Bachelor's thesis

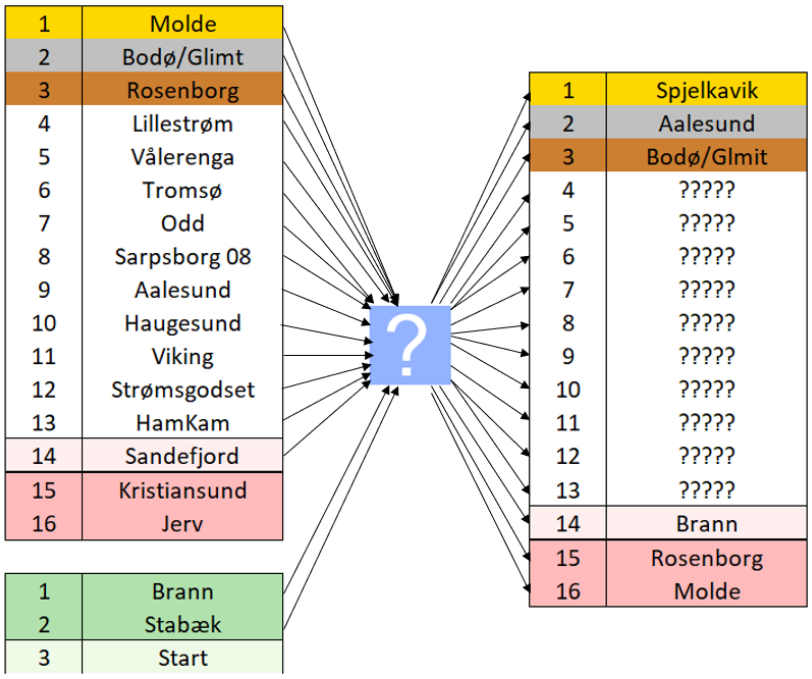
NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

Martin Valderhaug Larsen

Predicting Eliteserien using Regression Models

Predicting the Unpredictable

Bachelor's thesis in Mathematical Sciences
Supervisor: John Sølve Tyssedal
Co-supervisor: Øyvind Salvesen
June 2023



NTNU
Norwegian University of Science and Technology

Martin Valderhaug Larsen

Predicting Eliteserien using Regression Models

Predicting the Unpredictable

Bachelor's thesis in Mathematical Sciences
Supervisor: John Sølve Tyssedal
Co-supervisor: Øyvind Salvesen
June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

NTNU
Norwegian University of
Science and Technology

Preface

This thesis concludes my bachelor's degree in Mathematical Sciences with a specialization in statistics at NTNU. It has been some relaxing years in Trondheim.

Firstly I want to thank my Mom and Dad for putting me in the position to be where I am today. And, of course, the rest of my family. A special thanks to all my friends in Trondheim who have made this journey enjoyable. In particular, all the lovely people from the "kollektiv", my studies, and Realfagskjelleren.

Thanks to my co-supervisor Øyvind Salvesen despite not having any obligations to do so, took this role with his great statistical knowledge in sports. Maybe our ways meet at a later point.

Next up, my supervisor John Tyssedal, the guy who started his career with computer punch cards, and ending it with ChatGPT. This says all about his commitment and passion for the field of statistics. Having spent most of our meetings talking about statistics and football, I am grateful for his guidance throughout the project. Enjoy your days as a hobby statistician!

A special thanks to Bård Fineid for going through the statistical courses with me. Helping me, and contributing greatly towards "our" projects. Good luck in Oslo, hope to see you there.

Thanks to Arild Johnsen for his collaboration on my projects through my Bachelor's degree. Especially through our game development. An honorable mention to when you and Jon Marius Ustad visited me in Trondheim.

Thanks to Håvard Pettersen for being my regexpert throughout the hours of data collection. Good luck at Microsoft! Looking forward to Halloween!

Thanks to Elias Mathiesen for being my "die-hard football fan", and to Halvor Bergstøl Birkeland for his emotional and technical support.

Thanks to Fredrik Reite for being my remote PT, motivating me, and making sure my health stays on top.

Thanks to everyone who has proofread the thesis. Jonas Fladmark, Clara Laboureau-Grande, Maximilian Rønseth, Edvard C.B Schøyen, Iver Kjelsaas, Adrian Rydell, Karine Halleraker's mom, Tiril Fossheim and Maren Livelten.

Abstract

In this thesis, we attempt to predict Eliteserien using multiple linear regression and generalized linear models. We begin by finding two easily interpretable models that can be understood and used by a normal football supporter. Next, we use Poisson regression to find a model that predicts as well as possible. In the end, we predict the final table for Eliteserien 2023.

