Exploration of Validation Procedures

Eoin Houstoun

Supervisors: Dr. Kathleen O'Sullivan, Dr. John O'Mullane, Dr. Tony Fitzgerald

Second Reader: Dr. Supratik Roy

ST4092

University College Cork

Cork, Ireland

Acknowledgements

I would like to express my sincere gratitude to my supervisors, Kathleen O'Sullivan, John O'Mullane, and Tony Fitzgerald, for their invaluable support, guidance, and feedback throughout this project.

Contents

List of Figures

List of Tables

List of Abbreviations

ML	Machine Learning
PFE	Preliminary Feature Elimination
RFE	Recursive Feature Elimination
IV	Internal Validation
EV	External Validation
IEV	Internal-External Validation
MSE	Mean Square Error
RMSE	Root Mean Square Error
AUC	Area Under the Curve
CV	Cross Validation
LOO CV	Leave One Out Cross Validation
RK-Fold CV	Repeated K-Fold Cross Validation
MC CV	Monte Carlo Cross Validation
FFP	Financial Fair Play
MLR	Multiple Linear Regression

Abstract

Validation in machine learning is the process of training a model and evaluating the model's performance with a test dataset. The aim of this project is to explore validation procedures, with a focus on understanding the different methods and their respective benefits, limitations, and applicability. The study begins with a literature review that dissects the intricacies of various validation methods, distinguishing between internal and external validation techniques and discussing the circumstances under which each is most effectively employed. A pivotal aspect of this research is the examination of the theoretical view of model validation, coupled with practical insights into their execution. These concepts were applied to a specific case study: predicting the transfer fee of footballer players using machine learning. Through this application, the study evaluates the suitability of different validation procedures and their respective methods for the assessment of different model's performance in a real-world scenario. This research not only aims to shed light on the optimal strategies for model validation in predictive modelling but also seeks to contribute to the broader understanding of how these strategies can be tailored to specific contexts, thereby enhancing the accuracy and reliability of machine learning applications in diverse fields.

1 Introduction

1.1 Machine Learning

Validation methods in statistical and machine learning are crucial for ensuring the accuracy and reliability of models. Machine learning (ML) is a branch of study in artificial intelligence concerned with the development and study of statistical algorithms that can learn from data and predict on unseen data (Lista, 2023). It can be broken into 2 main categories, supervised and unsupervised learning.

Supervised learning typically learns a function that maps labelled input data to a dependent variable based on a ML algorithm. The most common supervised tasks are classification, which has a categorical dependent variable, and regression, which has a continuous dependent variable (Kuhn and Johnson, 2013).

Unsupervised learning analyses patterns and characteristics of a dataset. It can uncover insightful correlations without the guidance of labeled outcomes. It is usually employed for investigative and generative analysis and if the outcome variable is discrete, it is considered as clustering. In the case of a discrete outcome variable dimension reduction is used (Lista, 2023). A clear representation of the differences between supervised and unsupervised learning can be seen in Table 1.1 below.

Table 1.1: Supervised vs. Unsupervised Learning

	Supervised Learning	**Unsupervised Learning**
Discrete	Classification	Clustering
Continuous	Regression	Dimension Reduction

In the scope of this project, the focus was specifically on validation in regression as the dependent variable in my case study was the transfer fee of a footballer which is continuous.

A person who builds a prediction model in machine learning is known as a modeller or machine learner and they make the key decisions in the development and validation in machine learning. There are many steps for a modeller to take in the process of machine learning with a general diagram of the statistical learning framework described in Figure 1.1. The machine learning process can be slightly different from project to project but most will incorporate these steps in some form.

Data exploration is crucial in any machine learning process as the foundation of a modeller's decision making relies on strong exploration of the data. Exploring the data involves looking at the dataset size, dimensions, type, distribution, mean, variance and

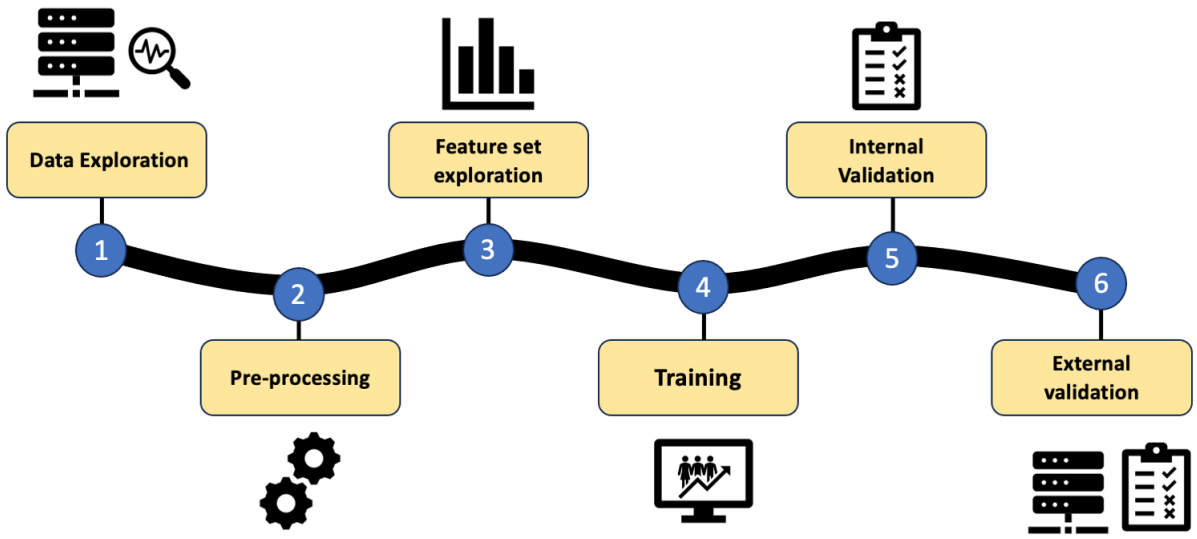


Figure 1.1: Machine Learning Pipeline
A general machine learning pipeline and an indication of where internal validation and external validation take place in the process

missing values, giving an informed decision on the machine learning category you are working with and any initial issues with the data.

Step two in the statistical learning framework is to pre-process the data. This step involves missing data imputation, data scaling, transformations and factor re-encoding (eg one-hot encoding if binary values are required in place of categorical levels). This is a vital step and should always be performed before the following steps.

Once the dataset has been pre-processed, the modeller will explore the features in the dataset. This involves inspecting predictors X with respect to the dependent variable Y. This will inform the machine learner on the correlation and collinearity between variables (i.e if variables tell the same information as each other about the outcome variable, they could be redundant) (Steyerberg, 2009). Based on this analysis, one can perform preliminary feature elimination (PFE) or recursive feature elimination (RFE). PFE is a straightforward approach to reduce the feature space by removing features that do not meet certain criteria before building the model. This can involve discarding features with too many missing values, those with little variance, features that are highly correlated with others (to reduce collinearity), or ones that don't meet a certain threshold of importance according to domain knowledge or simple statistical tests (Cai et al., 2018). RFE is a more systematic and iterative approach to feature selection that involves recursively building a model and choosing the best or worst performing feature (as indicated by the model's coefficients or feature importance scores), setting it aside, and then repeating the process with the rest of the features. This process continues until all features in the dataset have been ranked by their importance, after which the optimal subset of features can be selected for model training. RFE is often used with models that assign importance to features, such as regression models, support vector machines and random forests. RFE is repeated in steps 3, 4 and 5 in the machine learning pipeline

(Figure 1.1). Reducing the feature set can decrease complexity in models and improve predictive ability. After performing steps 1-3, the modeller is now at the stage of internal validation (steps 4 &5).

Steps 4 and 5 are the model building phase and it is here where internal validation is employed. These steps can be done iteratively with different models to see if they give similar results and which model is most applicable/accurate. Training a model is the same as fitting, or developing a model with these three terms being interchangeable. It is generally the term used in machine learning as you are training the machine to learn patterns from the data, which it can then use to make predictions or decisions based on new, unseen data.

Tuning the model through resampling techniques is crucial to finding the optimal parameter values that prevent overfitting and ensure the model generalises well to new, unseen data. Resampling techniques are discussed in detail in Section 2.1.1. Once the parameters are determined, the model is then fitted to the entire training dataset, leveraging the full range of information available to solidify its learning. At this stage, RFE is again employed to reduce the model. By iteratively evaluating and removing the least important variables, RFE hones in on the most important features, thereby improving the model's efficiency and potentially its performance on new data. It is beneficial to perform the training with multiple models so that they can be benchmarked and tested against each other for model selection (Kuhn and Johnson, 2013).

During model development, a modeller should be asking themselves the following questions: Which model should I choose? Is this model accurate or poor at predicting future events? How confident am I with these predictions ? Does the model overfit or underfit the data? Validation is the key to answering these questions and has been discussed in detail throughout this report.

1.2 Validation

Validation in machine learning is the process of training a model and evaluating the model's performance with a test dataset. It stands as a crucial step in the process of ML, as the primary goal of predictive modelling is to predict future events as accurately as possible. Hence, we need to be able to evaluate if a model's predictions are valid or not (Steyerberg, 2009).

Regression and classification models are highly flexible and capable of capturing complex relationships within data. However, they also run the risk of overfitting, where they may learn noise in the training data as if it were a true pattern (Kuhn and Johnson, 2013). The model should not be built to perfectly predict the existing data; instead, its goal should be to effectively predict future, unseen data. However, all the available data should be used to train a model, therefore, a modeller requires a methodological approach to validate these models to assess if it will generalise to unseen data.

Implementing the correct validation procedures and techniques can not only prevent overfitting but also increases the efficiency of model development by guiding you towards

models that are more reliable (Clark, 2004).

Moreover, using different validation procedures can produce disparate conclusions about the data such as "reproducability" and "generalisability". A model is reproducible if the validated results are reliable and consistent among repeated analysis and it can be effectively applied to future unseen data (Park and Hastie, 2007a). A model is generalisable if it's performance is consistent across multiple datasets, tested under different conditions, such as different groups, equipment, time, or environmental factors (Steyerberg, 2009). Hence, throughout this report an important distinction is made between internal validation (IV), external validation (EV) and internal-external validation (IEV) detailing their respective methodologies, offering guidance on their appropriate application, and discussing additional considerations vital for their effective use. This is guided by a thorough literature review, with the relevant methods applied to a specific case study, highlighting their practical significance.

To give the reader an indication on the general differences between the three procedures, the general overview of each validation procedure is displayed in Table 1.2 with a detailed exploration of each method in the next chapter.

Table 1.2: Validation Procedures Overview

Procedure	**Description**	**Purpose**
Internal Validation	Assess model's performance on the dataset it was developed with.	Reproducibility
External Validation	Assess model's performance on independent dataset that was not available during model development.	Generalisability
Internal-External Validation	Leave each "group" out once and assess model's performance on each "group", on the dataset it was developed with.	Generalisability

2 Literature Review

In this chapter, IV, EV and IEV have been explored (Section 2.1). This includes the theoretical foundations and best practices for these methods. IV and EV were then reviewed in tandem (Section 2.2), exploring the optimal times to employ the two methods. Guided by this thorough literature review, a decision was made on the suitable procedure for the case study in this project (Section 2.3).

2.1 Validation Procedures

2.1.1 Internal Validation

Internal validation assesses the model's performance on the dataset it was developed with and it evaluates the model's reproducibility (Austin et al., 2016). It is essential for assessing the performance of machine learning models on unseen data. By systematically partitioning the dataset into training and validation sets, IV helps identify models that generalise well beyond the training data. It enables practitioners to detect and mitigate issues such as overfitting and underfitting, ensuring reliable model performance. Incorporating internal validation into the model development process is crucial for building accurate and trustworthy predictive models when a model is not pre-specified. A general diagram of this procedure is seen in figure 2.1 to aid the reader in understanding the steps involved in the internal validation process.

The "training" data set is the general term for the samples used to develop the model, while the "test" or "validation" data set is used to qualify performance (Kuhn and Johnson, 2013). This diagram is a general approach to internal validation but in reality, each internal validation method has it's own unique way of splitting the data into its training and test set. The reason for this is that excluding data from the development process could omit crucial information and therefore, it is advisable to employ iterative training and testing methods, known as resampling techniques. When a practitioner uses these methods, it is considered rigorous internal validation, which is highly recommended.

In the assessment of predictive model performance, the selection of appropriate performance metrics is to be made. Among these, the Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R^2 are widely utilised for regression models (Sobol, 1991).

The MSE quantifies the average squared difference between the estimated values and the actual value. The MSE is defined by the formula:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{1}$$

where Y_i represents the actual observed outcomes, $\hat{Y}_i$ denotes the predicted values by the model, and n is the total number of observations

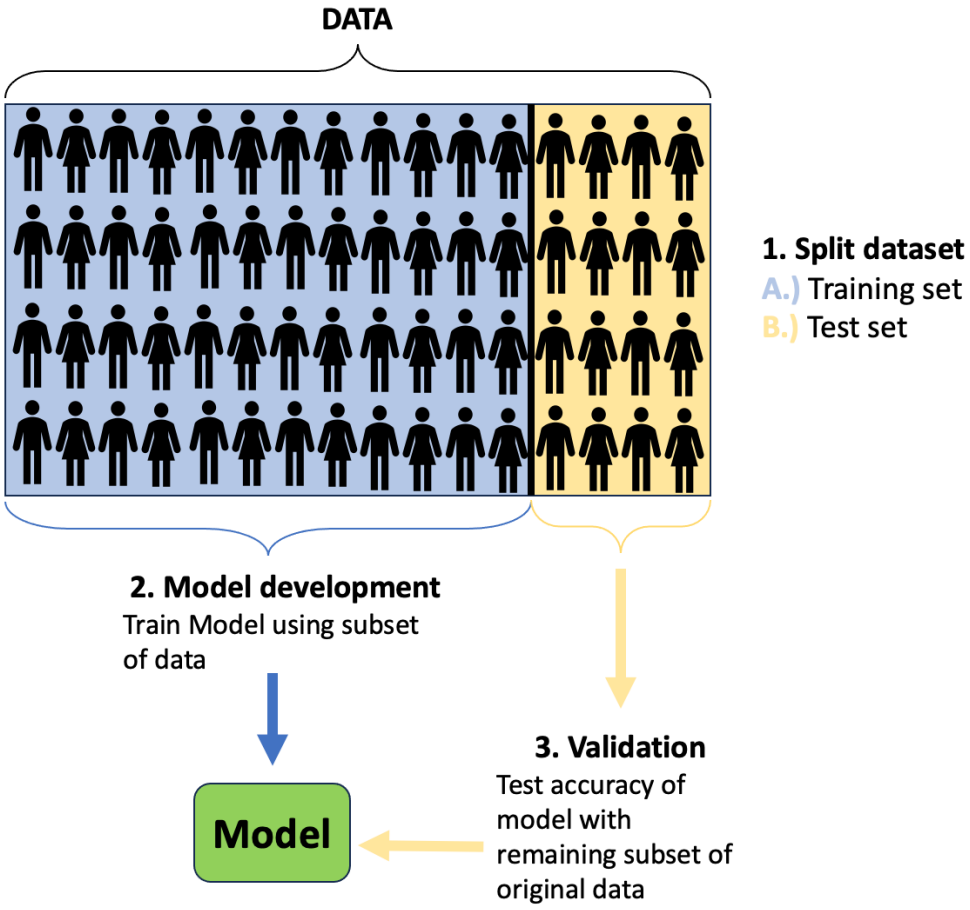


Figure 2.1: Internal Validation
Randomly split the data, fit the model to training data sample, validate with the test set.

RMSE is derived from the MSE, offering a measure of the standard deviation of prediction errors in the same units as the dependent variable. RMSE calculates the square root of the average of squared differences between predicted and actual observations. Its formula is expressed as:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2} \tag{2}$$

Similar to MSE, RMSE provides a scale-dependent accuracy measure that is particularly useful in comparing prediction errors across different datasets or models. Lower values indicate better model performance by showing closer agreement between predicted and observed values. The scale of MSE and RMSE values can vary significantly across different domains and types of data.

These metrics, as highlighted by Kuhn and Johnson, 2013 and Harrell, 2015, play a pivotal role in assessing a model's predictive accuracy. However, their effective application necessitates caution to prevent misinterpretation. Twomey and Smith, 1999 has disccused that there is no consensus as to which measure should be reported, and thus

direct comparisons among techniques and results of different researchers is practically impossible.

A fundamental aspect of an accurate performance evaluation lies in the utilisation of test data for the calculation of MSE and RMSE. Rather than relying on training or apparent data. This approach ensures that the assessment provides an unbiased view of its predictive performance. Analysing these metrics from both training and test datasets allows for a comprehensive evaluation, revealing potential overfitting issues. Overfitting is indicated by a model demonstrating excellent performance on the training data but significantly worse on the test data, suggesting that it may have learned the noise in the training set rather than the underlying data structure.

Another metric often used in regression is R^2. There are different ways of calculating R^2 but the most common one quantifies the proportion of the information in the data that is explained by the model (Sobol, 1991). It is a statistical measure of how well the regression predictions approximate the real data points. An R^2 of 1 indicates that the regression predictions perfectly fit the data. The formula for R^2 is:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{n}(Y_i - \bar{Y})^2} \tag{3}$$

where $\bar{Y}$ is the mean of the observed data.

R^2 is widely used in the context of linear regression models to assess the explanatory power of the model and measure the "goodness of fit" (Kuhn and Johnson, 2013).

Classification problems use different performance criteria to validate a model. An essential tool for evaluating classification models is the confusion matrix, which helps to visualise the performance of an algorithm. It shows how many predictions are correct and incorrect per class.

One of the key rates derived from this matrix is the misclassification rate, which quantifies the proportion of incorrect predictions and can be expressed as:

$$\text{Misclassification Rate} = \frac{FP + FN}{TP + TN + FP + FN} \tag{4}$$

where TP, TN, FP, and FN represent the true positives, true negatives, false positives, and false negatives, respectively Demir, 2022.

Following this, an exploration of sensitivity and specificity provides insight into the model's accuracy in identifying positive and negative cases. Sensitivity, or true positive rate, measures the proportion of actual positives correctly identified by the model, while specificity, or true negative rate, measures the proportion of actual negatives correctly identified:

$$\text{Sensitivity (Recall)} = \frac{TP}{TP + FN} \tag{5}$$

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{6}$$

These metrics are foundational for understanding the behavior of classification models under various thresholds, which can be used to plot an Area Under the Curve (AUC) plot. The AUC provides a single value that summarizes the overall ability of the model to discriminate between positive and negative classes across all possible classification thresholds.

These classification metrics are important for validating model performance and provide a framework for assessing classification algorithms. For a deeper understanding of these metrics, "An introduction to statistical learning: With applications in R" **stat** offers comprehensive coverage of these topics. The focus of this project's case study was on a regression problem, which is why regression metrics have been predominantly discussed.

Internal Validation Methods

There are many methods used in IV that assess the performance differently and it would be unwise for a modeller to choose a method at random without knowing which method is suitable for their model and data.

Apparent Validation

One method used is apparent validation. This is the simplest form of internal validation and is highly optimistic. It is assessment of the model's performance on the sample where the model was developed from (Steyerberg, 2009). In other words, the model is trained on 100% of the available data and then is tested on the same data. This method is prone to overfitting. The test is not on unseen data so you can not gauage the reproducibility of the model. Since the model parameters were optimised for this specific sample it is likely that this is a biased assessment. As a result, when performing IV, a more rigorous technique is advised to ensure a more accurate and unbiased evaluation of the model's capabilities. This method is not viable for small to medium-sized datasets. However, for very large datasets, it is less likely for overfitting to be present, so the results could be a reliable indicator of performance.

Split-Sample Validation

Another IV technique is split-sample validation (data splitting). This method involves randomly splitting the dataset the training sample and then validated on the test sample once (Ramspek et al., 2020). The test data is unseen at the development stage and therefore won't have the severity of the issue that apparent validation had when it comes to overfitting. The two most typical split ratios made are 50%:50% or 67%:33% (Steyerberg EW, 2015). There are numerous reasons as to why split-sample validation is not advised. Using either of these split ratios will result in you losing out on a lot of data during model development. Due to the randomness, there is an element of luck in how the model performs. It's possible that the model's accuracy may not fully reflect its true capability due to the specific data split used for evaluation. This split could either underestimate or overestimate the model's actual performance. A large data set is required to make split-sample validation feasible, however, apparent validation will already be a good indicator of model performance if N is large enough. Hence, split-sample is a method that only works when not needed (Harrell, 2015). This method is

also known as hold-out validation. Resampling techniques were introduced to apply this approach in a more methodical manner.

