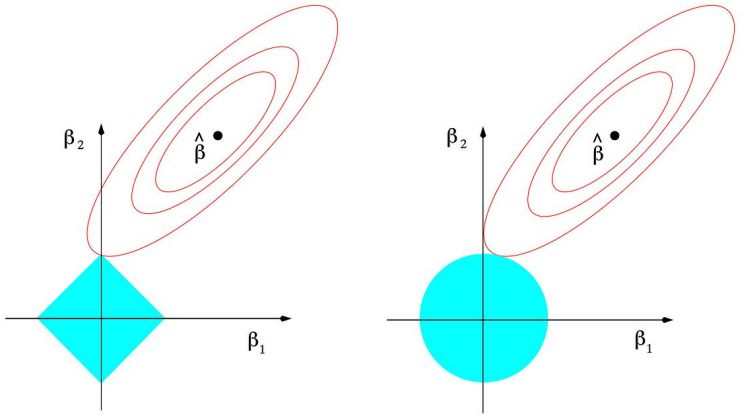
$$\frac{\partial J(\beta)}{\partial \beta} = -2X^T y + 2X^T X\beta + 2\lambda\beta = 0$$

$$(X^T X + \lambda I)\beta = X^T y$$

$$\beta_{ridge} = (X^T X + \lambda I)^{-1} X^T y$$

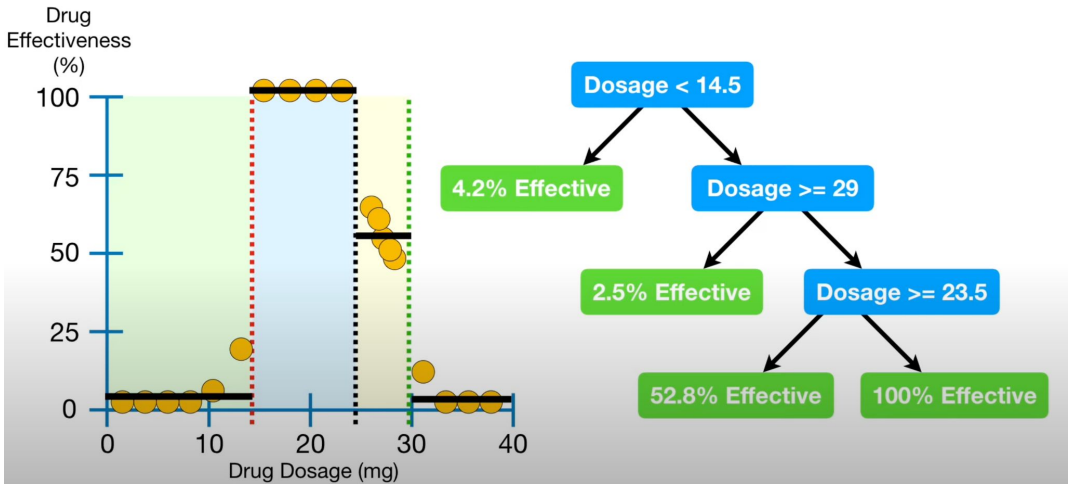
Benefits of Ridge Regression

- It is best to use Ridge Regression when you have many features that are correlated
- Regularization penalizes overly complex models, thus helping the model generalize better to unseen data
  - This can be controlled by modifying the hyperparameter λ
  - Penalty shrinks some coefficients close to zero leading to a reducing complexity
- Ridge is used over Lasso when all features are important



Gradient Boosted Trees

Regression Trees



YouTube. (2019, April 1). *Gradient boost part 2 (of 4): Regression details.* YouTube. https://www.youtube.com/watch?v=2xudPOBz-vs

Fitting a Regression Tree

Input

data $\{(x_i, y_i)\}_{i=1}^{n}$

Algorithm

Until model fits data perfectly or as defined by hyperparameters:

2: **For** all cutoff points:

Compute Ordinary Least Squares (OLS) for each cut-off

Save smallest OLS

3: Split the data into branches at a cut-off that minimizes the OLS

4: Set leaf value to the average of the data points that fall in it

Gradient Boosted Trees

Input

data $\{(x_i, y_i)\}_{i=1}^{n}$ and loss function $L(y, F(x)) = \frac{1}{2}(y - F(x))^2$, learning rate ν