Sammendrag

I denne avhandlingen forsøker vi å forutsi Eliteserien ved hjelp av flere lineære regresjonsmodeller og generaliserte lineære modeller. Vi starter med å finne to lett forståelige modeller som kan brukes av en vanlig fotballsupporter. Deretter bruker vi Poisson-regresjon for å finne en modell som gir best mulig prediksjonsevne. Til slutt predikerer vi sluttresultatet for Eliteserien 2023.

Oversatt av ChatGPT 3.5

Vi tar ikke ansvar for eventuelle grammatikkfeil som kan forekomme.

Contents

List of Figures

Introduction

When I started this project, I knew that I would either become rich or write a bachelor thesis. And here we are.

1.1 Background

Growing up with my dad and brother being devoted to football, I just followed along playing for my local team `Spjelkavik IL`. With me being most interested in having fun and not so much in training to become the best, I in my later years became more known for my yellow cards, and scoring only once. However, football is a team sport, and only one red card, a broken glass, and a handful of yellow cards later, I became a regional champion twice. At the same time, I also fell in love with lower-division Norwegian football, especially Spjelkavik. Because of this, I spent much of my time as a speaker at their games, starting to look into football data. At some point, I found a way to gain access to hidden information on fotball.no [1], but because of my ethical principles, I did not sell this information. Moving away from Spjelkavik to Trondheim for my studies, I stopped my data collection.

Two years later, this project began as an argument between my dad and my brother. They argued about which of their football achievements were the best. This ended in me collecting a bunch of Norwegian football data to settle the argument, ultimately letting my dad down. Having procrastinated the data collection for multiple hours, I had to turn it into a bachelor's thesis.

There was an urban myth on the institute (IMF) about a guy called `Øyvind Salvesen`, who had made a career out of predicting different sports results. Being eager to learn with a bright mind, took a long shot and phoned him up expecting him to not have time. To my shock, he actually found it really interesting and wanted to join as a mentor.

To build a solid team, I contacted one of the most acknowledged professors at NTNU `John Tyssedal`. With him onboard, everything was settled. Having created a diversified team of two professors who could have been my dad and my granddad, I felt confident in our abilities. Suddenly had no excuse for not producing a solid thesis.

1.2 Football and Eliteserien

Football is an easy sport, whoever scores the most goals after 90 minutes, wins the game and takes home 3 points. In the case of a tie, they received one point each. In a way, points are a function of scored and conceded goals. Teams compete in different formats. One way is through a cup, where teams play through knock-out stages, eliminating each other until one team is left as champion. However, the most common and recognized format is the league system. One such league is Eliteserien, which is the top Norwegian football league. The league consists of 16 teams playing each other twice for a total of 30 games each. At the end of the season, the team with the most points becomes the champion, while 2 or sometimes 3 teams are relegated to the lower division. This also means that 2-3 teams are promoted to Eliteserien each year.

1.3 The Data

To the best of our knowledge, there is no fully existing dataset on Eliteserien teams. When starting the thesis, the Norwegian Football Association was asked politely for the data. However, they were not too happy about giving the data away. In a desperate attempt to save time, we contacted the Rec Sport Soccer Statistics Foundation (RSSSF)[2]. This is an international organization dedicated to collecting statistics about football. However (again), they had no way of extracting the data into a simple file format, which the author found quite hilarious, with them dedicating their time to collecting non-usable data. But as we say in Norway: "he who laughs last laughs best" [3], and the author eventually had to sacrifice multiple nights of sleep to collect the data manually. This part of the thesis should be worth a bachelor's degree in monkey work.

The dataset contained data from teams in Eliteserien. The data spans from 2000 to 2019 containing a total of 301 observations of teams, where each team has 20 variables connected to their performance in the current season and the two previous seasons. The dataset was self-collected through different sources and structured using Python and SQL.

The results from the league and cup were gathered through the home page of NFF [1] but were also cross-referenced with RSSSF. This action was taken to reduce the chance of big errors in the data.

To find the results from the European competitions, the official website of UEFA (the Union of European Football Associations) [4] was used as a source. With the history of the competition being well documented, the data here is believed to be accurate.

The coach changes are based on a self-collected dataset containing all coaches for the Eliteserien teams from 1990 until today. These data were collected mainly through the website of Transfermarkt [5], with them having most of the data. "Only" the missing periods had to be filled out. Unfortunately, for the sleep schedule, it turned out that pre-Internet information on coaches is not the easiest to find. For every missing period of a club, different sources of old papers had to be manually researched. In addition, questions were asked on different forums. This could lead to errors in the data, but it can be argued that the football geeks

on these forums know what they talk about and that double-checking the data would make them trusted.