Resampling techniques

Split sample validation can be improved upon by incorporating resampling techniques. Resampling is a statistical technique used to validate models by repeatedly drawing samples from a dataset. They provide more accurate performance estimation due to them handling issues like bias and variance better (Kuhn and Johnson, 2013). In this context, bias refers to the degree to which the model oversimplifies the underlying patterns in the data, leading to systematic errors, while variance indicates the model's sensitivity to small fluctuations in the training data, potentially resulting in overfitting.

Resampling techniques, are employed to achieve a bias-variance trade-off by systematically assessing model performance across different subsets of the data. They enable practitioners to evaluate how well the model generalises to unseen data by iteratively training and testing on different subsets of the data, providing insights into the optimal balance between bias and variance for improved model performance.

The two main methods used for resampling are cross-validation (CV) and the bootstrap. In general, CV refers to when a model is developed on part of the data (training sample) and assessed on the model's performance on the remaining sample (test set). This process is repeated, so that every subset of the data was tested on once. The mean of the performance is the value evaluated (Kohavi, 1995). There are many types of CV and with slight differences and limitations. The bootstrap is a statistical method introduced by Bradley Efron in the 1970's, that involves repeatedly sampling from a data set with replacement to estimate properties or accuracy of a model. Replacement, in this context means that once a data point is chosen, it remains eligible for subsequent selection. Below, I discuss the implications and applications of these two resampling techniques, along with their different variants.

K-Fold Cross Validation

K-Fold CV is when the data is split into k subsets or folds, with the model trained on k-1 folds and tested on the remaining fold. Each fold is tested once, with the accuracy being the mean of the k performances. None of the data that was in the training data is included in the corresponding test fold $Data_{\mathsf{TRAIN}} \cap Data_{\mathsf{TEST}} = \varnothing$ (Berrar, 2018). The choice of k is up to the modeller but 10 is the most common value of k used in the literature because it provides a good balance between reducing the bias associated with the validation error estimate and decreasing the variance of the estimate (Harrell, 2015).

Using 10-fold CV results in 90% of the data used for developing the model and the remaining 10% is used for validation in each fold. This process returns ten error rates. Ideally, these values should be consistent across all folds, indicating that the model generalises well to unseen data, as it demonstrates similar performance regardless of the data subset it is tested on.

For lower values of k (such as k=2 or k=5), it tends to lead to estimates with higher bias,

as the validation set is smaller and less representative of the overall dataset. Bias may be introduced depending on how the data is partitioned. If the split disproportionately favors certain subjects of the data, it can lead to biased performance estimates. As k increases, the difference in size between the training set and the resampling subsets decreases. Hence, the bias of k-fold CV is smaller for larger k (Kuhn and Johnson, 2013). However, increasing the number of folds, k, in k-fold CV as much as computationally possible is not advisable due to the risk of high variance. While a higher k reduces bias, it leads to increased variance. This is because the results become highly sensitive to the particular samples included in each fold. High variance introduces greater uncertainty in the model's estimated performance. For medium to large datasets, k-fold CV remains advantageous due to its relative computational efficiency and methodical approach to breaking up the data into subsets so all the data gets tested once. However, there are better methods to use when the dataset is small, as k-fold cross-validation may not perform optimally with limited data. The smaller sample size in each fold can lead to less reliable and more variable estimates of model performance, potentially impacting the model's ability to generalise effectively.

The cross-validation MSE for k-fold cross-validation on the training set is given by the equation:

$$CV_{\text{train}} = \frac{1}{K} \sum_{k=1}^{K} MSE_{k,\text{train}} \tag{7}$$

and for the test set:

$$CV_{\text{test}} = \frac{1}{K} \sum_{k=1}^{K} MSE_{k,\text{test}} \tag{8}$$

where CV_{train} is the cross-validation error over K folds on the training set, and CV_{test} is the cross-validation error over K folds on the test set. $MSE_{k,\text{train}}$ and $MSE_{k,\text{test}}$ are the Mean Squared Errors for the k-th fold for the training and test sets respectively (James et al., 2022). To asses the result, the average of the k RMSE's is the error rate of the model as can be seen in the formula.

Leave-One-Out Cross Validation

The most extreme case of k-fold CV is called Leave-One-Out CV (LOO CV) where k $=$ n. Here, each individual case is held out once for testing. This is equivalent to the jackknife technique in CV (Harrell, 2015). Each data point represents the entire test sample once. However, this method is computationally intense and not appropriate for large datasets. Even with small datasets, model overfitting can be present with a high variance because $(n-1)/n$ of the data is used for development at each iteration with only $1/n$ being used for validation (Efron and Gong, 1983). Efron reported that there are more accurate techniques that are less computationally expensive but recommends this approach if an only if the data sample is extremely small ($n < 100$) and the model is stable, meaning that small changes in the data do not lead to large changes in the model. Molinaro's research revealed that leave-one-out and k-fold CV produced comparable results, suggesting that k $= 10$ is more advantageous due to computational efficiency (Molinaro et al., 2005).

The cross-validation MSE for LOO CV on the training set is given by the equation:

$$LOOCV_{\text{train}} = \frac{1}{N}\sum_{n=1}^{N} MSE_{n,\text{train}} \tag{9}$$

and for the test set:

$$LOOCV_{\text{test}} = \frac{1}{N}\sum_{n=1}^{N} MSE_{n,\text{test}} \tag{10}$$

Repeated K-Fold Cross Validation

As discussed, k-fold CV has its limitations with small datasets. However, by repeating k-fold CV, you can improve the accuracy of the k-fold process (Kohavi, 1995). By repeating the process, it helps to smooth out the variability in the estimation of the model's performance that might result from any single split of the data. This repetition provides a comprehensive evaluation, reducing the impact of random variations in the data split, which is crucial in scenarios where the number of data points is relatively limited and each data point's influence is significant.

The cross-validation MSE for repeated k-fold CV on the training set is given by the equation:

$$\text{Repeated } CV_{\text{train}} = \frac{1}{KR}\sum_{r=1}^{R}\sum_{k=1}^{K} MSE_{k,r,\text{train}} \tag{11}$$

and for the test set:

$$\text{Repeated } CV_{\text{test}} = \frac{1}{KR}\sum_{r=1}^{R}\sum_{k=1}^{K} MSE_{k,r,\text{test}} \tag{12}$$

At each iteration of the outer loop, the dataset is shuffled randomly before folds are created and k-fold CV is performed. Storing (RXK) estimates of accuracy in repeated k-fold cross-validation accounts for variability in performance due to both random sampling of data into folds and random assignment of folds into training and testing sets in each iteration, providing a fair evaluation of the model's performance.

Monte Carlo Cross Validation

Monte Carlo cross-Validation (MC CV), also known as random subsampling validation, randomly divides the complete dataset into training and testing sets M times, shuffling the data at each iteration. This process contrasts with k-fold cross-validation systematic partitioning of the dataset into mutually exclusive folds. The flexibility of MC CV allows for arbitrary division ratios between training and testing sets, which can be advantageous when the dataset size or distribution does not conveniently allow for equal partitioning as required in k-fold (Xu and Liang, 2001).

The major drawback of this method, is that it can lead to data overlap between training and testing sets across different iterations because of the shuffle at each iteration, potentially causing optimistic bias in the performance estimates. Using a sufficiently large number of iterations (M) increases the likelihood that all data points are included

in the test set at some point, thereby reducing the chances of bias in the model evaluation. The idea is that by increasing the number of iterations, you can more evenly distribute the data across the training and testing set. Given its characteristics, MC CV is recommended for larger datasets where the risk of significant overlap is minimised and when there is a need for quick model prototyping where computation resources are not a major constraint. It is particularly useful in studies where robustness against a variety of training conditions must be assured (Molinaro et al., 2005).

The MC Cross-Validation for the training and test set is defined as:

$$\text{MC } CV_{\text{train}} = \frac{1}{M} \sum_{m=1}^{M} MSE_{m,\text{train}} \tag{13}$$

$$\text{MC } CV_{\text{test}} = \frac{1}{M} \sum_{m=1}^{M} MSE_{m,\text{test}} \tag{14}$$

where M is the number of resamples. $M = 1000$ is typically enough iterations to assess the performance.

This method is frequently overlooked due to its similarities with the bootstrap method, which resamples with replacement and has a fixed train-test split. As detailed in the section below on the bootstrap, it is preferred over MC CV method because there is no data overlap the assessment of the error. Having said this, MC CV is flexible (train:test split), so, it is recommended when a practitioner wants to set their own split on a large dataset.

Nested Cross Validation

Nested cross-validation is a method for tuning hyperparameters while ensuring unbiased performance evaluation. It contains a nested for-loop and uses k-fold CV on the inner and outer loop. The inner loop is for hyperparameter tuning and an outer loop for assessing model performance. Within the inner loop, different hyperparameter settings are tested, and the setting that has the lowest cross-validated error is selected. The model then uses these optimal hyperparameters for training across the entire training set specified by the outer loop. Finally, the model is evaluated on the independent test set of the outer loop, ensuring a thorough and unbiased assessment of its performance (Wainer and Cawley, 2021). This methodical approach effectively separates hyperparameter selection from model evaluation.

The key advantage of nested cross-validation is that it provides a more accurate assessment of a model's predictive power because it effectively separates the model tuning and performance evaluation processes. This method is particularly valuable in scenarios where hyperparameter settings significantly impact the model's performance.

It is to be used when there are hyperparameters to be tuned. Its most common use is for regularisation of a regression model. These models include LASSO (Least Absolute Shrinkage and Selection Operator), ridge regression and elastic nets (Park and Hastie,

2007b). All three models include a hyperparameter, λ in their formulas. This hyperparameter controls the strength of the regularisation applied and requires careful tuning to optimise the balance between model complexity and prediction accuracy.

The Bootstrap

"While statistics offers no magic pill for quantitative scientific investigations, the bootstrap is the best statistical pain reliever ever produced"

— Xiao-Li Meng, Whipple V. N. Jones Professor of Statistics at Harvard University.

The bootstrap as described above under resampling techniques, is a widely used approach in validation. Each bootstrap sample maintains the same size as the original dataset and consequently, certain data points may appear multiple times within the bootstrap sample, whereas others might not be chosen at all. Due to the resampling with replacement, error rates derived from bootstrapping typically result in higher precision than CV for fewer iterations (Efron and Gong, 1983).

During a single iteration of bootstrap resampling, a model is constructed using the chosen samples and then applied to make predictions on the samples that were not selected. The samples that are not chosen are referred to as "Out-Of-Bag" (OOB) samples. 63.2% of the data points in the bootstrap sample are represented at least once, known as our "In-Bag" samples. The reason that the split converges to this is provided below:

Given a sample $\{X_1, \ldots, X_N\}$, the probability of not selecting a data point X_i in a single bootstrap resample X^* is

$$P(X_i \text{ not in } X^*) = \left(1 - \frac{1}{N}\right)^N$$

Since there is a $\frac{1}{N}$ chance that $X_j^* = X_i$ for $j = 1, \ldots, N$ and thus a $1 - \frac{1}{N}$ chance that $X_j^* \neq X_i$.

The asymptotic value of this probability:

$$\lim_{N \to \infty} \left(1 - \frac{1}{N}\right)^N = 0.368$$

Hence,

$$P(X_i \text{ is in } X^* \text{ at least once}) = 1 - 0.368 = \mathbf{.632}$$

The split of "In-Bag" and OOB samples averages at 63.2%:36.8%, and because the resampling is done with replacement there is potential for certain samples to be over represented in the training set (Kuhn and Johnson, 2013).

While there are numerous formulas and statistical methods applicable to bootstrapping, our focus here is primarily on utilising bootstrapping for validation purposes and so for

the purpose of validation, the following formulas are used. The training or "In-Bag" error assessment in bootstrap methods is calculated by:

$$Bootstrap_{\text{train}} = \frac{1}{B} \sum_{b=1}^{B} \text{MSE}_{b,\text{In-Bag}} \tag{15}$$

The test or OOB error assessment in bootstrap methods is calculated by:

$$Bootstrap_{\text{test}} = \frac{1}{B} \sum_{b=1}^{B} \text{MSE}_{b,\text{OOB}} \tag{16}$$

where B is the number of bootstrap samples.

To address the inherent bias in simple bootstrap procedures, several refinements have been developed. One notable enhancement is the "632 method", introduced by Efron (Efron and Tibshirani, 1997). This approach calculates the performance estimate by weighting the simple bootstrap estimate at 0.632 and the apparent error rate at 0.368. (0.632 $\times$ simple bootstrap estimate) + (0.368 $\times$ apparent error rate). Kuhn has suggested that this technique offers a more balanced estimate by reducing bias, particularly in evaluating classification models characterised by their error rates. However, while the "632 method" mitigates bias, it may yield unstable results with smaller sample sizes. Furthermore, it risks presenting overly optimistic evaluations for models that significantly overfit the data, as the apparent error rate, in these cases, tends to approach zero, hence, the standard bootstrap is generally preferred (Kuhn and Johnson, 2013).

Bootstrapping is most commonly used for ensemble models like random forest and gradient boosting models. This is because it is integral to bagging (bootstrap aggregating) models. Bagging involves training multiple versions of a model on different subsets of the training data, sampled with replacement, and then averaging the outputs to stabilise the predictions. This is effective in reducing variance and avoiding overfitting (Genuer et al., 2008).

As well as the OOB assessment for validation, the bootstrap will give information about the shape, centre, and spread of the sampling distribution of the statistic.

The overview table 2.1 offers a concise summary of the various internal validation methods, detailing each approach along with specific recommendations for their use. This table serves as an efficient reference to quickly compare and understand the different techniques available for model validation.

Table 2.1: Internal Validation Methods

Method	Description	Recommendation
Apparent	Assessing model performance on the sample where the model was developed from	Biased results, strong chance of overfitting. Accompany with an alternative method.
Split-sample	Randomly split data into training sample, and test sample. Typical split (67%:33%)	Only works when not needed. Choose an alternative method.
K-Fold CV	Data is split into k subsets/folds, with the model trained on k-1 folds and tested on the remaining fold. Each fold is tested once	Can have high variance and bias. Can choose method if N is large but R.k-fold CV and bootstrap is preferred otherwise.
LOO CV	Same as k-fold, where K $=$ N, each data point is tested once.	Computationally expensive. Only if dataset is extremely small and model is stable, use this method.
Repeated K-Fold CV.	K-Fold CV repeated R times.	Can choose this method with a small dataset.
MC CV	Randomly divides the complete dataset into training and test sets M times, shuffling the data at each iteration.	Use only on large datasets when you want to set the exact split of train and testing data
Nested CV	Nested for-loop to tune hyperparameters	Use this method for models with hyperparamters
Bootstrap	Randomly drawn subset of data, selection is done with replacement. Tested on "OOB" sample.	This method can always be considered. Definitely use if N is small

Bootstrap versus Cross Validation

The choice between using bootstrap and cross-validation (CV) often depends on the size of the dataset and the model choice. Bootstrap is typically preferred for small datasets because it excels at estimating both the bias and variance of a model. It maximises the utilisation of all available data and provides estimates of model performance by simulating how the model might perform under varied conditions, mimicking a larger dataset scenario.

On the other hand, cross-validation, especially k-fold CV, is more often used for medium to large datasets. All the data is used for both training and validation, making efficient use of larger data volumes. Additionally, k-fold cross-validation generally is less computationally intense compared to the bootstrap since it does not require creating numerous new datasets through resampling. This makes it particularly suitable for larger datasets where managing computational resources becomes important Berrar, 2018. Also, fine-tuning hyperparameters with cross-validation is preferred over bootstrap because it provides a better estimate of model performance on unseen data by systematically testing the model across different data subsets.

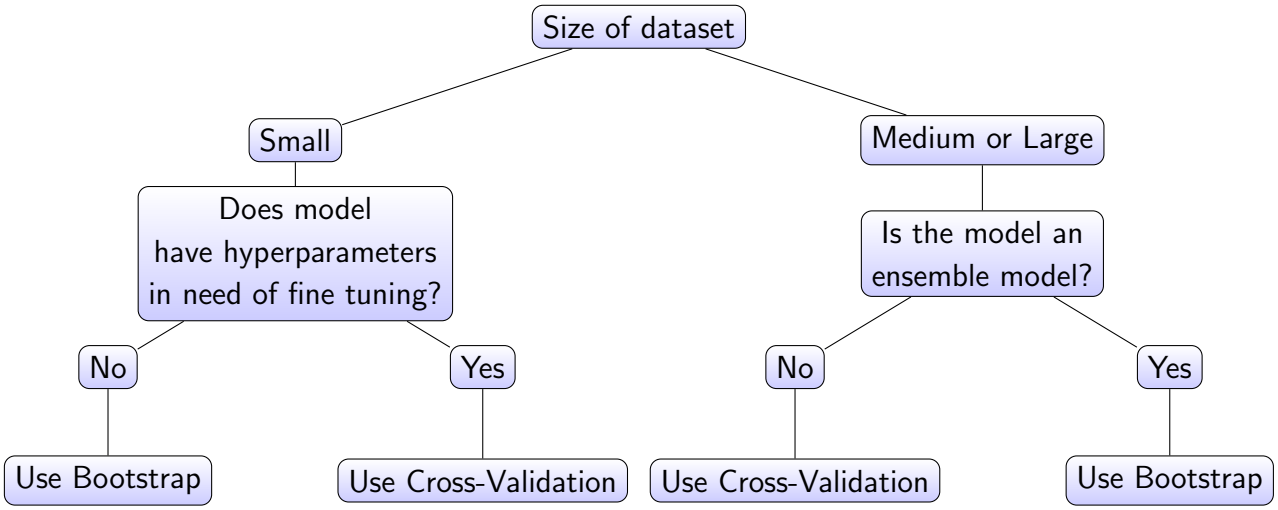


Figure 2.2: Cross-validation versus Bootstrap
Simple decision tree to decide which method is most suitable

2.1.2 External Validation

External validation assesses the model's performance on an independent dataset that was not available during model development (Ramspek et al., 2020). This procedure evaluates the model's "generalisability". If the external data set is very similar to the development data set, the assessment is for reproducibility rather than for generalisability (Steyerberg EW, 2015). A general diagram of this procedure is seen in figure 2.3. Unless both sample sizes are huge, external validation can be low precision (Harrell, 2015).

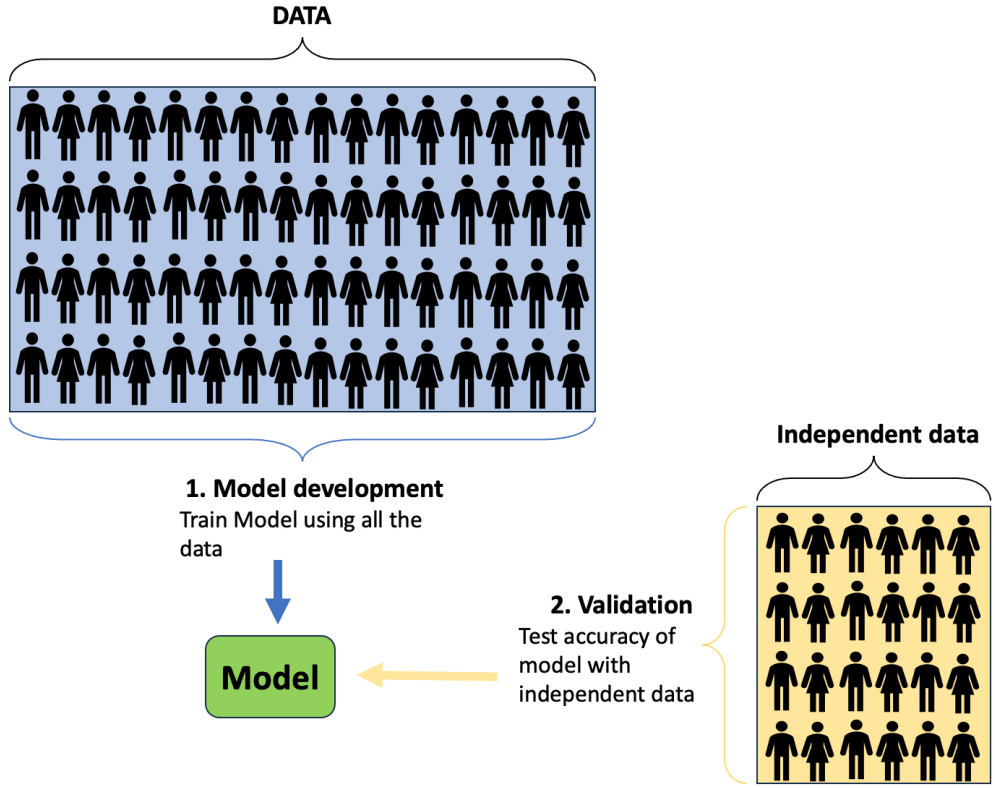


Figure 2.3: External Validation
Validate model on an external, independent dataset that is not available during model development.