Algorithm

1: Initialise the model: $F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{N} L(y_i, \gamma)$ which is $\frac{1}{n}\sum_{i=1}^{n} y_i$

For m = 0 to M

2: Compute "pseudo-residuals ": $r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}$ = $y_i - F_{m-1}(x_i)$ for $i = 1, \ldots, n$

3: Fit a regression tree r_{jm} on $(x_i, r_{im})_{i=1}^{n}$ with regions R_{jm}

4: Find $\gamma_{jm} = \arg\min_{\gamma} \sum_{x_i \in R_{jm}} L(y_i, F_{m-1}(x_i) + \gamma)$ = $\arg\min_{\gamma} \sum_{x_i \in R_{jm}} \frac{1}{2}(y_i - (F_{m-1}(x_i) + \gamma))^2$ = $\frac{\sum_{x_i \in R_{jm}} (y_i - F_{m-1}(x_i))}{|R_{jm}|}$

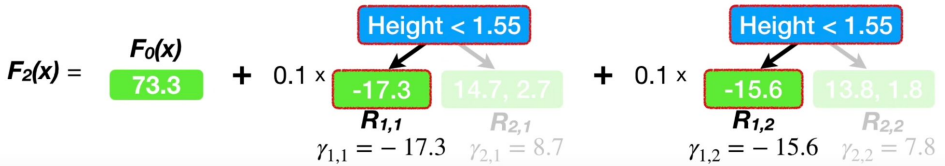
5: Update $F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$

6: Output $F_M(x)$

Gradient Boosted Tree Example

Height (m)	Favorite Color	Gender	Weight (kg)
1.6	Blue	Male	88
1.6	Green	Female	76
1.5	Blue	Female	56

Height (m)	Favorite Color	Gender	Weight (kg)
1.4	Green	Male	???



The **Predicted Weight** = 73.3 + (0.1 × -17.3) + (0.1 × -15.6) = 70

YouTube. (2019, April 1). *Gradient boost part 2 (of 4): Regression details.* YouTube. https://www.youtube.com/watch?v=2xudPOBz-vs

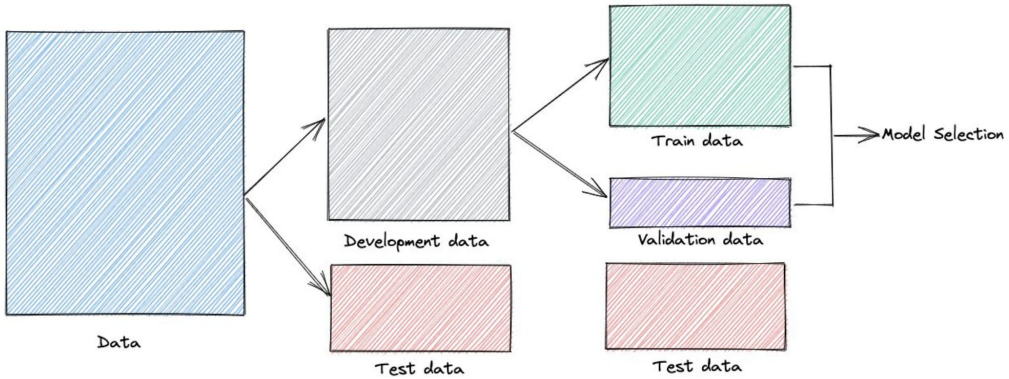
Gradient Boosted Trees vs Linear Regressors

- Modeling
  - Gradient Boosted Trees capture complex and non-linear behaviour
  - Gradient Boosted Trees may also be more prone to overfitting
- Interpretability
  - Feature importance is easier to understand in linear regression

Hyperparameter Tuning

Hyperparameter Tuning

- Tuned hyperparameters (specified parameters that control training process) using grid search
  - Ridge Regression: tuned Lambda hyperparameter with three-way holdout
  - Model selection corresponds to the hyperparameter with the best performance on validation data



Grid Search Using Three-way Holdout

Input:

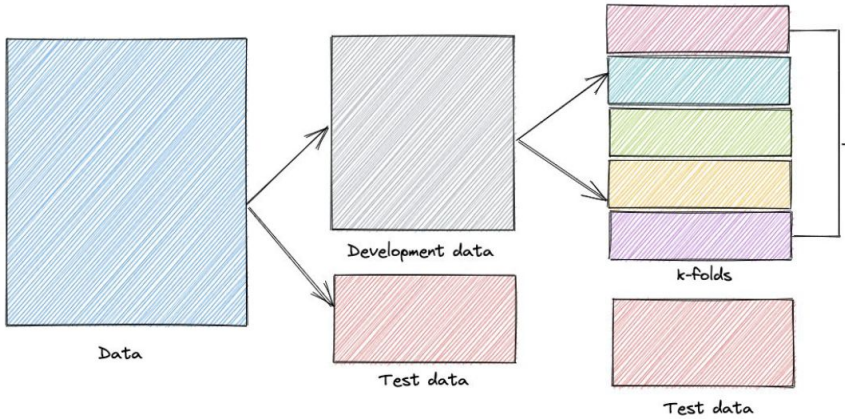
1. Set of hyperparameters: H = {hyperparameter_1, hyperparameter_2, ..., hyperparameter_n}
2. Data

Algorithm:

1. Split the data randomly into three non-overlapping subsets: train_set, validation_set, and test_set.
2. Initialize an empty list to store validation scores: scores = []
3. **For** each hyperparameter_i in the set H:
4. Rain a model using the hyperparameter on the train_set.
5. Calculate the performance score (e.g., R^2) of the model on the validation_set.
6. Append the pair (hyperparameter_i, validation_score) to the scores list.
7. **End for**
8. Determine the best hyperparameter by selecting the one with the highest validation score: best_hyperparameter = argmax(scores)
9. Return the best hyperparameter: best_hyperparameter

Hyperparameter Tuning

- Tuned hyperparameters (specified parameters that control training process) using grid search
  - Gradient Boosted Trees: tuned max depth, number of estimators, and learning rate with k-fold cross-validation
    - Split training/development data into k-folds randomly
    - For every hyperparameter, models is trained using k-1 folds and evaluated on the kth fold
    - Process is repeated for all k-fold and average model performance is calculated
    - Model selection corresponds to hyperparameter with best average performance



Grid Search with K-fold Cross Validation

Input:

1. Data
2. Set of hyperparameter combinations: H = {hyperparameter_1, hyperparameter_2, ..., hyperparameter_n}
3. Number of folds: k

Algorithm:

1. Initialize an empty list to store cross-validation scores for each hyperparameter combination: scores = []
2. Split the training data into k non-overlapping folds randomly
3. **For** each hyperparameter_i in the set H:
4. Initialize an empty list to store the k validation scores for hyperparameter_i: validation_scores = []
5. **For** each fold j in k:
6. Divide the data into training_set and validation_set where: training_set = Union(all folds except fold j), validation_set = fold j
7. Train a model using hyperparameter_i on the training_set.
8. Calculate the performance score (e.g., R^2) of the model on the validation_set.
9. Append the validation score to validation_scores.
10. **End For**
11. Calculate the average validation score for hyperparameter_i: average_score = Mean(validation_scores)
12. Append the pair (hyperparameter_i, average_score) to the scores list.
13. **End For**
14. Determine the hyperparameter combination with the highest average cross-validation score: best_hyperparameter = argmax(scores)
15. Return best_hyperparameter

Model Results

Model Results

- We used R^2 to compare our models
  - $R^2 \in [0,1]$ (typically), where higher values indicate a better model
  - This is because R^2 is inversely associated to the sum of squared errors (SSE↑, R^2↓)
- For iterpretatibility, we also looked at mean absolute error ($\sum_{i=0}^{n} |y_i - x_i\beta_i|$)
- Mean estimator (predicting mean value on all data points)
  - $R^2 = 0$
  - Mean absolute error: 11,658,029.8
- Ridge regression (optimized hyperparameters):
  - R^2: 0.56
  - Mean absolute error: 7,479,570.3
- Gradient boosted trees (optimized hyperparameters):
  - R^2: 0.78
  - Mean absolute error: 4,768,782.4
- Demo: https://colab.research.google.com/drive/1kyFYcuFGBZS116z07frgWn8F-zpbVr8A#scrollTo=Sd8LnWmrbe25