1.4 Outline

Our mission is to find a few models to predict Eliteserien. Chapter 2 introduces the Poisson distribution, multiple linear regression, and generalized linear models. Chapter 3 provides an overview of the data and analyzes and interprets some of the data. Chapter 4 justifies the model choices while providing a detailed explanation of how to proceed with the analysis. The fun begins in Chapter 5, where the results are presented and analyzed, and lastly, Eliteserien 2023 is predicted. Finally, in Chapter 6, we conclude our findings while leaving some concluding remarks.

Theory

When choosing my bachelor project, I wanted a project with minimal theory. I chose to analyze football results using regression thinking: " There cannot be much theory to write about this topic". ..... I was wrong.

2.1 The Poisson Distribution

The Poisson Distribution is a discrete probability distribution, which gives the probability of an event occurring k times in a given time interval. The probability mass function (PMF) is given by

$$f(k;\lambda) = \Pr(X{=}k) = \frac{\lambda^k e^{-\lambda}}{k!}, \quad \lambda > 0, k \in \{0, 1, 2 \dots\} \tag{2.1}$$

where X is the number of events in the time interval and λ is the expected number of events in the given interval.
The distribution has multiple interesting properties, with the most central being

$$\lambda = E(X) = Var(X) \tag{2.2}$$

It is also only defined for positive values of k and is best suited for situations in which events do not occur frequently. In the case of a large λ, the Poisson distribution converges towards a normal distribution. This follows from the central limit theorem.

2.1.1 The Distribution of Goals

When trying to predict the number of goals, the most fundamental goal is to determine the distribution that they follow. With many researchers conducting research on different sports that involve scoring, there is a common conception that the number of goals will follow a Poisson distribution. This is supported by the Ph.D. thesis of Øyvind Salvesen [6]. A study of the English football league also supports that this is the case for football [7]. With this in mind, it's reasonable to assume that the number of goals X_i scored in game i by a team is Poisson distributed with

$$X_i \sim pois(\lambda_i)$$

where λ_i is the expected number of goals scored by the team in game i.
Assuming independence between games, the total number of goals during a season

will also follow a Poisson distribution. Since the number of goals X scored in a season with n games is given by

$$X = \sum_{i=1}^{n} X_i, \quad \text{we have} \quad X \sim pois(\sum_{i=1}^{n} \lambda_i)$$

which follows from the additive property of the Poisson distribution

$$X \sim pois(\lambda_x) \quad \text{and} \quad Y \sim pois(\lambda_y) \implies X + Y \sim pois(\lambda_x + \lambda_y).$$

2.2 Multiple Linear Regression

If you are still following, we assume that you have a decent understanding of Multiple Linear Regression (MLR). If not, hey Mom and Dad.

In short, MLR is an extension of the simple one-variable regression model

$$y = ax + b + \epsilon.$$

In this case, we allow for one input value x, called a predictor. To allow for p predictors, we introduce the form

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \epsilon. \tag{2.3}$$

Adding the error term ϵ often called residuals, which is assumed to be normally distributed with mean 0 and variance σ^2. [8]

Given a set of n observed predictors and response values

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ 1 & x_{n1} & \cdots & \cdots & x_{np} \end{bmatrix} \quad \text{and} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

with each row $\mathbf{x}_i \in \mathbf{X}$ corresponding to the response value y_i, we want to estimate the values $\beta_0, \beta_1 \ldots \beta_p$ such that the model in equation 2.3 fits the data as best as possible.

2.2.1 Least-squares Estimation

One way of estimating our β_i's is through least-squares estimation. We define the best fit as the one that minimizes the residuals. Note that for each $\mathbf{x_i}$ and y_i we have

$$y_i = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_p x_{1p} + \epsilon_i. \tag{2.4}$$

We can therefore write

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

where $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_n)'$ and all ϵ_i are assumed to be independent. One way of minimizing the residuals is trough minimizing the squares of the residuals

$$\boldsymbol{\epsilon}'\boldsymbol{\epsilon} = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}).$$

This is minimized with

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}. \tag{2.5}$$

(see Appendix C.1). This implies that the coefficient estimates can be found using a linear system.

2.2.2 Model Evaluation

We also needed to evaluate the fitted model. This involves checking different assumptions and measures to determine whether our model sufficiently fits our data.