When a model is fully pre-specified, EV is used to validate the model and its estimated coefficients, otherwise, IV is used to develop and validate a model (Harrell, 2015). It is used for demonstrating the widespread applicability of a prediction model.

Practitioners often set aside a portion of data during model development and then test the model on this dataset, referring to this process as EV. However, this is a misconception. For true external validation, the dataset should not have been available at the time of model development. Ideally, all available data should be used to train the model, and then the model should be tested on a new dataset that was not available during the model's creation.

Unlike IV, which relies on randomly dividing samples between development and testing phases, EV uses groups who differ in certain aspects from those used to develop the model. This approach can be broken into three main categories: temporal, geographical

and fully independent (Steyerberg EW, 2015).

Temporal Validation

Temporal validation (TV) examines a model's effectiveness across different time periods. It entails validating the model on a new dataset who was included in the same study as original dataset in which the model was developed but sampled at a later time point (Ramspek et al., 2020). It provides information on both the reproducibility and generalisability of a model. In the context of temporal validation, a model developed at a later time point represents the distinct "group" required for external validation, thereby addressing concerns related to generalisability.

It can be a critical aspect of model testing, especially when dealing with time-series data. The need for this form of validation arises primarily to avoid overfitting. Traditional random splitting of data into training and test sets may not be suitable for time-series data since it can lead to models learning patterns that are too specific to the training period and not generalisable to future periods (Austin et al., 2016). Therefore, TV offers a more realistic evaluation by training on past data and testing on future data, which mirrors the real-world scenario where models are expected to make predictions for future events.

Furthermore, TV is instrumental in accounting for trends and seasonality, enhancing the model's ability to predict based on these temporal dynamics. Such validation is not only crucial in finance, where stock market predictions rely heavily on historical data (Jerez and Kristjanpoller, 2020) but also in sports science, where models predict future sporting outcomes based on past games (Horvat et al., 2020).

Despite its advantages, TV presents challenges, including data drift, where the underlying process generating the data may change over time, causing historical data to become less representative of the current or future state. Another challenge is the limited data available for very recent models or applications, which may restrict the effectiveness of temporal validation.

Overall, TV ensures that the predictive performance is not only based on historical data but is also applicable and reliable for future scenarios.

Geographical Validation

Geographical validation tests the model with data from different regions to the one the model was developed with (Austin et al., 2016). It is particularly used in fields like medicine, sociology, psychology, and marketing. This method involves assessing the generalisability of a model across diverse geographical locations, ensuring that research findings are not confined to a specific region but can be applied more broadly.

As an example, suppose a company develops a predictive model for customer behavior based on data from one region. To test its generalisability, they gather data from various regions or countries and validate the model. This process assesses the model's predictions to see if they are consistent across different geographic contexts, allowing businesses to make informed decisions tailored to diverse markets.

Consistent results across regions indicate the reliability and generalisability of the model. Additionally, geographical validation helps identify regional effects influencing the phenomenon under study, such as environmental factors specific to certain areas.

Fully Independent Validation

Fully independent Validation is when a model is validated by other investigators at other sites (Steyerberg EW, 2015). This validation involves utilising diverse data sources, such as insurance claims instead of patient records. By subjecting the model by external parties and different data types, fully independent validation serves as the most rigorous test that can be performed.

The resources needed for this method are expensive and one should complete strong IV before even considering this method. It has been used in the past when it is expected that the model developers may have biases influencing their outcomes.

The overview table of these methods is provided in the Table 2.2 below for future reference.

Table 2.2: External Validation Methods

Method	Description	Recommendation
Temporal Validation	Validate the model on a new dataset which was included in the same study as original dataset but sampled at a later time point	After performing IV, perform if large dataset becomes available to ensure that a model remains relevant and accurate in the face of evolving trends
Geographical Validation	Tests the model with people from different locations, such as various regions.	After performing IV, perform if large dataset becomes available and the model has reason to test for generalisability
Independent Validation	When a model is validated by other investigators at other sites	Reserved for situations where there is concern about potential bias from the model developers during internal validation, ensuring the model's integrity.

2.1.3 Internal-External Validation

In an attempt to examine generalisability without the use of an independent dataset, internal-external validation has been employed (Takada et al., 2021), (Royston et al., 2004). To perform IEV, by use of cross validation, each "group" is left out of development once and then assess the model's performance on the "group" left out. Iterate

process so that each group is left out once. Since the split made on the dataset was not random it is not considered IV (Steyerberg EW, 2015). IEV has similarities to EV, however since the validation set is available at time of development, it is not true EV. A general diagram of this procedure is seen in figure 2.4.

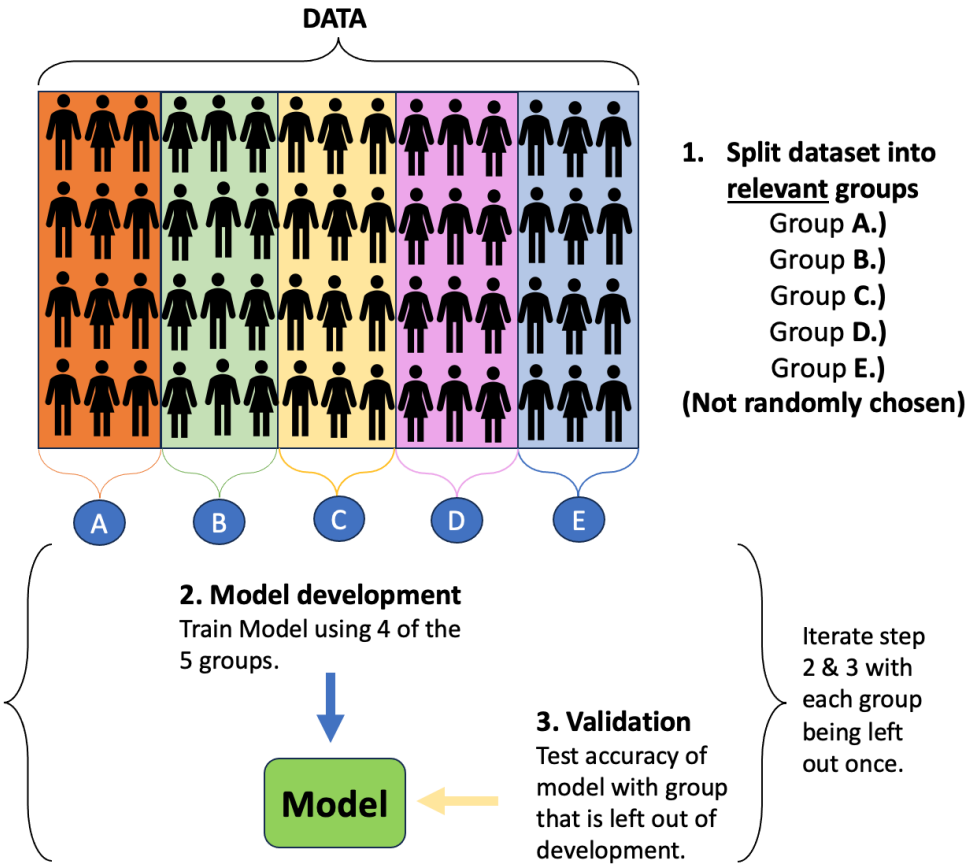


Figure 2.4: Internal-External Validation
Special type of CV where one "group" is left out of development and validated on. Iterate with each "group" left out once.

As an example, suppose a model is developed based on data collected from different hospitals. To assess its generalisability, researchers gather data from a different hospital and validate the model using this new dataset. This process, involves testing the model's performance across distinct hospitals to determine its generalisability. By comparing the model's predictions or outcomes between different hospitals, researchers can evaluate its suitability for broader contexts. This is done under the assumption that, hospital is not used as a predictor in the model.

While including hospital as a predictor in the model can capture some of the variability associated with different hospital settings, it may not fully address the need to assess generalisability across distinct institutions. Utilising hospital as a predictor treats all hospitals as interchangeable, potentially overlooking unique characteristics that could influence model performance. Conversely, performing separate testing on data from different hospitals via internal-external validation allows for a direct assessment of the model's performance in diverse settings, ensuring robustness and applicability across a range of contexts, which may be particularly crucial in fields like healthcare where in-

stitutional differences can significantly impact outcomes. Therefore, the choice between including hospital as a predictor or conducting separate testing depends on the research goals, with IEV offering a more comprehensive evaluation of generalisability in certain circumstances.This paper by Mitchell et al. (2021) provides a comprehensive analysis of generalisability issues in healthcare machine learning, offering insights and recommendations for future research in this area (Mitchell et al., 2021).

Out-of-time validation uses this same method in predictive modeling as explained above but instead of splitting by a "group", it is split by a certain time frame. It is used to test the model's ability to perform effectively on data that happened at a different time period than the one used for training. This approach helps assess the model's generalisability over time, especially in scenarios where patterns in the data may change due to evolving trends, seasonal variations, or other temporal factors. It is a strong replacement to the EV method temporal validation as it mimics this method without the requirement for an external dataset.

Out-of-time validation is particularly suitable for sports science data due to the evolving trends over time, which is well-documented in the literature Aydemir et al., 2023.

The overview table of these methods is provided in the Table 2.3 below for future reference.

Table 2.3: Internal-External Validation Methods

Method	**Description**	**Recommendation**
Internal-External CV	Leave each "group" out once and assess model's performance on each "group", on the dataset it was developed with.	Use this method to test for generalisability before employing external validation
Out-Of-Time CV	Cross validated in the same way but instead of a group being left out, it is a time period.	Use this method to test for the generalisability over time before employing temporal validation

2.2 Choosing Internal vs. External Validation

EV is often favored by non-statisticians, however these practitioners should be made aware of the capabilities of IV and IEV. IV is essential and should always be performed, as it incurs no additional resource costs beyond computation. Rigorous internal validation can sufficiently demonstrate a model's reliability, potentially mitigating the need for external validation and thus saving both time and resources. Once IV has been performed, if the model requires a test for generalisability, employ IEV.

After completing these steps, the model may be considered sufficiently validated for use, contingent, of course, on the outcomes of those validation methods. Should a sufficiently large external dataset become available after deployment, it would be prudent to use it for further validation.

However, EV is often problematic. If the model hasn't been predefined and involves feature selection based on the data, it's important to recognise that the chosen features may not be consistent. In this case, EV might merely affirm the performance of a model that's specific to the sample at hand rather than universally applicable. Rigorous IV serves to confirm the integrity of the model building process, capturing the variability inherent in feature selection. This approach is also recommended when neither the potential EV set nor the training set is sufficiently large.

EV becomes pivotal when there are differences in measurement tools. To ensure that a model built on one specific system holds true across others. This step is important for confirming a model's wider applicability (Harrell, 2015). Another instance calling for EV is when the data intended for validation was not part of the original model development and IV processes. Additionally, if there's any concern about the objectivity or thoroughness of the model developers' IV, seeking an EV becomes a necessary course of action for unbiased verification.

2.3 Validating the Predictions of Football Player's Transfer Fee in Machine Learning

After the thorough exploration of the validation procedures, a decision was to be made on which procedures is appropriate for my own case study: Predicting football player's transfer fee using machine learning.

The answer is that internal validation is crucial, temporal validation is optional and all other forms of external validation are not appropriate in this project.

The reason internal validation is vital is because the dataset is relatively small and therefore rigorous internal validation has to be performed to confidently validate the prediction accuracy of the model. A full exploration of the internal validation methods discussed in the literature review were employed with a discussion on the suitability of each method.

Temporal validation would be useful, as financial markets are often time sensitive, which means that the transfer fee's would likely increase over time due to inflation. The article written by (Poli et al., 2023) has stated that the prices of football players have increased annually by 9.0% over the past decade. The whole idea is to be able to predict future transfer fee's so it is important that the predictions have a way to incorporate inflation. When the data becomes available in the future, temporal validation could be used to validate the model's predictions.

The other external validation techniques are not appropriate in this project because the data and model will not be generalisable. The main reason for this, is that model is predicting transfers into Europe's top five leagues. Therefore the predictions would not be appropriate across other leagues and therefore is not generalisable. The leagues have different financial strengths and weaknesses so the transfer fee of player's is therefore heavily dependent on which league they are joining. We can conclude that the model is not generalisable without the need of an external dataset.

The internal-external validation method, specifically out-of-time CV, was considered for the case study as it has proved useful for similar sport science projects in the past (Aydemir et al., 2023). However, it was deemed unsuitable for my dataset given that the data only covers 2017 to 2021, with the years 2020 and 2021 deviating significantly from previous trends (due to the pandemic).

Additionally, given the model's lack of generalisability, this ruled out the need for any other internal-external validation method.

3 Data

This chapter contains a background as to why a club would want to be able to predict a footballer's transfer fee and my own motivation behind choosing this particular case study (Section 3.1). A detailed exploration of the data acquisition process is also discussed (Section 3.2).

3.1 Dataset Motivation

The objective of this study is to explore different validation procedures, to gain a comprehensive understanding of the reliability of machine learning validation. I thought that the best way to explore validation techniques was to develop a model of my own using these procedures to assess the accuracy.

To achieve this, I chose to explore the realm of football economics, focusing on the validity of the machine learning prediction of a football player's transfer fee.

Data has changed the face of the transfer market. Inspired by the principles outlined in "Moneyball", we seek to determine if the "Moneyball" method, successful in baseball, can be adapted and validated in the context of football. This method, as exemplified by the success of the Oakland Athletics, involves using statistics to make strategic player recruitment decisions, often outperforming teams with more significant financial resources.

In addition, the literature "Estimating transfer fees of professional footballers using advanced performance metrics and machine learning" by (McHale and Holmes, 2023) has been a significant source of inspiration for the choice of dataset. In particular, the advanced methodologies and insights presented in this study have sparked my interest in exploring the validation procedures associated with predictive models for assessing player transfer fees. Notably, much of the sports science literature lacks clear explanations for their chosen validation techniques, which spurred my interest in exploring this aspect further.

The financial stakes involved in player transfers are substantial, necessitating accurate predictive models. It is important that the correct validation procedures are taken when accessing the validity of the model that predicts a player's transfer fee as a club can gain a competitive edge by identifying undervalued players, either enhancing the team's performance or serving as a lucrative investment through subsequent player sales.

Football clubs have evaluation models to predict how much a player is worth. My idea is to build a predictive model to asses how much a player could be sold or bought for depending on the buying clubs historical data in the transfer market. To illustrate the practical difference between these models, consider the following scenario:

Imagine a football club employs a data-driven model (model 1) to assess the global market value of players, which is a common task for data scientists in sports. Alongside

this, they could use a second model (model 2 - the one I am developing) that predicts potential transfer fees based on historical transactions, tailored specifically to specific clubs. For instance, if model 1 values one of their own players at X million euro, and Manchester United offers the same amount, X million, the club could then use their transfer fee predictor to examine past deals. If model 2 indicates that Manchester United would pay greater than X (based on historical data used to develop model 2), the club gains a substantial advantage in negotiations, knowing there is a precedent for a higher payment from this buyer.

This example demonstrates how strategic use of predictive models can give significant advantages to clubs in the context of player transfers. To build a model of this nature requires the correct validation procedures and methods.

Additionally, the over/under evaluation of player's fee has seen clubs breach the rule of "Financial fair play" (FFP). The FFP regulations is the requirement, where clubs are ordered to not spend more than the income that they generate. Just this season, it was announced that Everton breached FFP and were deducted ten points from the league table. Their breach was due to their over evaluation of the player's that they sold. A notable example is Richarlison, who was sold to Tottenham in the Summer of 2022. Aware of his desire to leave and anticipating his sale, Everton evaluated him at 80 million euros. Facing a delay in the deal, Everton proceeded to acquire replacements worth 80 million euros before completing the sale of Richarlison. However, Tottenham's highest offer after all the negotiations for Richarlison was 60 million euro and ultimately Everton sold him for this amount. The 20 million euro discrepancy proved insurmountable for Everton, leading them to spend more in the transfer window than their revenue allowed and breaching FFP.

This got me thinking. Everton valuing Richarlison at 80 million euro seemed like a fair price tag given the nature of the player and they knew they were negotiating with Tottenham. For context, the most Tottenham have ever spent on a player is 63 million (Tanguy Ndombele). Without a second prediction model, one should already be able to see that the likelihood Tottenham were ever going to spend 80 was extremely low (as it would have been 17 million more than their record fee). Everton did not take this into consideration and ultimately had severe financial and competitive repercussions.

A predictive model, based on transfer history per club would have been a better indication on how much Everton could have sold Richarlison to Tottenham for.

So I explored how I was going to get the data needed for such a model and with the use of suitable validation methods, assess the performance.

3.2 Dataset Acquisition and Pre-processing

Using Figure 3.1 as a visual aid for my data preprocessing workflow, I began by obtaining two datasets from Kaggle. Dataset A, known as the complete player database, encompasses information on each player across 110 columns. These data points, provided by the global franchise "FIFA" on an annual basis, are spread across six CSV files, covering

the years 2017 to 2021. In total, Dataset A includes details on 106,510 players. This dataset allows us to examine player information before their transfers to clubs.

Dataset B comprises details on each European transfer from 1992 to 2022. It includes the player names, the clubs involved, and our dependent variable, the transfer fee. The data in Dataset B was scraped by an anonymous author from Transfermarkt.com.

With 27,123 transfers recorded in dataset B, the next step was to merge the two datasets to have all the relevant information in the one dataset. To align dataset B with Dataset A, only transfers from 2017 to 2021 were retained. The merging process was executed using the Pandas package in Python.

Loans and free transfers were omitted from the dataset as they do not contain a transfer fee. Matching players from the two datasets based on their names presented challenges. To address this, the `fuzzywuzzy` string matching package played a crucial role. A pivotal decision was to apply an 80 percent confidence threshold for name similarity, a measure taken to mitigate the risk of overlooking transfer data points due to discrepancies, such as the inclusion of middle names in Dataset A. To make sure the merge was done on the correct player, additional considerations, including player positions and original club names, were taken into account in the code. This was iterated for each year from 2017 to 2021 as it was important to have the information from the player at the time of transfer and not data on the player after they were transferred.

Any transfer observations that had missing values were filled in manually by searching for the data online as all the information is public and the missing data was minimal ($<$20). The complete dataset contains 1316 transfers into the *"top five leagues" from 2017 to 2021.

*The top five leagues were chosen by the top five richest leagues in the world according to Deloitte's report based on the year 2022 (Deloitte, 2023). This is displayed in Table 3.1.

Table 3.1: Top 5 leagues estimated worth.