Feature Importance

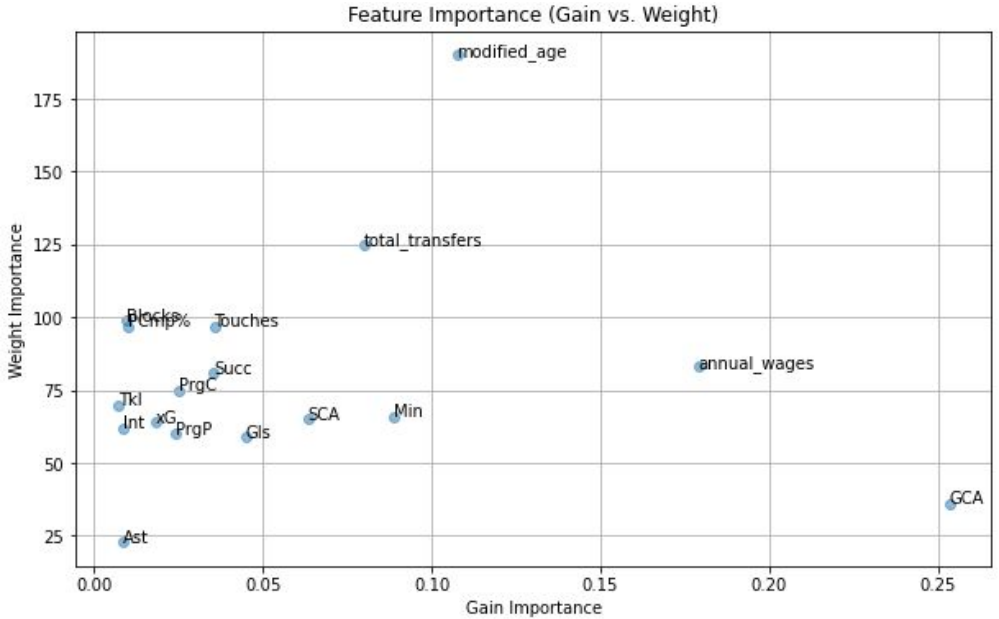
- Gain is defined as the improvement in accuracy from adding a feature to a particular branch

$$\text{Gain} = \sum_{\text{all trees}} (L_{\text{before}} - L_{\text{after}})$$

- Weight is defined as of times a feature occurs in the splits of trees in the model

Feature Importance

- Gradient boosted trees (XGboost) feature importance is given by different metrics



Feature Importance

- Features with less possible values tend to have lower weights, given they will occur in less trees since there are less possibilities. However, they will tend to have higher gain.
- This helps understand why GCA is at the top. GCA has fewer possibilities than most, if not all, other features. However, its place in the top features in terms of gain is surprising.
- Variables like annual_wages, modified_age and total_transfers being in the top 5 passes sanity check

Limitations of our model

- The only marker of player reputations and marketability we used is wages
- Did not include position
- We only used total transfer business to encode the effect of a team in a player's valuation
- We only use match stats for past 2 seasons
- We do not account for national team experience
- We only consider the top 5 European leagues

Future of our predictor

- Since TM is used in negotiations, entities can use personalized algorithms to complement or rebutt discussion that include TM valuations. For example if a club values touches more than TM, they would be able to identify the overestimate in TM and discuss accordingly
- As team transfer spending disparities widen, which has been the trend in recent years, innovation could help medium or small size clubs compete with the rich clubs
    - Brentford's rise thanks to xG

TIPPETT, J. (2019). *The expected goals philosophy*. INDEPENDENTLY PUBLISHED

Our Future

Tomi:

- Will be staying in Columbia for an MS in CS

Amin:

- Will be working on FX Automated Market Making at Goldman Sachs

Blake:

- Undecided - potentially will continue to work in FX Trading

Maximo:

- Will be working as a quantitative strategist at Goldman Sachs