Significance of the Regression Coefficients Given the estimated coefficients $\hat{\boldsymbol{\beta}}$, we need a method to determine whether each β_i is non-zero. Even though our estimated $\hat{\beta}_i$'s are non-zero, there is a chance that this is due to randomness. Therefore, we must find a way to determine if the estimates are nonzero. Given our estimate $\hat{\beta}_i$, we can set up the hypothesis test

$$H_0 : \beta_i = 0, \quad H_1 : \beta_i \neq 0.$$

This can be checked with the test statistic

$$T = \frac{\hat{\beta}_i}{se(\hat{\beta}_i)}$$

where $se(\hat{\beta}_i)$ is the standard deviation of $\hat{\beta}_i$ which can be found on the diagonal of the matrix $\hat{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1}$. Using a Student's t-distribution, the p-value can be computed and compared to a given significance level [9, p. 482]. These computations are automatically performed in the `lm()` function in R.

Coefficient of determination The most common way of measuring the goodness of fit for an MLR is through the coefficient of determination

$$R^2 = \frac{RSS}{TSS} = 1 - \frac{SSE}{TSS} = 1 - \frac{\sum_{i=1}^{n} \epsilon_i^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}$$

[10, pp. 419–427]. R^2 provides the proportion of the variation explained by the model, ranging from 0 to 1, with 1 indicating a perfect fit.

Residual Plot In MLR, we assume that the residuals ϵ_i are normally distributed. This assumption can be checked with the help of a residual plot, which plots the estimated values $\hat{y}_i$ against

$$\epsilon_i = y_i - \hat{y}_i. \tag{2.6}$$

If these assumptions hold, the points should be equally distributed around $y = 0$.

Quantile–Quantile Plot Quantile–Quantile Plot is another way of checking the residuals. The quantiles of the two distributions are plotted against each other. If the distributions are similar, the points lie on the line $y = x$ [11, p. 37]. To evaluate whether our residuals follow a normal distribution, we standardize the residuals and plot them against the standard normal distribution $\mathcal{N}(0, 1)$. If the points form a line close to $y = x$, then it is reasonable to assume that the residuals are normally distributed.

2.3 Maximum Likelihood Estimation

Maximum likelihood estimation (MLE) is a method of estimating a set of unknown parameters $\boldsymbol{\theta}$ for an assumed PMF. To see how this works, we start with a random sample $X_1, X_2, \dots X_n$, with each X_i having a PMF $f(k_i; \boldsymbol{\theta})$. Using the fact that the X_i's are independent, the joint PMF also called the likelihood function $L(\boldsymbol{\theta})$ becomes

$$L(\boldsymbol{\theta}) = P(X_1 = k_1, X_2 = k_2 \dots X_n = k_n)$$
$$= f(k_1; \boldsymbol{\theta}) \cdot f(k_2; \boldsymbol{\theta}) \dots f(k_n; \boldsymbol{\theta}) = \prod_{i=1}^{n} f(k_i; \boldsymbol{\theta})$$

Then, we find the $\hat{\boldsymbol{\theta}}$ that maximizes L by using derivatives.

2.3.1 MLE for Poisson Distribution

The maximum likelihood estimator of the Poisson distribution comes up a handful of times in different metrics and regressions. Therefore, it is useful to understand this technique. Recalling the PMF from 2.1, we get

$$L(\lambda) = \prod_{i=1}^{n} f(k_i; \lambda) = \prod_{i=1}^{n} \frac{\lambda^{k_i} e^{-\lambda}}{k_i!} \tag{2.7}$$

The $\hat{\lambda}$ that maximizes L, is the same as the one that maximizes $ln(L)$. Thus we can find the maximum likelihood estimator by solving

$$ln(L(\lambda)) = 0.$$

We have

$$ln(L(\lambda)) = ln(\prod_{i=1}^{n} \frac{\lambda^{k_i} e^{-\lambda}}{k_i!})$$

$$= \sum_{i=1}^{n} ln(\lambda^{k_i}) + \sum_{i=1}^{n} ln(e^{-\lambda}) - \sum_{i=1}^{n} ln(k_i!) = ln(\lambda) \sum_{i=1}^{n} k_i - n\lambda - \sum_{i=1}^{n} ln(k_i!)$$

giving

$$\frac{\partial}{\partial\lambda} ln(L(\lambda)) = \frac{1}{\lambda}\sum_{i=1}^{n} k_i - n = 0 \implies \hat{\lambda} = \frac{1}{n}\sum_{i=1}^{n} k_i = \bar{k}.$$

Thus the maximum likelihood estimator is just the sample average. This should give the reader a basic understanding of how the maximum likelihood can be used to estimate parameters, and later how we extend it to generalized linear models.