	Top five leagues Sorted by wealth	Value Estimate (Billion €)
	Premier League	10.6
	La Liga	4.8
	Serie A	4.6
	Bundesliga	4.3
	Ligue 1	3.7

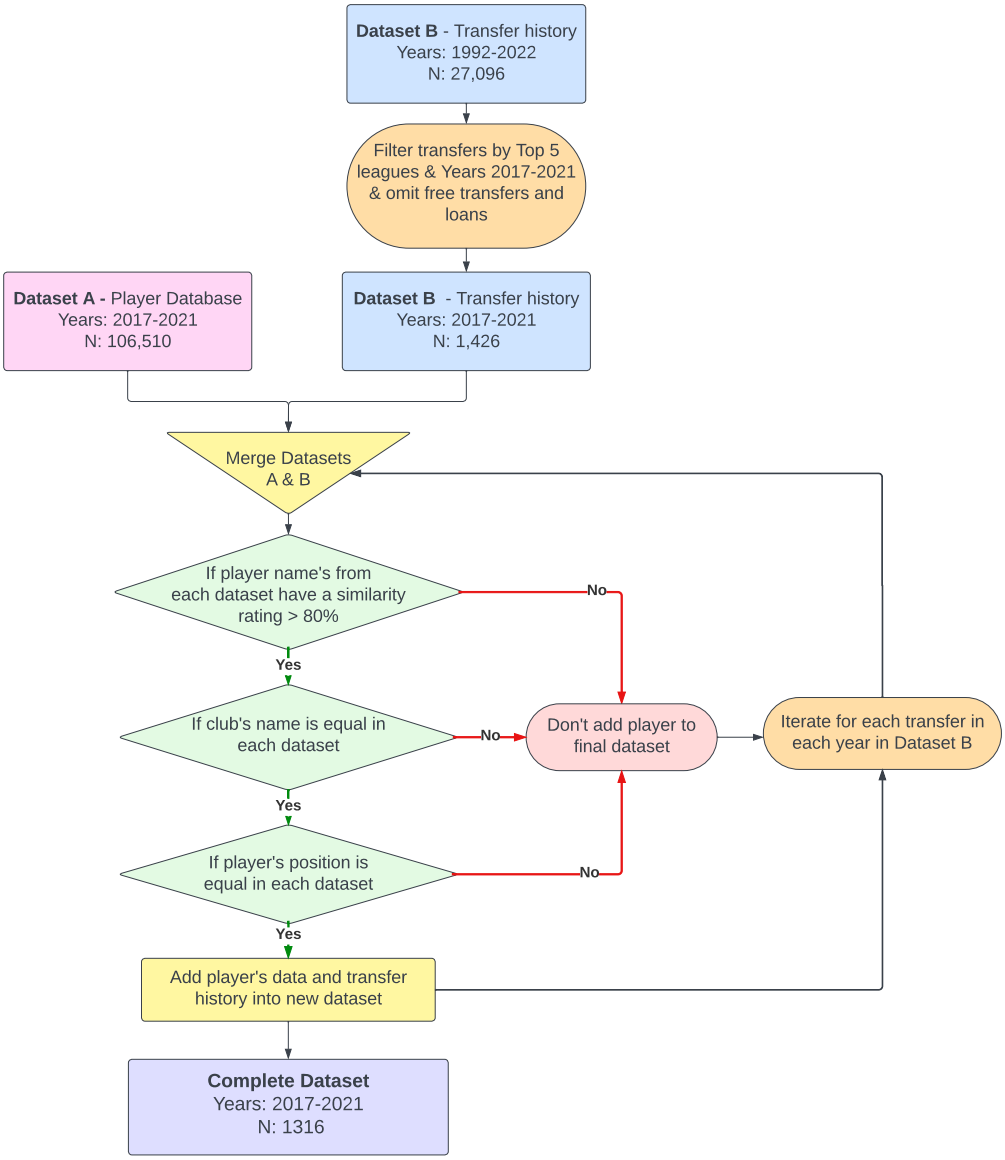


Figure 3.1: Data preprocessing flowchart.
Steps taken to gather the complete dataset and have it ready for implementation. Complete dataset contains 1316 Transfers into the top five leagues from 2017-2021.

4 Methods and Implementation

This chapter contains the methods from the literature review implemented on the case study. It is split into three parts of analysis. Section 4.1 contains the preliminary analysis which discusses the exploration of the predictors and the dependent variable. Section 4.2 contains the primary analysis which discusses the predictive model built and the validation procedures applied to it. Section 4.3 contains the secondary analysis in which we take a look at the overall predictions of the model how other models performed with this dataset. The methods and implementation section is combined because the methods discussed in the literature review are implemented here, with an explanation of their application in this specific context.

4.1 Preliminary Analysis

4.1.1 Feature exploration

Following the identification of potential factors influencing transfer fees within the complete dataset, the dataset was found to comprise 19 columns, with each variable described in Table 4.1.

Before proceeding to construct and validate a predictive model for transfer fees, a meticulous examination of the variables was conducted, facilitated through statistical techniques and visualisations in R.

This step is critical in the machine learning process, as it helps to determine which model could be best suited to the data. By examining the relationship between the features and dependent variable, we lay the foundation for effective validation procedures and model selection. This initial exploration enables us to gain insights into the data's structure and identify patterns or trends that can guide our modelling decisions.

The examination of the dependent variable, transfer fee, was initiated to understand its distribution, as depicted in Figure 4.1. The distribution revealed a pronounced exponential decrease in density as transfer fees increased, indicating that the majority of observations had lower transfer fees, while fewer observations were associated with higher fees. The presence of right-skewness in the transfer fee data could pose challenges in predictive modelling, particularly concerning outliers.

In response to the observed right-skewed distribution in the transfer fee data, a logarithmic transformation was performed because extreme values can disproportionately affect statistical analyses and the predictive models.

The second plot in Figure 4.1 depicts the distribution of the log-transformed transfer fee variable. Following the transformation, the dependent variable exhibited a normal distribution. A normal distribution for the dependent variable holds significant advantages for model development and interpretation (Kuhn and Johnson, 2013).

Table 4.1: Variable descriptions

Variable Name	Description
transfer_id	Id number given to each transfer transaction.
player_name	Name of player transferred.
age_sold	Age of player when sold.
nationality	Country the player was born.
continent_born	Continent the player is born. [1: Europe], [2: South America], [3: Africa], [4: ROW (Rest of World)].
overall_rating	"FIFA" rating assigned to each player. This metric is made up of 35 attributes calculated based on previous performances. This then becomes the ability level that is rated on a 1-99 scale.
team_importance	Player's role for the majority of games played while enrolled at previous club. [1: Starter], [2: Bench], [3: Reserve].
contract_remaining	Years remaining on a player's contract when the player was sold. [1], [2], [3], [4+].
club_left	Club that sold the player.
club_joined	Club that signed the player.
league_left	League of club that sold the player. [1: Premier League], [2: La Liga], [3: Bundesliga], [4: Seria A], [5: Ligue 1], [6: Other].
league_joined	League of club that signed the player. [1: Premier League], [2: La Liga], [3: Bundesliga], [4: Seria A], [5: Ligue 1].
position	Player's specific position.
position_category	Player's general position. [1: Goalkeeper], [2: Defender], [3: Midfielder], [4: Forward].
transfer_period	Transfer window the player was sold in. [1: Summer], [2: Winter].
year	Year the player was sold by their old club and signed by their new club. [2017], [2018], [2019], [2020], [2021].
season	Season the player was bought to play in. [2016/2017], [2017/2018], [2018/2019], [2019/2020], [2020/2021], [2021/2022].
FFI	Football Finance Index (FFI) is a score awarded to each club by Soccerex Football Finance. The Model includes 5 variables of playing assets, tangible assets (stadiums, facilities, etc), cash in the bank, owner potential investment and net debt.
fee	The Transfer cost for the club to sign the player in millions (€)

Table 4.2: Summary Statistics of Fee and Log Fee

	Min.	**1st Qu.**	**Median**	**Mean**	**3rd Qu.**	**Max.**
Fee	0.10	2.30	6.00	12.03	15.00	222
Log Fee	-2.30	0.83	1.79	1.72	2.71	5.40

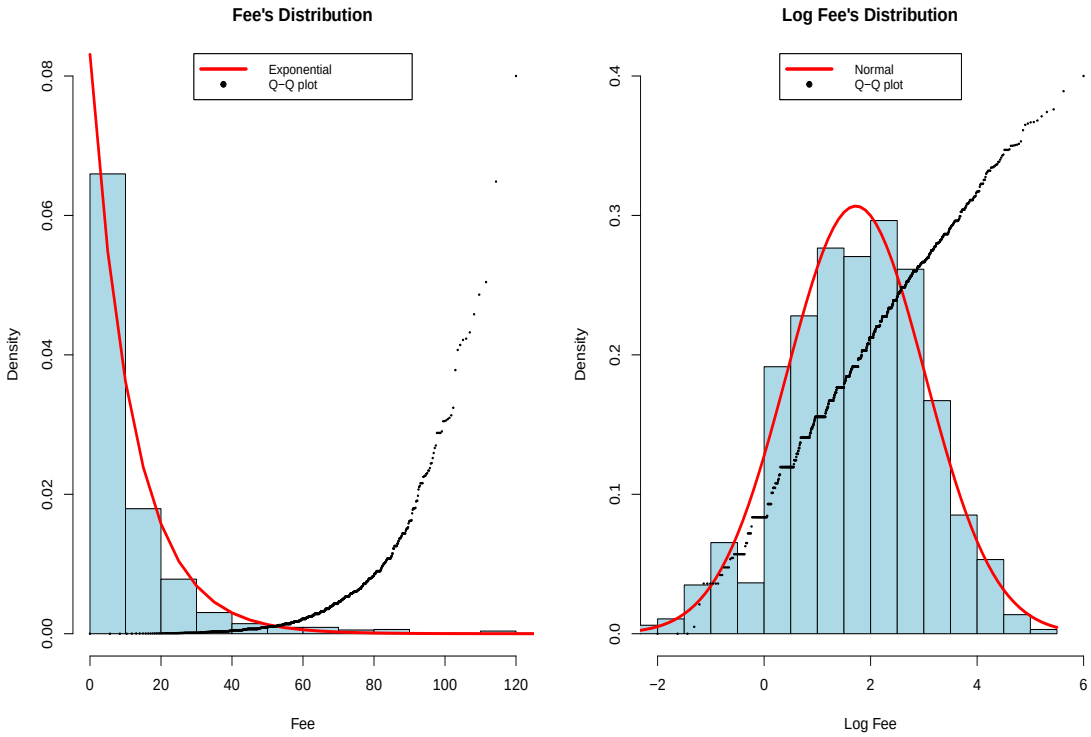


Figure 4.1: Dependent Variable - Transfer Fee
(left) Distribution of Transfer fee in millions.
(right) Distribution of Log Transfer fee in millions.
Transformed due to extreme range of values.

To gain insight into how different predictors influenced transfer fees, categorical variables were plotted on bar plots against the outcome variable, transfer fee. Each bar plot was constructed using the mean transfer fee associated with respective categories of the categorical variable. This systematic approach allowed for a visual examination of how each categorical predictor influenced transfer fees, providing valuable insights into their potential significance in the modelling process.

In Figure 4.2, observing that players leaving the Premier League and La Liga command the highest average transfer fees, at €18 million and €23 million respectively, suggests that clubs within these leagues hold players of perceived higher value, potentially due to the quality and competitiveness of these leagues. Conversely, players departing the Serie A or leagues outside the top five European clubs tend to have lower average transfer fees, indicating a potentially lower perceived value or less demand for players from these leagues in the transfer market. What stood out in this analysis is the proportion of La Liga players being considerably less than the other leagues and therefore, imbalance in this predictor could cause issues when modelling.

On the other hand, when considering the league_joined variable, it's notable that the Bundesliga, Serie A and Ligue 1 pays the lowest average transfer fee of approximately €8.5 million. Meanwhile, La Liga and the Premier League attract players at significantly higher average fees of €15 million and €20 million, respectively, underlining the allure and financial strength of these leagues. The findings from the bar plot align with our

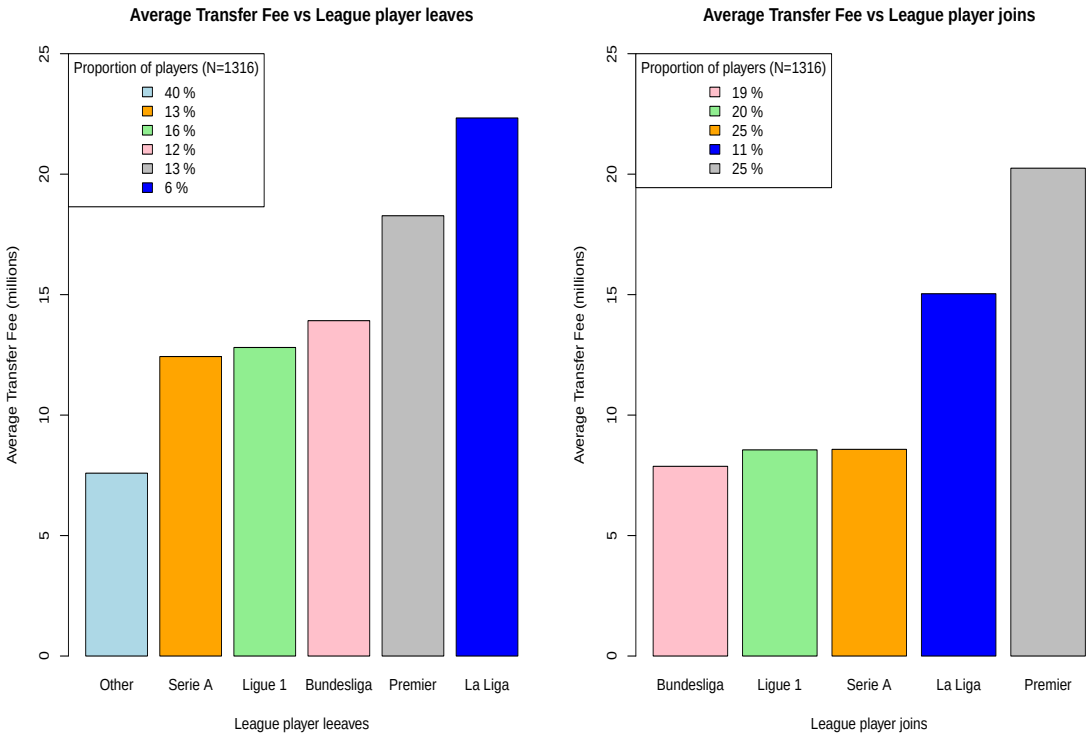


Figure 4.2: Transfer fee per League
(left) The average fee each league the player's left.
(right) The average fee for each league the player's joined.

earlier observations detailed in Table 3.1, where the Premier League emerged as the league with the highest estimated worth. This disparity suggests that clubs command higher transfer fees when selling players to the Premier League compared to other leagues (Depken II and Globan, 2021).

La Liga notably features a lower proportion of players compared to other leagues, suggesting potential differences in transfer activity volume. Stringent regulations on squad registration and foreign player quotas imposed by La Liga limit the influx of new players into the league. Consequently, the lower transfer activity volume in La Liga may reflect less frequent player turnover like some other leagues.

The player_role variable categorises players based on their involvement in matches as to whether they primarily started, were on the bench, or received minimal playing time. It's intuitive that players who regularly feature in the starting lineup hold higher value due to their perceived importance and contribution to the team. This expectation aligns with the findings depicted in Figure 4.3, where a significant increase in transfer fees is observed for players who were regular starters compared to those who served as reserves, with a difference of €10 million on average.

Furthermore, the position on the pitch also plays a crucial role in determining a player's value, as illustrated in Figure 4.3. As one moves up the pitch, from goalkeepers to defenders to midfielders to forwards, there is a corresponding increase in transfer fees. This trend is logical, considering the significant impact that players in more advanced positions, particularly forwards, have on goal-scoring opportunities and match outcomes.

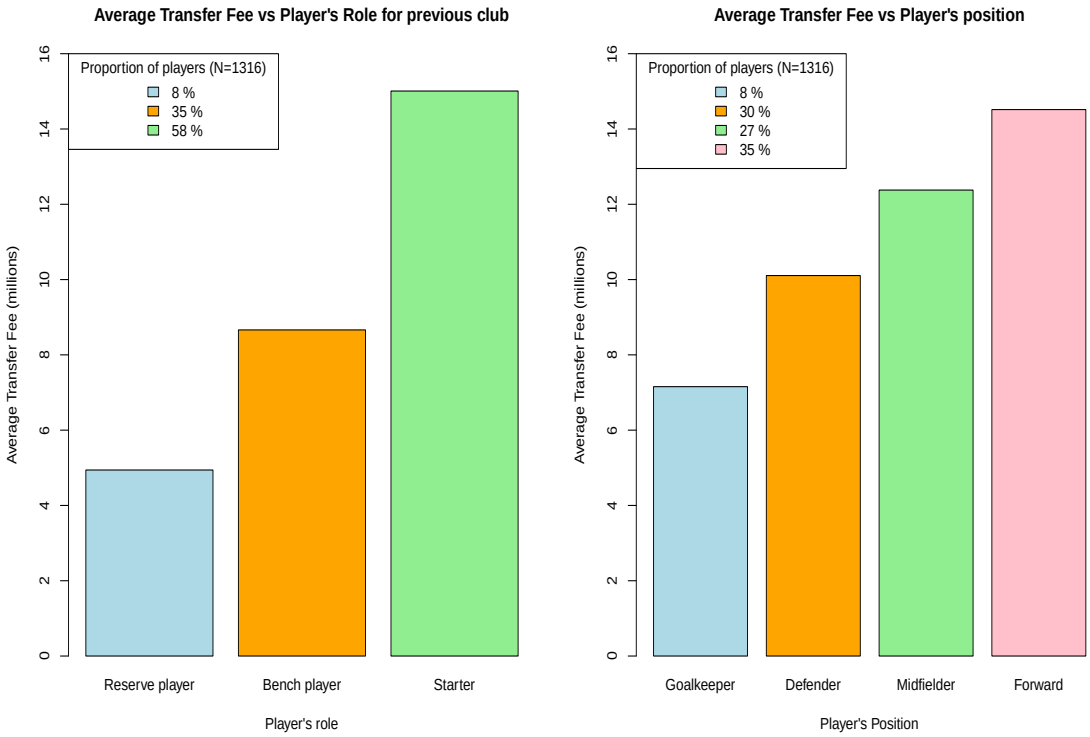


Figure 4.3: Transfer fee per Player Role
(left) The role of the player's for the club that sells them.
(right) The position of the player on the pitch

Players often find themselves tied into lengthy contracts with clubs, making it challenging for them to depart. Consequently, clubs tend to demand higher prices for players with more years remaining on their contracts, as depicted in Figure 4.4. To mitigate the risk of overfitting, contracts with four or more years remaining were grouped into a single category, ensuring they represent a substantial portion of the dataset (above a 5 percent threshold). It's exceedingly rare for players to be sold with such long contract durations, as clubs typically aim to maximise their utility from the player before considering transfers.

With 84 unique nationalities among transferred players, it became apparent that using nationality directly in a predictive model might not be feasible, given that some countries were represented by only one player in the dataset. However, recognising that nationality could still provide valuable insights into a player's transfer fee, the data was aggregated into a new column called continent_born, as depicted in Figure 4.4. Given the focus on Europe's top five leagues, it is unsurprising that European players dominate the proportion of continents represented, accounting for 80% of the dataset. Interestingly, South American players emerge as the most costly, commanding nearly double the average transfer fee. This trend may be attributed to the longstanding prominence of Brazilian and Argentine players in world football, underlining the influence of player nationality on transfer fees. Asia, Australia and North America aggregated into a rest of world category as the proportion of players from these continents respectively was $<1\%$.

The quantitative variables were visualised using scatter plots against log transfer fee as

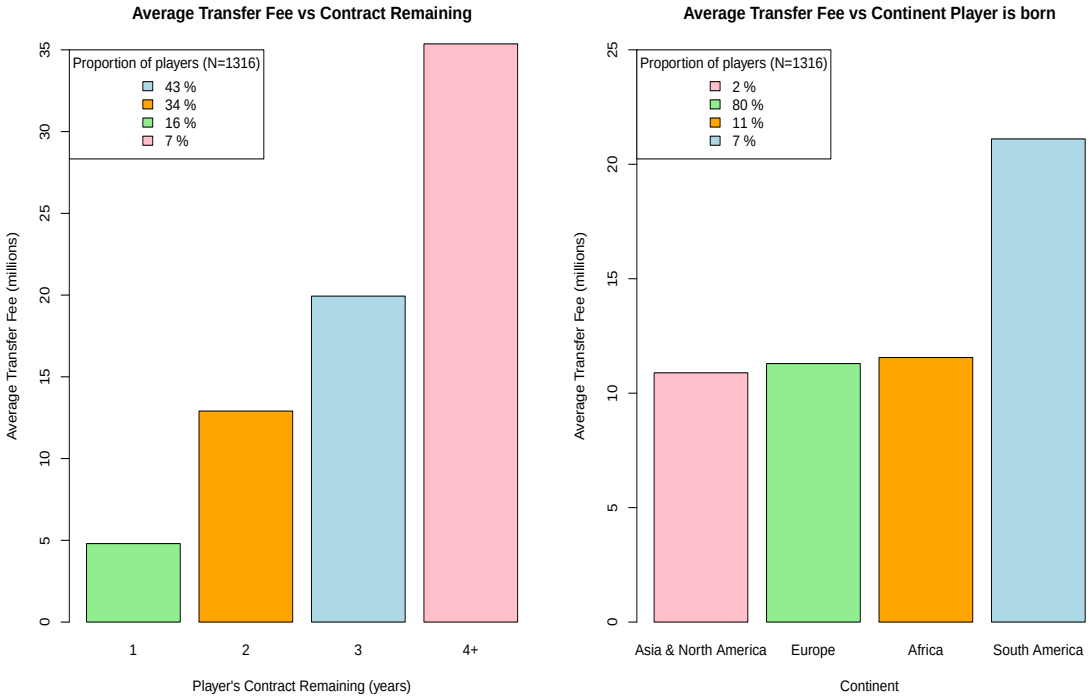


Figure 4.4: Player information
(left) The average transfer fee versus years remaining on player contract at the club that sells the player.
(right) The average transfer fee per continent the player is born.

depicted in Figure 4.5.

The "overall" variable, calculated by FIFA, is the metric I used to quantify a player's ability. Any model predicting a footballer's transfer fee has to have some way to depict "how good" a footballer is and these ratings provide a quantifiable measure of player quality and recent form. Notably, there exists a linear correlation of 45% between Log Transfer fee and Overall rating, indicating that higher-rated players tend to command higher transfer fees.

Similarly, the "Football Finance Index" (FFI) serves as a significant feature, reflecting the financial prowess of clubs that acquire players. Wealthier clubs typically allocate more resources to player acquisitions, resulting in higher transfer fees. Although the FFI values vary widely, ranging from 0.01 for several clubs to 4.11 for Manchester City, taking the logarithm of the FFI enhances its statistical properties. There is a linear correlation of 69% between Log Transfer fee and log FFI.

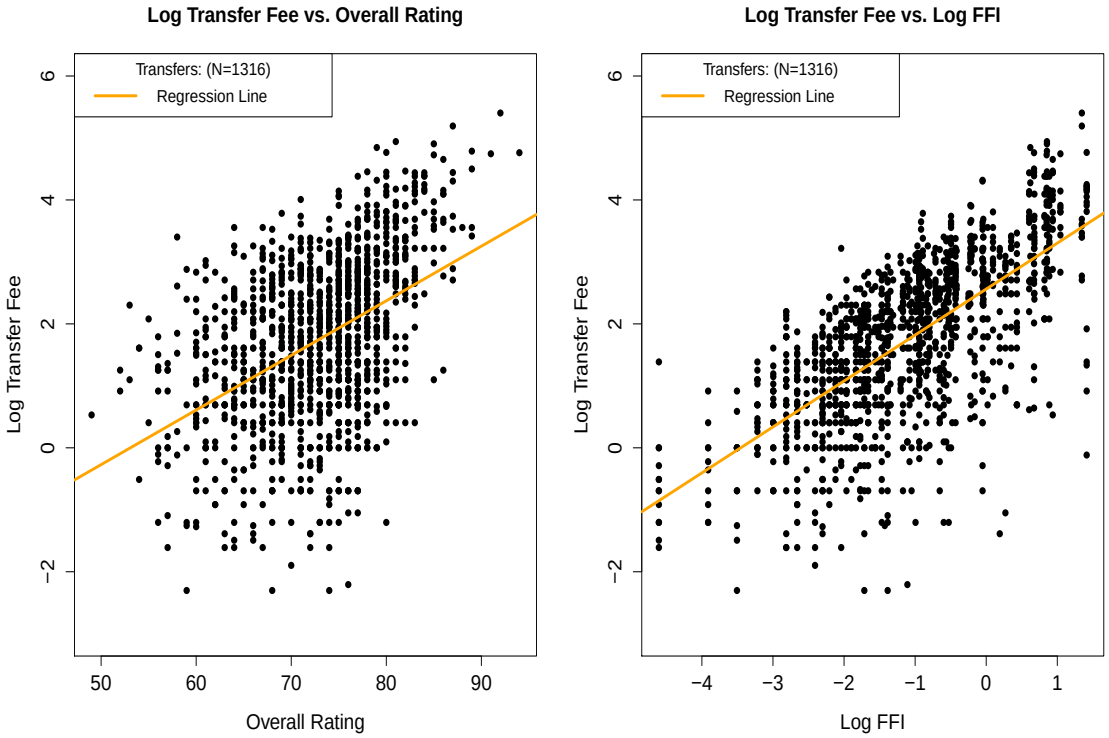


Figure 4.5: Quantitative Variables
(left) Log transfer fee versus player's overall FIFA rating when sold (calculated by player's form annually).
(right) Football finance index (FFI) is the financial power of the club purchasing. Log transfer fee versus purchasing club log FFI.

The exploration of features is a pivotal step in understanding the dataset and discerning potential determinants of transfer fees. By scrutinising features, we unearth patterns, relationships, and outliers within the data, providing invaluable insights into the dynamics of the transfer market. This analysis not only can help identify variables strongly correlated with transfer fees but also guides subsequent model selection and feature engineering processes. It is imperative to complete this step meticulously before initiating modelling and validation, as a thorough understanding of features is foundational to developing accurate and interpretable predictive models.

After examining the dataset, including the distribution of variables and their impact on the dependent variable, the focus shifted to model development and the validation of the chosen model.

4.2 Primary Analysis

4.2.1 Multiple Linear Regression Model

A linear regression model incorporating multiple predictors proved to be an appropriate choice for the analysis of the validation procedures. With the dependent variable Y and the predictors being x, the multiple linear regression (MLR) model is:

$$Y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \cdots + \beta_p x_{i,p} + e_i = \mathbf{x}_i^\mathsf{T} \boldsymbol{\beta} + e_i \tag{2.1}$$

for $i = 1, \ldots, n$. Each coefficient β_j for $j = 1, \ldots, p$ quantifies the expected change in the dependent variable Y associated with a one-unit change in the corresponding predictor $x_{i,p}$, holding all other predictors constant. p is the number of predictors, n is the sample size and the random variable e_i is the ith error. The assumption is that the errors $e_1, \ldots, e_n$ are independent and identically distributed (iid) with $\mathbb{E}(e_i) = 0$ and $\text{Var}(e_i) = \sigma^2 < \infty$ for $i = 1, \ldots, n$. (Olive, 2017).

The formula for this case study:

$$\begin{aligned}
\text{Log(fee)} = {} & \beta_0 + \beta_1(\text{age_sold}) + \beta_2(\text{overall_rating}) + \beta_3(\text{FFI}) \\
& + \beta_4(\text{contract_remaining_2}) + \beta_5(\text{contract_remaining_3}) + \beta_6(\text{contract_remaining_4+}) \\
& + \beta_7(\text{team_importance_Reserve}) + \beta_8(\text{team_importance_Starter}) \\
& + \beta_9(\text{position_category_Goalkeeper}) + \beta_{10}(\text{position_category_Midfielder}) \\
& + \beta_{11}(\text{position_category_Forward}) + \beta_{12}(\text{continent_born_ROW}) \\
& + \beta_{13}(\text{continent_born_Europe}) + \beta_{14}(\text{continent_born_South America}) \\
& + \beta_{15}(\text{league_left_Other}) + \beta_{16}(\text{league_left_La Liga}) \\
& + \beta_{17}(\text{league_left_Serie A}) + \beta_{18}(\text{league_left_Ligue 1}) \\
& + \beta_{19}(\text{league_left_Premier League}) + \beta_{20}(\text{league_joined_Ligue 1}) \\
& + \beta_{21}(\text{league_joined_La Liga}) + \beta_{22}(\text{league_joined_Serie A}) \\
& + \beta_{23}(\text{league_joined_Premier League}) + \epsilon
\end{aligned}$$

The regression model resulted in a multiple R^2 of 0.7006, indicating that approximately 70% of the variability in the football player's transfer fee can be explained by the predictors included in the model. This is considered a good fit, as explaining over 70% of the variability in such a complex and variable-dependent context as transfer fees suggests that the model captures significant underlying patterns effectively.

The coefficients can be seen in Table 4.3. The model's intercept is 1.571, which would be the base value of the dependent variable (log transfer fee) when all other predictors are at their reference level or have values of zero. Since the variables have been scaled, this intercept is not directly interpretable in the original units of the transfer fee.

The numerical predictors age_sold and overall_rating, were scaled using the `scale` function in R. This means their coefficients represent the number of standard deviations a given predictor is from the mean and how these standard deviations affect the log transfer fee. The reason it was important to scale these variable is because the overall_rating

Table 4.3: Linear Regression Model Results

Predictor	Coefficient	Std. Error	P Value
(Intercept)	1.571	0.106	$<.05$
Scaled Age Sold	-0.451	0.028	$<.05$
Scaled Overall Rating	0.531	0.031	$<.05$
Log FFI	0.420	0.021	$<.05$
Contract Remaining:			$<.05$
1	-	-	-
2	0.474	0.048	$<.05$
3	0.655	0.062	$<.05$
4+	0.712	0.090	$<.05$
Team Importance:			$<.05$
Bench	-	-	-
Reserve	-0.219	0.082	0.008
Starter	0.090	0.045	0.046
Position Category:			$<.05$
Defender	-	-	-
Goalkeeper	-0.160	0.082	.051
Midfielder	0.121	0.053	0.021
Forward	0.260	0.050	$<.05$
Continent Born:			0.067
Africa	-	-	-
ROW	-0.131	0.174	0.453
Europe	0.008	0.065	0.896
South America	0.207	0.098	0.036
League Left:			.01
Bundesliga	-	-	-
Other	-0.042	0.077	0.583
La Liga	-0.038	0.110	0.727
Serie A	0.114	0.092	0.215
Ligue 1	0.137	0.084	0.104
Premier League	0.144	0.091	0.114
League Joined:			$<.05$
Bundesliga	-	-	-
Ligue 1	0.028	0.069	0.682
La Liga	0.082	0.082	0.317
Serie A	0.082	0.069	0.237
Premier League	0.302	0.071	$<.05$

ranges from 1-99 with the minimum value in this dataset being 57 and the maximum being 94. Comparing this to the age_sold which ranged from 18-36, the overall_rating would overpower this and the continuous predictors. After scaling, the [min,max] ranges of age_sold and overall_rating were [-1.8,3.3] and [-3.6:3.2] respectively.

The coefficient for age_sold is -0.451, indicating that as a player's age at the time of sale increases by one standard deviation, the transfer fee decreases, holding other factors constant. This effect is statistically significant ($p<.05$). There is no surprise here that as a player gets older, their value tends to decrease.

With a coefficient of 0.531 for overall_rating, an increase in the rating of a player by one standard deviation is associated with an increase in the transfer fee. This is also significant($p<.05$). Since the overall rating of a player is calculated by the form the player has been on in the last year, the expectation was that this would have a positive coefficient.

The FFI initially presented a non-linear relationship with the transfer fee, suggesting that a simple linear model might not adequately capture the effect of FFI on the transfer fee. To address this non-linearity, a logarithmic transformation was applied to the FFI variable. As depicted in Figure 4.5, this transformation successfully linearised the relationship, allowing for a more accurate and interpretable inclusion of FFI in the regression analysis. Log FFI has a positive coefficient of 0.531, which is significant ($p<.05$), suggesting that higher values of the FFI are associated with higher transfer fees. This means that wealthier clubs would be predicted to spend more on a player than a less wealthy club, even though the player has not changed. This is one of the reasons that selling clubs will try to sell to richer clubs as they know they can make more money from that transaction.

As observed during the feature exploration phase, the dataset includes a number of categorical variables. To incorporate these categorical variables into a multiple linear regression model, which inherently requires numerical input a method known as one-hot encoding was employed. This is a statistical technique used to convert categorical variables into a binary numerical format that can be understood by statistical models, with 1 indicating the presence and 0 the absence of that category for each observation.

The dash ("-") in the coefficients table 4.3 signifies the reference category for the categorical variables. The coefficients of the remaining categories indicate the expected change in the transfer fee compared to this reference group. A positive coefficient suggests a higher transfer fee than the reference, and a negative one indicates a lower fee. These coefficients allow us to measure the impact of each category relative to the baseline established by the reference category.

To obtain p-values for each categorical predictor as a whole, the `summary` function applied to the model in R does not provide this directly. Therefore, I used an ANOVA (Analysis of Variance) to compare a model that includes the categorical variable with one that does not. This was done for each categorical variable, allowing for the assessment of the overall significance of each predictor.

Players with 2, 3, and 4+ years left on their contracts have positive coefficients, indicating

higher transfer fees compared to players with only 1 year left. This predictor is significant (p<.05) and each categroy is significantly different to a player with 1 year remaining on their contract. A club is less willing to sell a player if they have more years remaining on their contract and therefore the transfer fee would be expected to increase to as the years on their contract increased.

Being a Reserve has a negative coefficient of -.219 and is statistically significant (p<.05) compared to a benched player. If a player starts games the majority of the time they are more expensive (p<.05). This predictor is statistically significant.

Compared to defenders, goalkeepers have lower transfer fees, midfielders have slightly higher fees, and forwards have higher fees with all the p values <.05. Therefore this predictor is important in the model.

Players born in South America have higher transfer fees compared to those born in Africa (reference category). What is evident from the table is that the significance between the different continents outside of South America is statistically negligible. It could be the case that this predictor need only be a binomial predictor with South America being yes or no.

The league_left variable appears to be statistically significant with the variable's p value being <.05 however each category is not statistically significant when compared to the reference category, Bundesliga. While the other categories may not have shown significant differences from the Bundesliga, they could still have statistical differences among themselves.

To further investigate the significance of the `league_left` variables further, I examined the adjusted R^2 value. This adjusts the R^2 value based on the number of predictors in the model and the sample size. The formula for adjusted R^2 is:

$$\text{Adjusted } R^2 = 1 - (1 - R^2)\frac{n-1}{n-p-1} \tag{17}$$

where n is the sample size, and p is the number of predictors. Unlike R^2, which can increase with the addition of variables regardless of their relevance, adjusted R^2 can decrease if the added predictor does not improve the model sufficiently relative to the penalty for additional variables. Therefore, if the inclusion of `league_left` improved the adjusted R^2, it would suggest that despite their non-significance compared to the Bundesliga, these variables improve the model after accounting for the number of predictors. Conversely, if adjusted R^2 decreased with the inclusion of these variables, it might indicate that they do not add value to the model. This measure is especially useful over R^2 when deciding on the inclusion of variables in a model because it provides a more accurate reflection of the trade-off between model complexity and explanatory power. Model with league_left included. $R^2 = 0.7006$; $n = 1316$; $p = 23$

$$\text{Adjusted } R^2 = 1 - (1 - 0.7006)\frac{1316-1}{1316-23-1}$$

$$\text{Adjusted } R^2 = 0.6952$$

Model without league_left included. $R^2 = 0.697$; $n = 1316$; $p = 18$

$$\text{Adjusted } R^2 = 1 - (1 - 0.697)\frac{1316 - 1}{1316 - 18 - 1}$$

$$\text{Adjusted } R^2 = 0.6928$$

When league_left is included in the model the adjusted R^2 is greater than and without league_left in it. Based on this measure, and also the p value being $<.05$, the decision was made to keep the league_left variable in the final model due to the value being higher when kept in the model.

The most notable category in the league_joined variable is that players joining the Premier League have higher transfer fees than those joining the Bundesliga (reference category), however, the other league that player's joined did not have much of a difference statistically. This category was retained due to the importance of knowing if it was premier league or not, but it is definitely worth noting that this categorical variable may have only needed to be a yes or no answer to the question, "is the player joining the Premier League?"

Looking at the graphs in Figure 4.6 to visually see the performance of the model, displays the relationship between the observed values of the dependent variable and the values predicted by the linear model. The points in the plot on the left appear to follow a diagonal line quite closely, suggesting a good agreement between the observed and predicted values. This indicates that the model is generally effective in capturing the underlying trend in the data. However, there is a wider spread of values at the lower end of the predicted values compared to the upper end. This could be due to a number of reasons. One is that the relationship between the predictors and the outcome might not be perfectly linear, especially at lower values. This could mean that the model isn't capturing some of the distinctions that occur at different levels of the predictor variables.

The graph on the right plots the residuals (the differences between the observed and predicted values) against the predicted values, which is essential for checking the assumptions of homoscedasticity and independence. Homoscedasticity is the assumption of the constant variance of residuals. The residuals are distributed quite randomly around the zero line, however, there is a skew evident on the plot. The residuals range further on the negative side (down to -3) than on the positive side (up to 2). This skew might indicate that the model systematically over predicts certain values, particularly where the residuals are deeply negative or that there is a presence of outliers or influential points in the lower range of the data that could drag the regression line away from an optimal fit, resulting in skewed residuals. To hone in on this issue, I standardised the residuals to examine their distribution.

The standardised residuals aren't normally distributed, as shown by the divergence from the expected line in the Q-Q plot and the irregular distribution in the histogram in figure 4.7. Bootstrapping can help address this issue as it generates numerous resamples from the original data to empirically estimate the distribution of the regression coefficients.

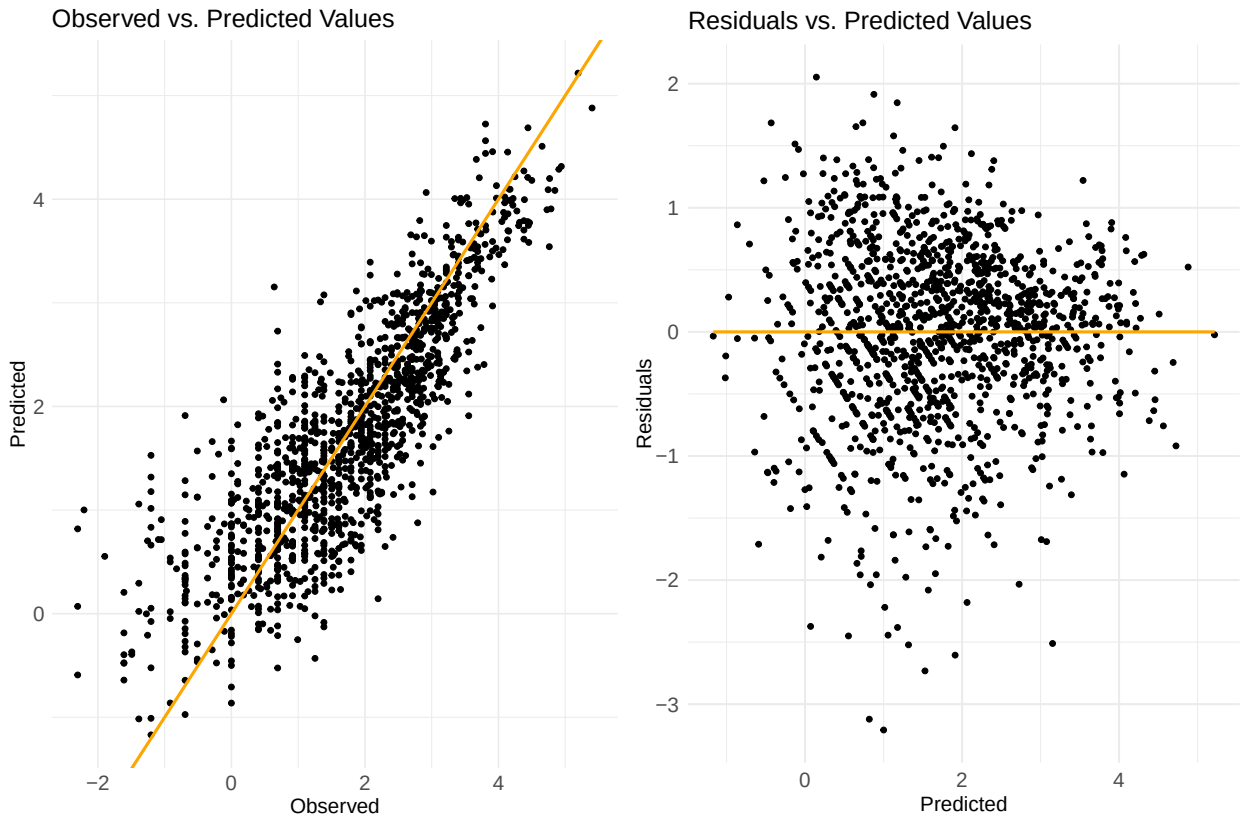


Figure 4.6: Linear Model - Log Transfer Fee
(left) The closer the points are to the line, the better. Wider spread of values at the lower end of the predicted values compared to the upper end.
(right) Checking for constant variance of residuals. Slight skew.