2.4 Generalized Linear Models

In Section 2.2, we assume that the response is linear in the coefficients. However, this is insufficient in many cases. Therefore, the concept of generalized linear models (GLM) is introduced to provide more flexibility in building a model. With the GLM, we can capture nonlinear relationships between the response and the predictors with the help of a link function. Where MLR assumes that the residuals follow a normal distribution, GLM allows the residuals to take other distributions in the exponential family. To understand this concept, we introduce Poisson regression.

2.4.1 Poisson Regression

Poisson Regression is a GLM model used for count data. It assumes that the response follows a Poisson distribution. Having already discussed that the number of goals follows a Poisson distribution, Poisson regression is a good candidate for predicting goals.

2.4.2 Link Functions

GLM captures a nonlinear relationship using link functions. Formally a link function g relates the distribution mean $\boldsymbol{\mu}$ to a linear predictor $\boldsymbol{X\beta}$, such that for each $\mu_i \in \boldsymbol{\mu}$ we can write

$$g(\mu_i) = \mathbf{x_i}'\boldsymbol{\beta}$$

The most trivial link function is the identity link function $g(\mu_i) = \mu_i$, which indicates that a linear relationship already exists with $\boldsymbol{\mu} = \mathbf{X}\boldsymbol{\beta}$. For Poisson regression, we often use the log-link function $g(\mu_i) = log(\mu_i)$. Saying that μ_i's can be related with $\mu_i = e^{\mathbf{x}_i'\boldsymbol{\beta}}$. With the Poisson distribution allowing only non-negative values, the log-link function is often used to guarantee that only positive values are passed.

2.4.3 MLE in Poisson Regression

Earlier MLE was used to fit a Poisson distribution to an independent random sample. By extending this concept, a GLM model can be fitted. Using the theory from [12], we start with a modified version of equation 2.7

$$L(\boldsymbol{y};\boldsymbol{\beta}) = \prod_{i=1}^{n} \frac{\mu_i^{y_i} e^{-\mu_i}}{y_i!}$$

with μ_i being connected with the log-link function. Using this we get

$$ln(L(\boldsymbol{y};\boldsymbol{\beta})) = ln(\prod_{i=1}^{n} \frac{\mu_i^{y_i} e^{-\mu_i}}{y_i!}) = \sum_{i=1}^{n}(-e^{\mathbf{x_i}'\boldsymbol{\beta}} + y_i \mathbf{x_i}'\boldsymbol{\beta} - ln(y_i!)).$$

Recalling how to do multi-dimensional derivatives

$$\frac{\partial}{\partial \boldsymbol{\beta}} ln(L(\boldsymbol{y};\boldsymbol{\beta})) = \sum_{i=1}^{n}(y_i \mathbf{x_i} - e^{\mathbf{x_i}'\boldsymbol{\beta}}\mathbf{x_i}).$$

Lastly setting the derivative to zero, we obtain

$$\sum_{i=1}^{n}(y_i - e^{\mathbf{x_i}'\boldsymbol{\beta}})\mathbf{x_i} = \mathbf{0}.$$

This is clearly not a linear system when solving for $\boldsymbol{\beta}$. Fortunately, we can exploit the fact that $-L(\boldsymbol{y};\boldsymbol{\beta})$ is a convex function and solve it with a convex optimization technique such as gradient descent [13]. It is worth noting that solving such a system is more complex than solving a linear system in MLR. These techniques are implemented in different statistical software, and we will not go deeper into them.

2.4.4 Model Evaluation of Poisson Regression

As in Section 2.2.2, we also need a way to evaluate whether different assumptions and measures for the GLM to sufficiently fit our data. For simplicity, the focus is on how to do this for Poisson regression.

Significance of the Regression Coefficients As with MLR, GLM also needs a way of determining whether each β_i is non-zero. This becomes harder for GLM due to the method of solving for the $\hat{\beta}_i$'s. However, there are advanced methods for computing the variance. Using this and the fact that the β_i's follow a normal distribution, the significance is obtained with the test statistic

$$Z = \frac{\hat{\beta}_i}{se(\hat{\beta}_i)}.$$

This is automatically computed in the `glm()` function in R.

Pearson Residual Plot For MLR, we plotted the residuals to determine whether the residuals were normally distributed. In the case of Poisson regression, we expect the residuals to increase as the estimated $\hat{\mu}_i$ increases. This follows from the variance being equal to the expected value (Equation 2.2). To fix this and get residuals that do not depend on $\hat{\mu}_i$, we introduce the Pearson residuals

$$r_i = \frac{y_i - \hat{\mu}_i}{\sqrt{\hat{\mu}_i}}$$

simply by dividing by the standard deviation of the estimated $\hat{y}_i$ to cope with the variance increase [14, p. 37]. We can then plot the Pearson residuals as with the residual plot in MLR using the obtained r_i's.