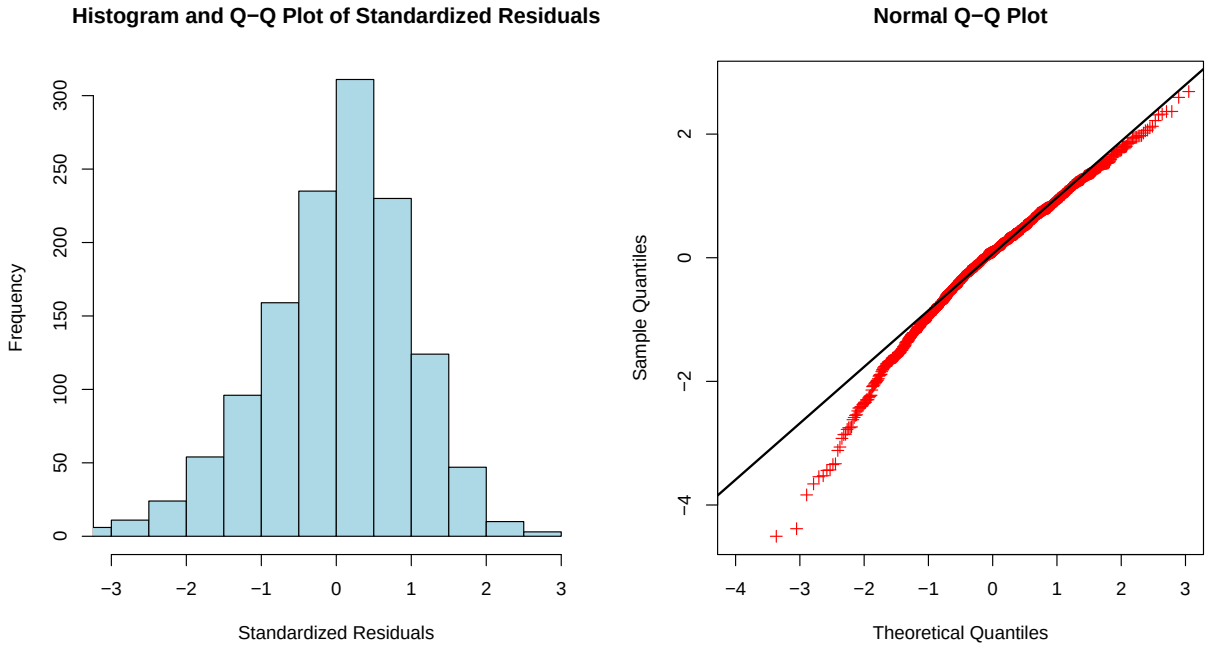


Figure 4.7: standardised Residuals
Positive Skew

4.2.2 Internal Validation Results

It is essential to perform rigorous internal validation using the resampling techniques in this case because as discussed in the literature review, internal validation should be performed in predictive modelling and the importance of it increases as the dataset size gets smaller. Since the dataset is N=1316, internal validation was performed. The method that should be used is completely dependent on what model you are using and what you want to find out about the model.

Throughout this section, I explain the validation methods which are appropriate and how they were applied to the data and which methods are inappropriate with a justification for each. Listed below are the appropriate techniques and inappropriate techniques for this multiple linear regression model.

Appropriate Internal Validation Techniques (from best to worst)	**Inappropriate Internal Validation Techniques**
1. Bootstrapping	1. Apparent Validation
2. Repeated Cross-Validation	2. Split Sample Validation
3. Monte Carlo Cross-Validation	3. LOO Cross-Validation
4. K-Fold Cross-Validation	4. Nested Cross-Validation

We have put bootstrapping as the most appropriate internal validation method for this model. The reason for this, is because in the multiple linear regression plots we noticed a skew. Bootstrap validation can be more robust to data with a skew because it inherently involves resampling with replacement. This method allows each bootstrap sample to potentially represent a different version of the dataset, helping to mitigate the impact of outliers or skewed data on the model validation process. It can provide insights into how the model might perform under different sample conditions and detect biases or skew in the model predictions that might not be apparent during standard validation.

Firstly, by resampling the original dataset with replacement many times and refitting the model to each of these bootstrap samples, a distribution for each coefficient in the model can be created. The bootstrap estimates are close to the original regression coefficients, as can be seen in the table 4.4 below. This reinforces confidence in the model's findings and suggests stability in the estimates. The intercept and the 23 predictors coefficient's distribution are displayed in figure 4.8. All the standard coefficients fall inside the bootstrapped confidence intervals.

Secondly, the OOB ("Out-Of-Bag") estimates can help in assessing the stability of the model predictions across different bootstrap samples. If the skew is significant enough to potentially bias the model's predictive performance, using the OOB RMSE could offer a more realistic assessment of how the model will perform on unseen data.

It allows for the test of the model's robustness to variations in the data that might not be captured through traditional cross-validation, particularly if the skew affects only certain segments of the data or occurs randomly (due to resampling with replacement). Bootstrapping is advantageous when analysing small datasets or when the data include outliers (both of these are true in our dataset).

Table 4.4: Linear Regression - Standard versus Bootstrapped Results

Predictor	Coefficient	Std. Error	Boot Coefficient	Boot Std. Error
(Intercept)	1.571	0.106	1.584	0.124
Scaled Age Sold	-0.451	0.028	-0.453	0.035
Scaled Overall Rating	0.531	0.031	0.531	0.035
Log FFI	0.420	0.021	0.425	0.024
Contract Remaining:				
1	-	-	-	
2	0.474	0.048	0.470	0.053
3	0.655	0.062	0.650	.060
4+	0.712	0.090	0.711	.091
Team Importance:				
Bench	-	-	-	
Reserve	-0.219	0.082	-0.215	0.105
Starter	0.090	0.045	0.090	0.047
Position Category:				
Defender	-	-	-	
Goalkeeper	-0.160	0.082	-0.155	0.115
Midfielder	0.121	0.053	0.133	0.050
Forward	0.260	0.050	0.263	0.050
Continent Born:				
Africa	-	-	-	
ROW	-0.131	0.174	-0.113	0.214
Europe	0.008	0.065	0.010	0.068
South America	0.207	0.098	0.204	0.096
League Left:				
Bundesliga	-	-	-	
Other	-0.042	0.077	-0.045	0.081
La Liga	-0.038	0.110	-0.030	0.102
Serie A	0.114	0.092	0.110	0.093
Ligue 1	0.137	0.084	0.127	0.079
Premier League	0.144	0.091	0.145	0.086
League Joined:				
Bundesliga	-	-	-	
Ligue 1	0.028	0.069	0.027	0.071
La Liga	0.082	0.082	0.069	0.073
Serie A	0.082	0.069	0.081	0.062
Premier League	0.302	0.071	0.292	0.067

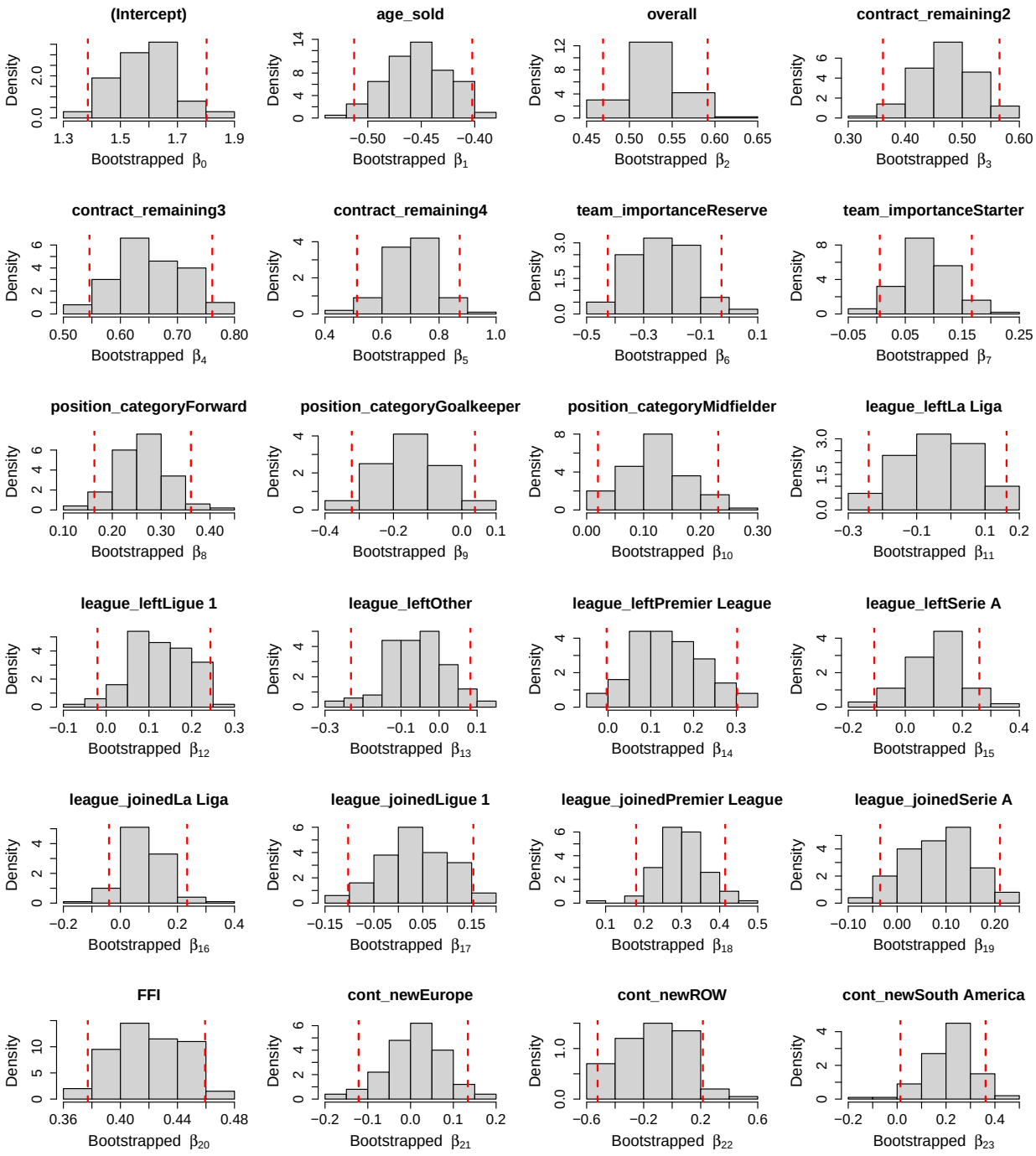


Figure 4.8: Distribution of Bootstrapped Coefficients
Red lines indicate the 95% confidence intervals.

While OOB bootstrapping is traditionally associated with ensemble methods such as random forest, it is easily implemented to multiple linear regression. The following steps were performed in in R.

- **Bootstrap Sampling:** Create multiple bootstrap samples **(with replacement)** from the original dataset. The number of bootstrap samples B used was 1000 as it was computationally possible.
- **Model Fitting:** Fit the linear regression model to each of the "In-Bag" samples. $\approx 63.2\%$ of the dataset for each sample. Store the bootstrap estimates $\hat{\theta}_b$.
- **OOB Prediction:** For each bootstrap model, predict the outcomes for the data points **not** included in its sample (the OOB data). $\approx 36.8\%$ of the dataset for each sample.
- **Validate Performance:** Assess the model's performance based on these OOB predictions. With the 1000 different RMSE values, display on a box plot to look at the variability of the RMSE and take the mean of the 1000 RMSE values to assess the model's performance.

After performing bootstrap resampling, the average RMSE value on the fit of the model was .71 with a standard deviation of 0.017. This seems reasonably low, however it is very important that the OOB RMSE value is the value that is focused on as this is the data that the model has not seen and one of the main focus's of validation is to mitigate overfitting. The average out-of-bag (OOB) root mean square error (RMSE) for our model was 0.73, with a standard deviation of 0.023 across the bootstrap samples. This suggests that the model's error has low variability and is expected to perform with similar accuracy on different subsets of the data. Looking at the two values on the box plot in figure 4.9, a difference between the two can be seen.

The ideal situation is that the two errors are statistically the same which would signify overfitting isn't present.

To test if they are significantly different values, a paired t-test was performed. The assumption of a paired t-test is that the differences between paired observations should be approximately normally distributed which was tested this with a Shapiro Wilk's test and the p-value $= 0.7427$ so the assumption was met.

Using a paired t-test, the null hypothesis is $H0$: There is no difference in the RMSE between the training and test sets. The test statistic was -22.231, which is quite far from 0. This indicates a difference between the two sets of measurements. The p-value $<.05$ and therefore we reject the null hypothesis. There is a difference between the two RMSE values, therefore, overfitting cannot be ruled out.

After completing the bootstrapping process, our model has been internally validated. We identified potential issues with certain variables and observed signs of overfitting. The model's error rate is reported at 0.73, which shows a significant improvement over the naive method of guessing the mean each time for the transfer, which has an error rate of 1.3. This indicates a 43% enhancement in model performance using our linear

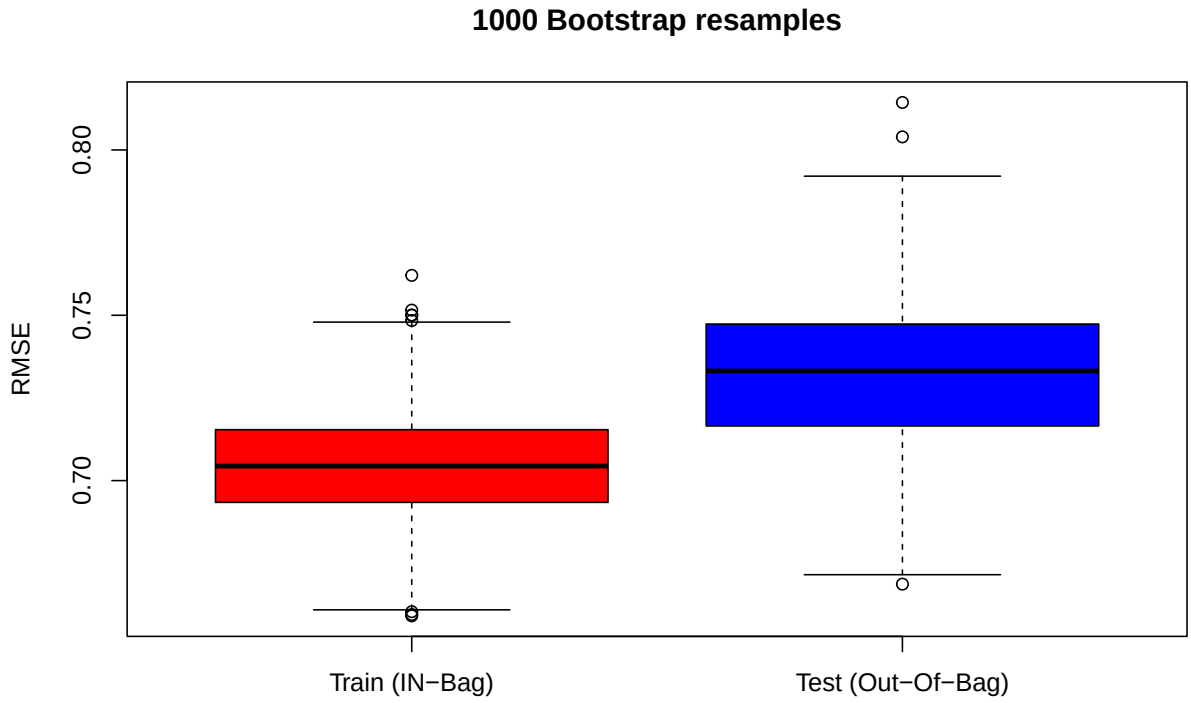


Figure 4.9: Box Plot - Bootstrap RMSE
The training RMSE is the model fit which is tested on the "In-bag" data 1000 times on 63.2% of the data each time. The test RMSE is on the remaining unseen data or "out-of-bag" on 36.8% of the data each time.

approach. To further refine our model and tackle the issue of overfitting, additional steps were considered.

To validate this further, I looked at shrinking the less important covariates as this can reduce overfitting. I looked at applying regularisation to the model. Considering the incorporation of regularisation techniques such as LASSO (Least Absolute Shrinkage and Selection Operator) could be beneficial for the further validation of the features. LASSO regularisation adds a penalty equivalent to the absolute value of the magnitude of coefficients, which can be expressed as:

$$\sum_{i=1}^{N}\left(Y_i - \beta_0 - \sum_{j=1}^{p}\beta_j X_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = RSS + \lambda\sum_{j=1}^{p}|\beta_j| \qquad (18)$$

where RSS represents the residual sum of squares, β_0 is the intercept, β_j are the coefficients for predictors X_{ij}, λ is the regularisation parameter, N is the number of observations, and p is the number of predictors. This method not only helps in reducing overfitting but also assists in feature selection by shrinking some coefficients to zero, thus simplifying the model.

As discussed in the literature review, LASSO is best implemented with nested CV. λ is

usually selected with nested cross validation. Through the different iterations of k-fold CV, the value of lambda will be chosen by the one that gives the minimum performance criterion which is RMSE in our case.

I stated that nested CV was an inappropriate validation technique for the multiple linear model used. This was because nested CV is used to fine tune hyperparameters and there are none in a linear model, but with the inclusion of regularisation, there is now a hyperparameter (λ) to fine tune.

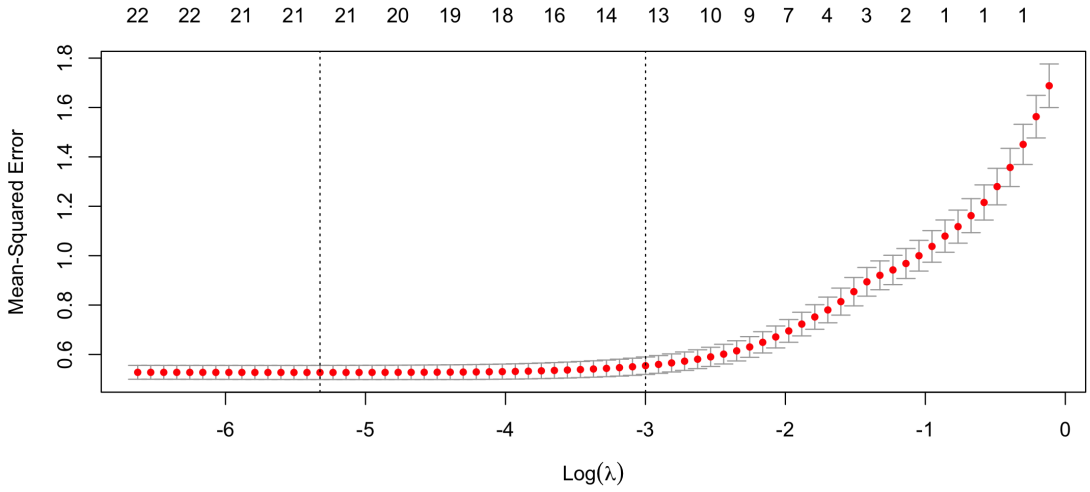


Figure 4.10: Cross-validated MSE against log(λ)
Lower lambda values result in less regularisation, while higher values increase regularisation. The line on the left indicates the lambda value that minimise the cross-validated MSE. The line on the right shows the most regularised model such that the error is within one standard error of the minimum MSE. The value of λ chosen = log(-4.615) = 0.0099

I then compared the LASSO model to the linear model in Figure 4.11. LASSO's model is optimised by performing nested CV which only has one difference to repeated k-fold CV which is the fine tuning of the hyperparameter. For this reason, to ensure a fair comparison between the models, I employed repeated k-fold CV to the linear model as well as the bootstrap already performed. This ensures that the sole difference between the models validation methods be regularisation so I could see its effect. Repeated k-fold CV is particularly suited to smaller datasets like the one that has been implemented in this project because it ensures that all available data is utilised effectively for both training and validation. However, due to the skew in the linear model we would expect the variability to be higher than the bootstraps predictions. The expectation is that the difference in results would not be large and this is why repeated k-fold CV was put second on the list of appropriateness.

To align the repeated k-fold CV with the bootstrap approach, I aimed for a similar number of test points in each fold. Given $N = 1316$ and the out-of-bag (OOB) bootstrap test set being 36.8%, the number of tested points used for each bootstrap is calculated as

follows:

$$N \times 36.8\% = 1316 \times 0.368 = 484.288 \approx 484$$

To make the repeated k-fold cross-validation (CV) comparable, set $\frac{N}{K} = 484$. Solving for K:

$$\frac{1316}{484} \approx 2.71$$

Since K must be an integer, this rounds to a 3-fold cross-validation. Now, letting $R \times K = B$, where R is the number of repetitions and B is the total number of bootstrap samples (1000), we find R:

$$R = \frac{1000}{3} \approx 333$$

This ensures the setup for repeated k-fold CV is comparable to the bootstrap approach. There are 999 (RXK) RMSE values in the repeated k-fold CV, with 33% of the data set tested on for each repeat which is close to the number of OOB RMSE values (1000) so it is a fair comparison.

The effect of LASSO did not remove any variables, but slightly shrunk some of them. As can be seen in the diagram, the addition of a regularisation did not reduce the error. Interestingly, this would suggest that the current linear model is not overfitting the data. This validation method indicates that implementing a LASSO model is redundant and could unnecessarily complicate the model without providing additional benefits.

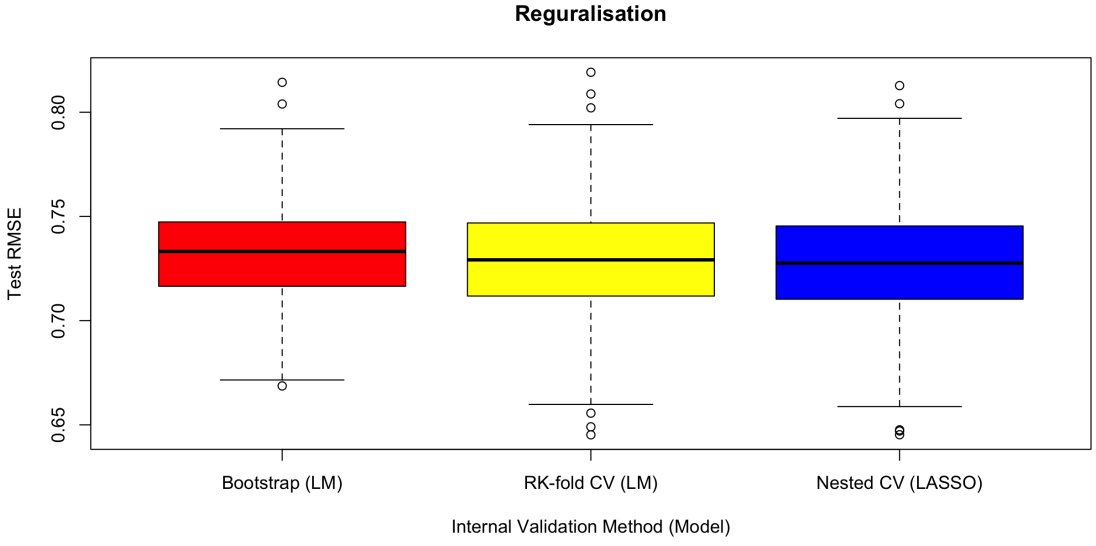


Figure 4.11: LASSO Box Plot
Variability of the 1000 OOB RMSE's, 999 Rk-fold CV test RMSE's and 999 Nested CV test RMSE's. LASSO model has similar error rates and also similar variability on those errors.

For the purpose of further exploration of internal validation methods, I applied the less suitable methods and provided a comparison of their performance in Table 4.5. This comparison reveals consistent model fit across the methodologies. Below, I delve into the reasons behind considering certain methods as less appropriate for this project and investigate whether their results align with these expectations.

Table 4.5: Method Comparison

Validation Method	Train Sample	Test Set	Mean RMSE Train	Mean RMSE Test	SD RMSE Test
Bootstrap (B=1000)	63.2%	36.8%	0.71	0.73	0.023
RK-Fold-CV (R=333),(K=3)	66.7%	33.3%	0.71	0.73	0.012
Apparent	100%	0	0.71	-	-
Split Sample	63%	37%	0.7	0.75	-
Monte Carlo-CV (M=1000)	63%	37%	0.71	0.73	0.028
LOO-CV (K=N)	99.9%	0.1%	0.71	0.56	0.464
K-fold-CV (k=10)	90%	10%	0.71	0.73	0.049

Monte Carlo CV ranks third in terms of suitability for our analysis and would be expected to give slightly optimistic results due to data overlapping in the train and test set. As discussed in the literature review, there is only one fundamental difference between MC CV and bootstrapping. MC CV does not resample with replacement. Therefore you can set the 'In-bag' samples size and OOB test set size instead of being fixed at approximately 63.2:36.8 split like the bootstrap is, you set this ratio. The reason that it is appropriate is because the dataset size being quite small, it is an efficient way to validate the model many times. The reason I have it set it less suitable than repeated k-fold CV is because without the systematic approach of k-fold, there's a higher risk that some models might train on subsets of data that are not representative of the overall dataset, especially since the dataset is small. For 1000 resamples, the results show the same error rate as them for both the fit and test error. This consistency is a good sign that the model's performance is reliable.

The dataset size (1316) is quite small for k-fold CV so this wouldn't be an ideal choice because repeated k-fold CV is computationally achievable. K-fold CV is more effective with larger datasets. I performed 10-fold validation to see the difference between this method and the previously mentioned ones. The error rates remained the same again, however the variability doubled in size. This is likely due to testing on only 10% of the data at each fold rather than a third of the data like we had seen with the previous methods. Increasing the value k here would increase the accuracy of model fit but

decrease the accuracy of model's prediction error rates as shown in the box plots in figure 4.12.

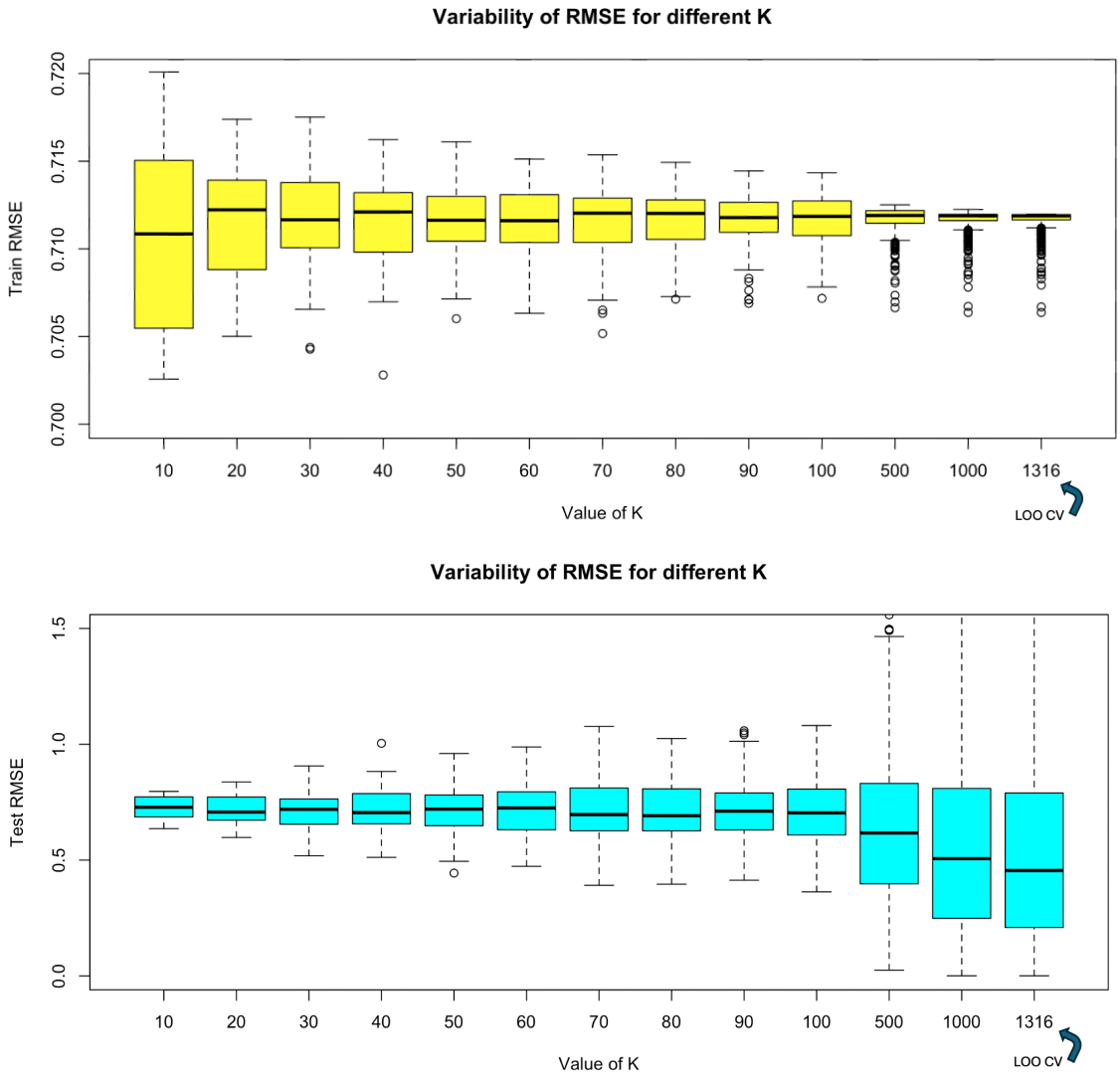


Figure 4.12: K-fold CV Box Plot
(Top) Variability of Train or fit RMSE decreases as k increases.
(Bottom) Variability of Test RMSE increases as k increases. The mean error rate decreases but it is not a reliable estimate when the variability is so high. Note: y axis are not comparable between the two plots as the test RMSE's variability is much higher.

While it may seem that the choice of validation method is inconsequential given the consistent results across different methods in this instance, it's important to recognise that this is one specific case. In different datasets or with various models, the outcomes could vary. Each validation method has its own merits and limitations. Therefore, careful consideration should always be given to selecting the most appropriate validation method based on the specific characteristics and requirements of each analytical scenario.

The model is not stable enough to appropriately analyse LOO CV's results. The variability in the RMSE values is large and therefore not a reliable validation method in this case. The over estimation of the error rate is apparent with the mean of its test RMSE being .56 but with its standard deviation being significantly higher than the other

methods of 0.464, the error rate of .56 is optimistic. However, this could indicate that outliers are effecting the model heavily.

The dataset is too small to justify the use of apparent validation, where 100% of the data is used for development without a separate test set. Relying solely on apparent validation provides no insight into potential overfitting, as it does not assess how the model performs on unseen data.

As outlined in the literature review, split sample validation is generally not recommended. For this dataset, a resampling technique is essential to ensure reliable results. Using a split sample resulted in a test RMSE of 0.75, which likely overestimates the model's true error rate as using the resampling techniques we saw this value was lower.

In summary, while the selection of validation methods did not significantly impact the outcomes of this project, it is important to recognise that this may not be the case in all scenarios. Best practices in internal validation suggest employing resampling techniques to assess model fit and utilising test set RMSE to identify overfitting. Properly selecting and applying the most appropriate validation method is important for generating reliable predictions and accurately evaluating model performance. Although applying all internal validation methods to a single dataset was done here for exploration and comparison, it is generally advised to only use one or two resampling technique as long computational resources allow. This approach ensures a balance between thoroughness and practicality in model validation.

4.3 Secondary Analysis

4.3.1 Multiple Linear Regression Predictions

To investigate the reasons behind the varying accuracy of our model, I analysed the 15 most accurate predictions and the 15 least accurate ones. This review aimed to identify the key factors influencing the model's performance.

As discussed by Wooldridge, 2012, the use of logarithmic transformations in econometrics is fundamental for dealing with relationships in economic data. Wooldridge also provides guidance on reversing these transformations, which I have applied to ensure that predictions are directly comparable to actual values.

The tables below present the top 15 and bottom 15 predictions, sorted by their residuals. It is worth noting that the residuals were from the model with the transformation applied so it was the top 15 and bottom 15 predictions sorted by those residuals and not the differences seen in the table. The predictions and actual values have since been back-transformed for interpretability. This sorting is purposeful, as it highlights the most and least accurate predictions, thereby providing a clear measure of the model's performance across different data points.

The well predicted players fees have a much broader range of fees compared to those poorly predicted fees. Predominantly, the inaccuracies arose from overpredictions, as

we indicated by the negatively skewed histogram of the standardised residuals. This trend of overprediction is likely influenced by extreme outliers within the dataset, such as high-profile transfers like Cristiano Ronaldo and Neymar, whose significant transfer fees might cause the model to overestimate the fees for other players.

Table 4.6: Best 15 Predictions sorted by lowest residuals

Player Name	Year	Club Left	Club Joined	Fee	Predicted
Fabio Borini	2018	Sunderland	AC Milan	5.50	5.504
Harrison Reed	2020	Southampton	Fulham FC	6.50	6.506
Óliver Torres	2019	FC Porto	Sevilla FC	11.00	11.019
Denis Cheryshev	2019	Villarreal	Valencia CF	6.00	5.980
Alexander Sörloth	2018	FC Midtjylland	Crystal Palace	9.00	8.939
Will Hughes	2017	Derby	Watford FC	9.10	9.171
Pierre-Emile Höjbjerg	2020	Southampton	Spurs	16.60	16.427
Nathaniel Chalobah	2017	Chelsea	Watford FC	6.30	6.228
Nuno Da Costa	2017	Valenciennes FC	RC Strasbourg	1.50	1.483
Nikola Vlasic	2021	CSKA Moscow	West Ham United	30.00	29.491
Kieran Trippier	2019	Spurs	Atlético de Madrid	22.00	22.412
Salif Sané	2018	Hannover 96	FC Schalke 04	7.00	7.136
Valère Germain	2017	Monaco	Olympique Marseille	8.00	8.161
Adama Diakhaby	2017	Stade Rennais	AS Monaco	10.00	9.794
Giuseppe Pezzella	2020	Udinese Calcio	Parma Calcio 1913	5.75	5.876

To further explore the model, I analysed hypothetical future transfers of several players who could transfer clubs this upcoming year. These scenarios, while speculative, aim to examine how the choice of purchasing club might influence the transfer price. This is shown on example predictions in Figure 4.13.

Given the observed annual inflation rate of 9%, I adjusted the predictions from the multiple linear model by applying the following equation:

$$\text{Adjusted Fee} = \text{Predicted Fee} \times (1 + 0.09)^{\text{Years since base year}} \tag{19}$$

Due to the impact of COVID-19, which affects two years out of the five-year dataset, including the year variable in the model was problematic.

These predictions show that just by changing the club and subsequently a different league, for the same player, there is a different prediction of transfer fee. Real Madrid have a slightly higher FFI and are therefore wealthier than Manchester United however Manchester United are in the Premier League, so the model has predicted that Manchester United would have to pay more than Real Madrid. This highlights the importance of the league joined predictor.

Lets take the bottom graph in Figure 4.13. The young player Pedri who currently plays for Barcelona. If Barcelona had a predictive model like the one employed in this project, they could see that if they were negotiating with Manchester United and Manchester United offered his evaluated price of €80 million (blue line), it would be unwise to accept

Table 4.7: Worst 15 Predictions sorted by highest residuals

Player Name	Year	Club Left	Club Joined	Fee	Predicted
Robert Snodgrass	2021	West Ham	West Bromwich Albion	0.110	2.721
Marco Sau	2019	Cagliari Calcio	UC Sampdoria	0.100	2.267
Isaac Lihadji	2020	Marseille B	LOSC Lille	0.300	4.609
Leonardo Candellone	2020	Torino	SSC Napoli	0.500	6.766
Valter Birsa	2019	Chievo Verona	Cagliari Calcio	0.300	3.736
Mohamed Ihattaren	2021	PSV Eindhoven	Juventus FC	1.900	23.405
Loick Landre	2018	Genoa	Nîmes Olympique	0.150	1.739
Julian Pollersbeck	2020	Hamburger SV	Olympique Lyon	0.250	2.880
Mattie Pollock	2021	Grimsby Town	Watford FC	0.300	3.248
Diant Ramaj	2021	FC Heidenheim	Eintracht Frankfurt	0.100	1.073
Vid Belec	2018	Benevento	UC Sampdoria	0.300	2.762
Alexandre Mendy	2017	Guingamp	FC Girondins Bordeaux	0.600	4.807
Antonino La Gumina	2018	US Palermo	Empoli FC	9.000	1.155
Romain Philippoteaux	2019	AJ Auxerre	Nîmes Olympique	0.300	2.301
Dominik Kohr	2017	FC Augsburg	Bayer 04 Leverkusen	2.000	15.284

this. This model tells them that they should ask Manchester United for a much higher fee in the region of 140 million as based on the predictions, Manchester United would be willing to pay this amount if they were seriously interested in the player. On the other hand, if Dortmund offered the €80 million, the negotiation tactic would have to be more different because Dortmund would be unlikely to pay much more than there offer based on the models prediction.

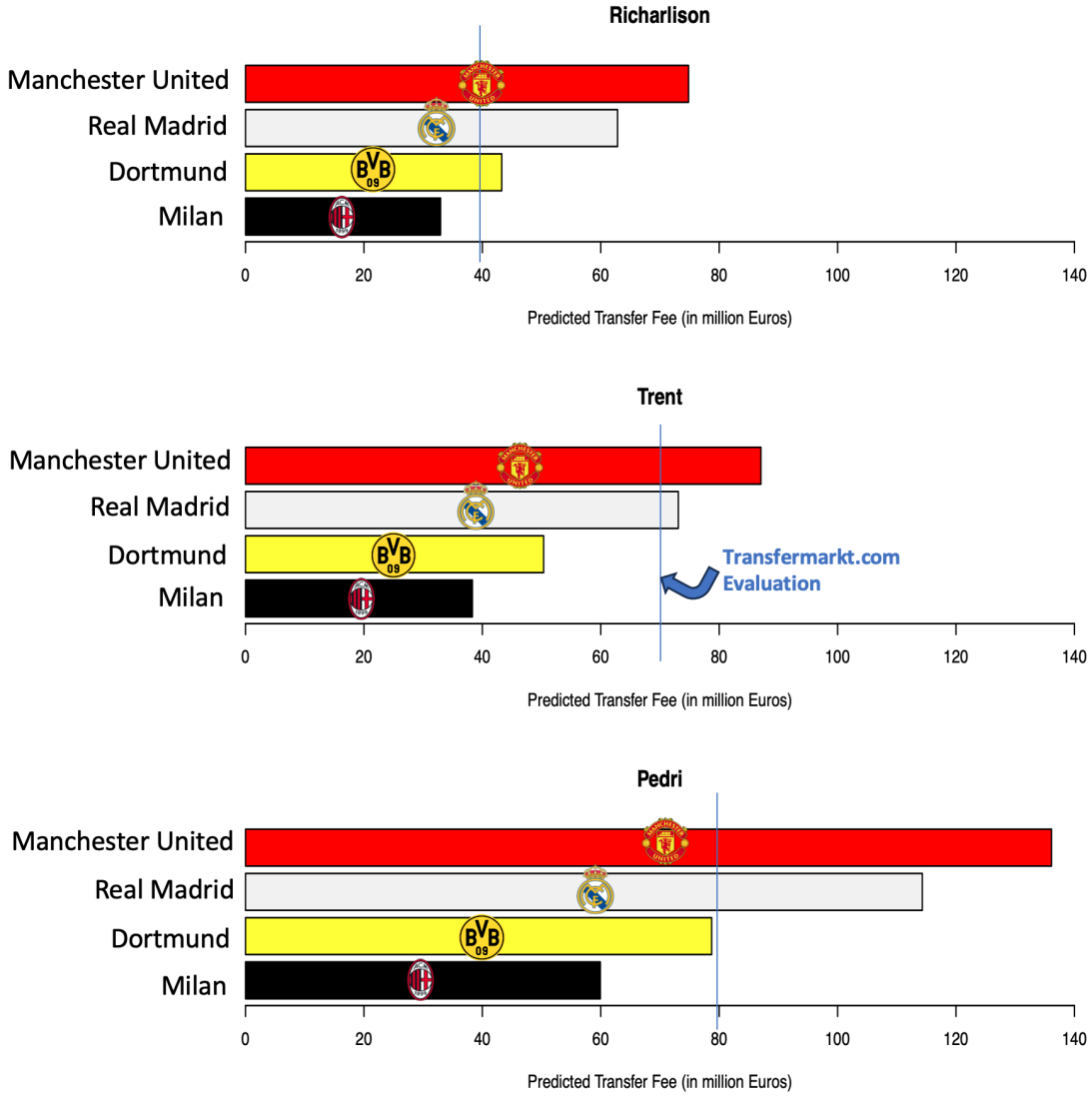


Figure 4.13: Transfer Fee Predictions 2024
Different Leagues predicted to pay different amounts for the same player. The prediction model is predicting the fee a club would pay based on their transfer history and not on how much a player is actually worth. The blue line signifies transfermarkt.com's estimated worth of player. The prediction is that the clubs would pay this much to get a player but in reality if a player is far below the blue line, the current club would be unlikely to sell to this new club.