Pearson χ^2 goodness-of-fit test Because GLM uses MLE instead of least squares, we have to get a bit more creative when measuring the goodness-of-fit of our model. Overall, there are no good or standard ways of doing this, but there are ways to get an idea of the fit, which can be done through the Pearson χ^2 goodness-of-fit test. This is obtained by rewriting the original way of doing the goodness-of-fit test

$$\chi^2 = \sum_{i=1}^{n} r_i^2 = \sum_{i=1}^{n} \frac{\epsilon_i^2}{\hat{\mu}_i} \sim \chi^2_{n-p} \tag{2.8}$$

[14][15]. If we reject H_0, then there is evidence that the model does not fit. Similarly, if H_0 is not rejected, there is no evidence to claim that the model does not fit our data. A test alone will not give a solid decision basis for the model fit.

Quantile-Quantile Plot For the residual plot, we must also adjust the residuals for our Q-Q plot. One way would be to standardize the Pearson residuals and plot them in the same way as the Q-Q plots in MLR. However, the built-in `plot()` function in R employs a similar approach. Instead of Pearson residuals, deviance residuals were used. The deviance residuals adjust the residuals to avoid bias toward higher estimates $\hat{y}_i$, similar to the Pearson residuals. More information about deviance residuals can be found in [14, p. 39].

2.5 Performance Measures

We can choose from a few performance measures when evaluating the performance of a model. One common way of measuring total error is the root squared mean error, but some might prefer the use of mean absolute error because it is easier to interpret.

Mean Absolute Error is a method for measuring the total mean difference between predicted and actual values.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |\hat{y}_i - y_i|$$

Root Mean Squared Error is another method of measuring the total difference between the predicted and the actual values.

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

RMSE will be more affected by large residuals than MAE.

AIC and BIC Even though our previous measures provide a good estimation of the fit of the data, they will be biased towards adding more variables to the model, potentially over-fitting the model. Therefore, we introduce AIC and BIC to penalize the addition of variables and provide a less biased method of selecting variables. The Akaike information criteria and Bayesian information criteria are both ways of estimating the optimal model, given by

$$\text{AIC} = 2p - 2ln(\hat{L}) \quad , \quad \text{BIC} = p \cdot ln(n) - 2ln(\hat{L})$$

where $\hat{L}$ is the maximized value of the likelihood function [16] [17] and p is the number of predictors used in the model. Both criteria find their most suited model by being minimized, and BIC will be stricter on adding more predictors.

Mallows C_p is defined for the purpose of doing performance tests on MLR.

$$C_p = \frac{SSE}{\hat{\sigma}^2} - n + 2p$$

where $\hat{\sigma}^2$ is the variance estimator [9, p. 500]. In MLR, C_p and AIC are proportional, allowing C_p to replace AIC in the error plots.

2.5.1 Cross Validation

With measures such as AIC and BIC, we have solid information on which and how many predictors to include in a model, but it does not provide a concrete measure of how the model actually performs on the data. We would like to use the RMSE or MAE to evaluate the model without worrying about bias toward the addition of predictors. The most basic way to do this is to leave a randomly sampled test set out of the model training data and then use the test set to evaluate the model. Unfortunately, this approach requires a greater amount of data to obtain accurate and stable test results.

To get more precise test data, we can use cross-validation. Different variations of cross-validations exist, with the main variation being k-fold cross-validation. Starting by dividing the data into k subgroups, the model is trained on all but one subgroup. With the model never having seen these test data, we avoid the possibility of overfitting. By repeating this process for each subgroup as the test group and averaging the error, we obtained a representative test measure.

2.5.2 Model Selection

With many methods for measuring the performance of a model, model selection may sound trivial. The model with the most optimal test results is selected. This would be true if we could check all possible models, but the number of possible models grows ridiculously fast toward high numbers. If you do not have infinite time or computing power, different strategies may be required for model selection. In our analysis, only the brute force approach is needed.

Best subset selection is the idea that we just introduced. With k different predictors, for each possible model of size i, we check all the $\binom{k}{i}$ possible ways of

making a model and choose the model with the best measure. After performing this for all possible model sizes i, we compare them using different measures to obtain the best model size. To give an idea of how fast this becomes impossible, the number of models we have to check for k predictors is

$$\binom{k}{0} + \binom{k}{1} + \cdots + \binom{k}{k} = \sum_{i=0}^{k} \binom{k}{i} = 2^k.$$

This implies that we have an exponentially increasing computation time. If the reader still does not understand how big that is, say, we have 20 predictors, we would have to check $2^{20} = 1.048.576$ different models.