4.3.2 Model Comparison

Validation is not only for assessing model accuracy but also often used for selecting the best model. To this end, I implemented a decision tree, along with two ensemble machine learning algorithms being a random forest and gradient boosting model to compare their performance against a multiple linear regression model.

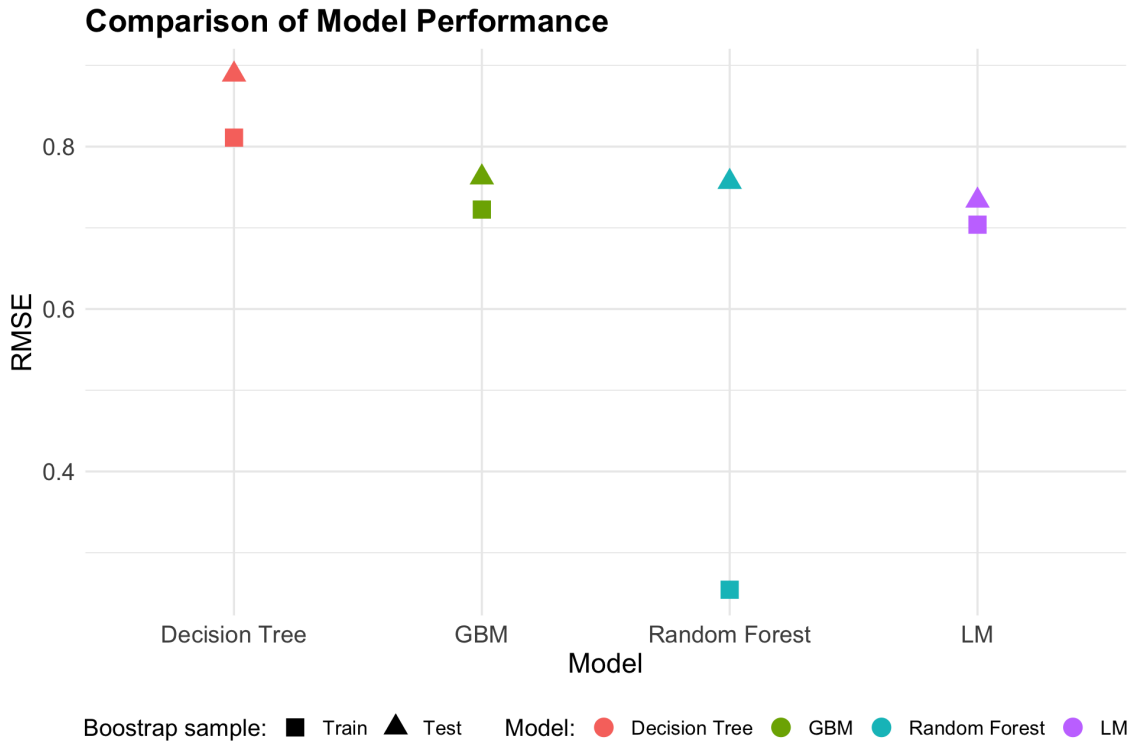
A decision tree splits the data into branches at decision points, making it easy to visualise and interpret the path from inputs to outcomes. Using the `tree` package for regression tree analysis (Ripley, 2019), the decision to split at a particular node is based on finding the split that will result in the largest reduction in the total sum of squared residuals compared to the parent node. The residual for each data point is the difference between the observed value and the mean of the response variable in the node. I resampled the model using the validation method, the bootstrap. The decision tree was resampled 100 times and tested using the OOB bootstrap set. With an OOB mean RMSE of .89 the performance was worse than that of the linear model (.73).

This is unsurprising as decision trees are particularly simple and typically don't perform as well as other models. However they are useful for gaining an overview of the predictors and understanding how different variables contribute to the outcome. To improve a decision tree, a random forest is created through the use of bagging. As discussed in the literature review, this method employs the bootstrap approach to generate multiple random subsets of the original dataset. Each subset is used to train a separate decision tree, and the final output is determined by averaging the results from all the trees in regression. Different predictors are selected for each resampling in the random forest algorithm. This is why it is called an ensemble model (Ripley, 2019). I applied bootstrap validation to the random forest (separate to the bagging), to compare it to the decision tree and linear model's performance.

As can be seen in Figure 4.14, the random forest overfits the data. The training error is extremely low in comparison to all the other error rates (squares on the graph). This is why, for the purpose of validation, it is extremely important to assess the OOB result (Triangle on the graph) as the more appropriate error rate to assess.

Gradient boosting machines (GBM) are an advanced ensemble learning method where weak learners (typically decision trees) are sequentially combined to optimise a loss function, improving accuracy progressively (Friedman, 2001). Unlike bagging methods like Random Forests that build trees independently, GBM focuses on correcting the predecessors' errors, thereby boosting performance incrementally. However, if not managed with parameters such as tree depth and learning rate, there is a risk of overfitting. In this analysis, the *gbm* package in R was utilised to implement the gradient boosting machine algorithm. The performance of the model was similar to the linear model. Since both models perform the same it is best practice to choose the simpler model which us the multiple linear regression model.

Figure 4.14: Bootstrapped Model Performance - RMSE



5 Discussion

5.1 Discussion of Case Study

The linear regression model's stronger performance in predicting transfer fees suggests that, despite the allure of advanced algorithms in machine learning, there remains significant value in considering simpler models that are more interpretable and less prone to overfitting, especially when they align well with the linear nature of the data.

The bootstrap method was chosen as the most suitable internal validation method to perform on the predictions of transfer fee. With the overall results of the validation methods it had the joint best error rate alongside repeated k-fold, MC CV and the standard k-fold CV if we discount LOO CV as we concluded that its error rate was extremely optimistic and unreliable. The adaptability of bootstrapping to small datasets, offered consistent performance even with the intrinsic randomness of the resampling process.

External validation, geographical and independent validation were deemed unsuitable for this case, due to leagues being a predictor in the dataset so therefore using external validation to test for generalisability would have been nonsensical. Without a test for generalisability, we concluded that the model is not generalisable. This decision highlights the importance of understanding the validation strategies with the limitations of the available data and showcases the importance to not pursue irrelevant analysis, saving time and resources.

Temporal validation is the external validation method that could be used when the new data becomes available and is something that will be pursued in the future.

The Random Forest model showed a low error rate on the training data, however, through the use of validation, overfitting was revealed. This highlighted the value of rigorous validation in uncovering such issues. This discovery is a testament to the necessity of thorough validation to ensure that models generalise well to unseen data, rather than merely capturing noise in the training set.

Implementing the predictive model resulted in a 43% improvement over the baseline approach of guessing the average transfer fee. This substantial enhancement not only validates the effectiveness of the predictive modeling approach but also highlights the potential for such models to offer actionable insights in decision-making processes.

As we saw in the predictions, the over prediction of cheap players is an issue with the model. An interesting analysis to be done could be to see how the model performs with transfers starting off at €1 million rather than €100,000. The effect of outliers was also noticed as an issue in the model.

The use of this particular model could see a club using it on their own players and seeing which club they should try to sell to and if an offer came in for a player, they could compare it to the predictions to see if a club is likely to pay more than they are offering,

so that clubs gain a strategic advantage in negotiations.

5.2 Limitations and Recommendations

In evaluating various validation techniques, this study highlights several considerations for practitioners when choosing appropriate methods for model validation. A key conclusion from our analysis is that validation should be done on data that has not been seen by the model during development. However, it is not justifiable to hold data out (split-sample validation) as you could be leaving out valuable information for the building of the prediction model. This traditional method, while straightforward, often fails to capture the variability and potential biases present in real-world data. Instead, we advocate for the use of resampling techniques, which have demonstrated superior ability to estimate model performance.

Our findings indicate that bootstrap methods generally offer higher precision with fewer iterations compared to cross-validation techniques, however, the difference between these methods results will be similar. This efficiency in bootstrap validation is best used in scenarios where computational resources are limited or when quick iterative testing is required. However, the choice between bootstrap and other forms of CV should be guided by specific project needs and the nature of the data.

Moreover, while leave-one-out cross-validation (LOO CV) is comprehensive, our analysis confirms its computational intensity and tendency to produce high variability in results. This makes LOO CV less suitable for large datasets or models that are computationally expensive to train. Practitioners should weigh these factors carefully against the benefits of LOO CV's thoroughness. Though the estimate of accuracy is not reliable, the information gained from LOO CV can still be useful by showcasing outliers.

Another insight from our study involves the use of internal-external validation. This approach, which augments typical external validation methods by altering the standard cross validation method and by splitting by the groups, can provide insights into a model's generalisability across different datasets without the need for an external dataset. Given the challenges of obtaining sufficiently large external datasets, the ability to implement internal-external validation could save time and resources.

In the literature review, temporal validation emerged as an important method in fields that examined time sensitive data, where patterns change over time, such as finance and healthcare. Validating models across different time frames can reveal insights into the model's performance under varying temporal conditions.

In summary, there is no "one-size-fits-all" validation method but examining the data and model is the best way of determining the validation procedure suitable for you.

5.3 Future Directions

As more data becomes available, future research can explore improving predictive modeling in the football transfer market. The existing programming scripts I created for data pre-processing and model validation, facilitates the collection of additional data as it becomes available, enabling for temporal validation. This validation approach will be particularly intriguing to observe the differences between the current dataset and newly acquired transfer data, considering that inflation rates likely do not increase linearly over time. Furthermore, with an expanded dataset, the exploration of more advanced modeling techniques, such as neural networks, could improve the accuracy of football transfer fee predictions.

5.4 Conclusion

The aim of this study was to analyse validation procedures for prediction models using suitable techniques. Our focus was on obtaining error rates to gauge prediction accuracy and interpreting these accuracy's to identify potential instances of overfitting or underfitting the data. Through this analysis, we aimed to provide insights into the reliability of the models, facilitating informed decision-making in their application.

In sport science literature, the validation of predictive models often lacks clear guidance and standardised methods, leaving researchers and practitioners without a definitive framework for evaluating the accuracy and reliability of sporting predictions. While predictive modeling techniques are increasingly employed in sports analytics to forecast various outcomes such as match results, player performance, and injury risk, the validation procedures applied in these studies vary widely and are often inadequately reported.

Unlike fields such as medicine or finance, where well-defined metrics and protocols exist for assessing the validity of predictive models, sport science lacks a universal framework. As a result, researchers may resort to ad-hoc validation approaches, leading to inconsistencies and difficulties in comparing results across studies. To address these challenges, there is a need for efforts to establish standardised validation protocols and guidelines tailored to the unique requirements of sport science research.

The need for appropriate internal, internal-external, and external validation for prediction models has been fully examined and with the exploration of validation procedures complete, this report provides an in depth discussion on the benefits and drawbacks of validation methods which will be extremely beneficial for not only this case study but to new machine learning projects with new datasets that will require appropriate validation.

6 Bibliography

References

Austin, P. C., van Klaveren, D., Vergouwe, Y., Nieboer, D., Lee, D. S., & Steyerberg, E. W. (2016). Geographic and temporal validity of prediction models: Different approaches were useful to examine model performance. *Journal of Clinical Epidemiology*, *79*, 76–85. https://doi.org/https://doi.org/10.1016/j.jclinepi.2016.05.007

Aydemir, A. E., Temizel, T. T., & Temizel, A. (2023). A machine learning ensembling approach to predicting transfer values [Accessed: date]. *Journal Name*. https://blog.metu.edu.tr/ttemizel/files/2022/03/Player_Valuation_accepted.pdf

Berrar, D. (2018, January). Cross-validation. https://www.researchgate.net/publication/324701535_Cross-Validation

Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, *300*, 70–79. https://doi.org/https://doi.org/10.1016/j.neucom.2017.11.077

Clark, T. E. (2004). Can out-of-sample forecast comparisons help prevent overfitting? [Copyright - Copyright Wiley Periodicals Inc. Mar 2004; Document feature - references; equations; Last updated - 2023-11-27; CODEN - JOFODV]. *Journal of Forecasting*, *23*(2), 115–115+. https://www.proquest.com/scholarly-journals/can-out-sample-forecast-comparisons-help-prevent/docview/219183121/se-2

Deloitte. (2023). Annual review of football finance, 8.

Demir, F. (2022). 14 - deep autoencoder-based automated brain tumor detection from mri data. In V. Bajaj & G. Sinha (Eds.), *Artificial intelligence-based brain-computer interface* (pp. 317–351). Academic Press. https://doi.org/https://doi.org/10.1016/B978-0-323-91197-9.00013-8

Depken II, C. A., & Globan, T. (2021). Football transfer fee premiums and europe's big five. *Southern Economic Journal*, *87*(3), 889–908. https://doi.org/https://doi.org/10.1002/soej.12471

Efron, B., & Gong, G. (1983). A leisurely look at the bootstrap, the jackknife, and cross-validation. *The American Statistician*, *37*(1), 36–48. Retrieved April 1, 2024, from http://www.jstor.org/stable/2685844

Efron, B., & Tibshirani, R. (1997). Improvements on cross-validation: The .632+ bootstrap method. *Journal of the American Statistical Association*, *92*(438), 548–560. Retrieved April 4, 2024, from http://www.jstor.org/stable/2965703

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of statistics*, 1189–1232.

Genuer, R., Poggi, J.-M., & Tuleau, C. (2008). Random forests: Some methodological insights.

Harrell, F. E. (2015). *Regression modeling strategies: With applications to linear models, logistic and ordinal regression, and survival analysis* (2nd ed.). Springer International Publishing.

Horvat, T., Havaš, L., & Srpak, D. (2020). The impact of selecting a validation method in machine learning on predicting basketball game outcomes [Available online: https://www.mdpi.com/ (accessed on 7 March 2020)]. *Symmetry*, *12*(431). https://doi.org/https://doi.org/10.3390/sym12030431

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2022). *An introduction to statistical learning: With applications in r* (2nd ed.). Springer.

Jerez, T., & Kristjanpoller, W. (2020). Effects of the validation set on stock returns forecasting. *Expert Systems with Applications*, *150*, 113271. https://doi.org/https://doi.org/10.1016/j.eswa.2020.113271

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, 1137–1143.

Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling*. Springer.

Lista, L. (2023). Machine learning. In *Statistical methods for data analysis: With applications in particle physics* (pp. 225–276). Springer International Publishing. https://doi.org/10.1007/978-3-031-19934-9_11

McHale, I. G., & Holmes, B. (2023). Estimating transfer fees of professional footballers using advanced performance metrics and machine learning. *European Journal of Operational Research*, *306*(1), 389–399. https://doi.org/https://doi.org/10.1016/j.ejor.2022.06.033

Mitchell, W. G., Dee, E. C., & Celi, L. A. (2021). Generalisability through local validation: Overcoming barriers due to data disparity in healthcare. *BMC Ophthalmology*, *21*(1), 228. https://doi.org/10.1186/s12886-021-01992-6

Molinaro, A. M., Simon, R., & Pfeiffer, R. M. (2005). Prediction error estimation: A comparison of resampling methods. *Bioinformatics (Oxford, England)*, *21*(15), 3301–3307. https://doi.org/10.1093/bioinformatics/bti499

Olive, D. J. (2017). *Linear regression* (1st ed.) [57 b/w illustrations]. Springer Cham. https://doi.org/10.1007/978-3-319-55252-1

Park, M. Y., & Hastie, T. (2007a). L1-Regularization Path Algorithm for Generalized Linear Models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *69*(4), 659–677. https://doi.org/10.1111/j.1467-9868.2007.00607.x

Park, M. Y., & Hastie, T. (2007b). L1-Regularization Path Algorithm for Generalized Linear Models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *69*(4), 659–677. https://doi.org/10.1111/j.1467-9868.2007.00607.x

Poli, R., Ravenel, L., & Besson, R. (2023). Inflation in the football players' transfer market (2013/14-2022/23). *Journal of Sports Economics*, *20*(10), XX–XX.

Ramspek, C. L., Jager, K. J., Dekker, F. W., Zoccali, C., & van Diepen, M. (2020). External validation of prognostic models: what, why, how, when and where? *Clinical Kidney Journal*, *14*(1), 49–58. https://doi.org/10.1093/ckj/sfaa188

Ripley, B. (2019). *Tree: Classification and regression trees* [R package version 1.0-40]. https://CRAN.R-project.org/package=tree

Royston, P., Parmar, M. K. B., & Sylvester, R. (2004). Construction and validation of a prognostic model across several studies, with an application in superficial bladder cancer. *Statistics in Medicine*, *23*(6), 907–926. https://doi.org/https://doi.org/10.1002/sim.1691

Sobol, M. G. (1991). Validation strategies for multiple regression analysis: Using the coefficient of determination. *Interfaces*, *21*(6), 106–120. Retrieved April 24, 2024, from http://www.jstor.org/stable/25061557

Steyerberg, E. (2009, January). *Clinical prediction models: A practical approach to development, validation, and updating* (Vol. 19). https://doi.org/10.1007/978-0-387-77244-8

Steyerberg EW, J. C. E., Harrell FE Jr. (2015). Prediction models need appropriate internal, internal-external, and external validation. https://doi.org/doi:10.1016/j.jclinepi.2015.04.005

Takada, T., Nijman, S., Denaxas, S., Snell, K. I., Uijl, A., Nguyen, T.-L., Asselbergs, F. W., & Debray, T. P. (2021). Internal-external cross-validation helped to evaluate the generalizability of prediction models in large clustered datasets. *Journal of Clinical Epidemiology*. https://doi.org/https://doi.org/10.1016/j.jclinepi.2021.03.025

Twomey, J., & Smith, A. (1999). Validation and verification. *Artificial Neural Networks for Civil Engineers: Fundamentals and Applications*, 2–5. https://www.researchgate.net/publication/2576791_Validation_and_Verification

Wainer, J., & Cawley, G. (2021). Nested cross-validation when selecting classifiers is overzealous for most practical applications. *Expert Systems with Applications*, *182*, 115222. https://doi.org/https://doi.org/10.1016/j.eswa.2021.115222

Wooldridge, J. M. (2012). *Introductory econometrics: A modern approach* (5th ed.). South-Western Cengage Learning.

Xu, Q.-S., & Liang, Y.-Z. (2001). Monte carlo cross validation. *Chemometrics and Intelligent Laboratory Systems*, *56*(1), 1–11. https://doi.org/https://doi.org/10.1016/S0169-7439(00)00122-2