This brute-force search method is formally called exhaustive search and can be computed for linear models and GLMs with the R packages `leaps` [18] and `bestglm` [19] respectively.

The Data

My supervisor said that I was allowed to start all chapters with some out-of-context paragraph as long as it was in an italic font. With it being well over a year since starting this data collection, I am excited to finally see some visual plots.

3.1 Data Formatting

With all raw data collected from the different datasets, the next task was formatting. After connecting every dataset into a SQL database [20], formatting became a matter of connecting specific queries to obtain the correct data for each observation. For each team participating in an Eliteserien season, we checked the previous year's results, both domestic and European cup results, and also the coach timeline of the team. The relevant information for the team season was stored as one observation in the final dataset. As an example, we can consider the Aalesund FK 2010 season.

Table 3.1.1: Aalesunds FK 2010.

year	tos	toc	plc	poi	eur	eur_1lag	cup_1lag	cc	cc_1lag
2010	46	37	4	47	1	0	1	0	0

tos_1lag	toc_1lag	plc_1lag	poi_1lag	div_1lag
34	43	13	36	0
tos_2lag	**toc_2lag**	**plc_2lag**	**poi_2lag**	**div_2lag**
33	48	13	29	0

A full explanation of the variables can be found in Appendix A. From the Table, it can be observed that Aalesund scored 46 and conceded 37 goals, placing 4th. They also are playing qualification for a European tournament, came through to the quarter-final in the domestic cup the year before, and had no coach changes. Looking at earlier years league information would not show great achievements.

3.1.1 Response and Predictors

It is important to clarify the variables that can be used as predictors. By predicting the results at the start of the current season, we will not have access to most of the current season data. The total scored goals, total conceded goals, placement, and points cannot be accessed because the season has not ended from our perspective.

All the other predictors can be classified by how many years they go back. It is worth noting that both year and whether a team is playing European qualifiers is known at the start of the season, and can therefore be used as a predictor. We classified the variables into four groups.

Table 3.1.2: Variable grouping

	Response	**Predictors**	**Predictors (_1lag)**	**Predictors (_2lag)**
Number of variables	4	3	8	5
Variables	tos, toc, plc, poi	year, eur, cc	tos, toc, plc, poi, div, cup, eur, cc	tos, toc, plc, poi, div

Some might question why there are fewer predictors going two years back; this has to do with the fact that the _2lag predictors were added at a later stage of the project. Therefore, using the time to collect more predictors from two years back was not prioritized. One could also argue that they are not important based on the importance of their _1lag counterparts.

3.1.2 Data Modification

Adjusting for less games Thirty games were played in the current format of the top two divisions. This removes any worries about adjusting for games played for the current divisions but does not account for different formats in the earlier years of the divisions. Therefore, the data needed to be adjusted according to the number of games played. This is solved by scaling all relevant variables to correspond to 30 games. For simplicity and to use Poisson regression, the new values were rounded to the nearest integer.

For example, Rosenborg's 71 goals scored over 26 games in 2001 (Appendix B), which corresponds to 82 goals today, with

$$\frac{71 \cdot 30}{26} = 81.923... \approx 82.$$

The Removal of Fredrikstad 2005 season When looking at the results going two years back, there is a theoretical possibility that a team achieves promotion twice from the 3rd to the 2nd tier, and straight to the 1st tier. The dataset containing only data from the top two tiers could lead to the problem of not having data on such a team. To solve this problem, the naive approach was taken assuming that this could not occur.

Later, it was found that this was exactly what happened to Fredrikstad. Promoting from tier 3 in 2003, promoting immediately from tier 2 in 2004, and ending in Eliteserien (tier 1) in 2005. With this in mind, possible changes in the dataset were discussed, but with Fredrikstad being the only team coming from tier 3, their observation was deemed unnecessary because it had a minimal impact on the model. The Fredrikstad 2005 season was therefore dropped from the dataset.

3.2 Data Overview

To obtain an overview of the dataset, the correlation matrix is plotted

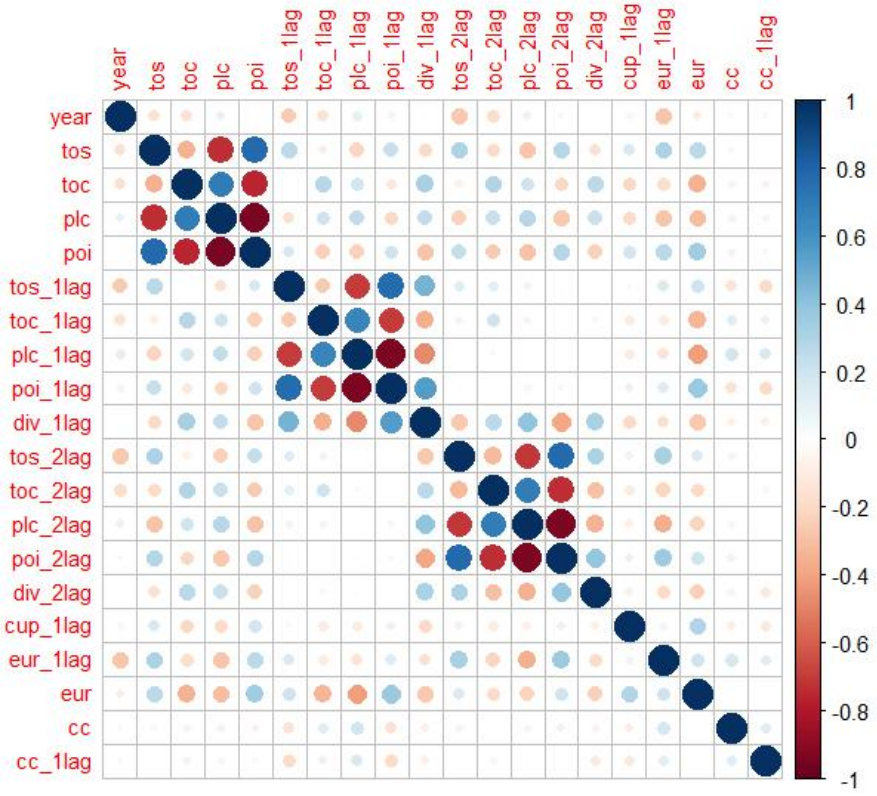


Figure 3.2.1: Correlation plot of all variables.

Considering each year's goals, points, and placements, it is not surprising that they are highly correlated. It is also observed that their lags correlate with our possible responses, providing hope for future predictions. For the European tournament predictors, we find a decent correlation with league results. With most European tournament spots being a result of league placement, this is also expected. For the domestic cup and coach changes, there is minimal to no correlation with the other variables. Again, with the author having spent many hours collecting the coach dataset, as of writing, it was quite annoying.

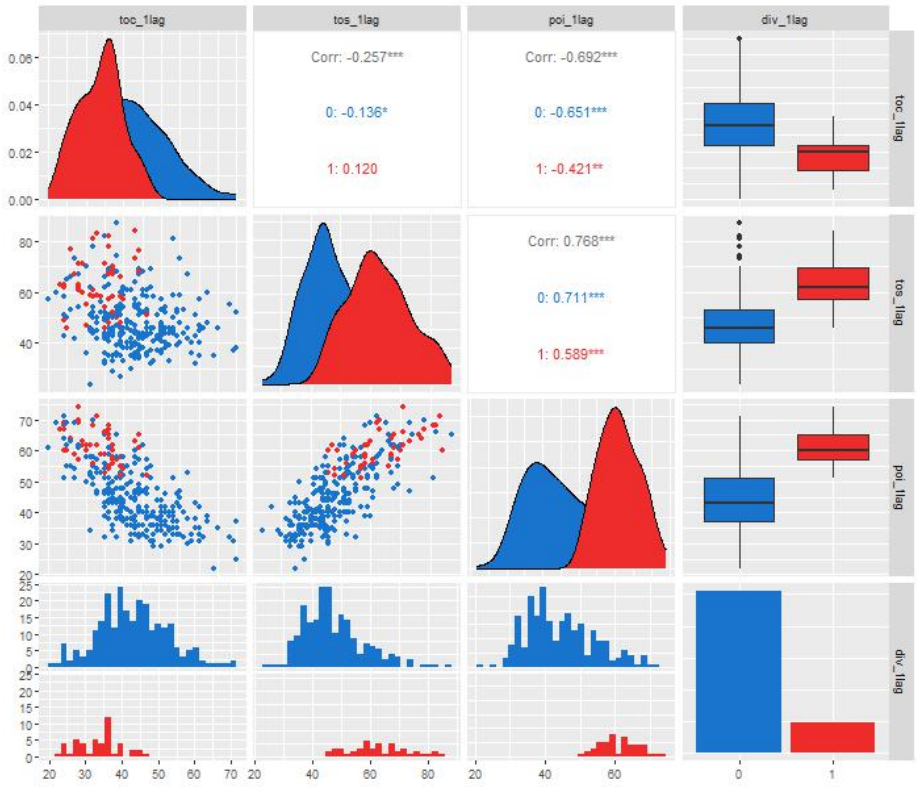


Figure 3.2.2: Pairs plot of previous years scored goals, conceded goals, points (promoted teams in red).

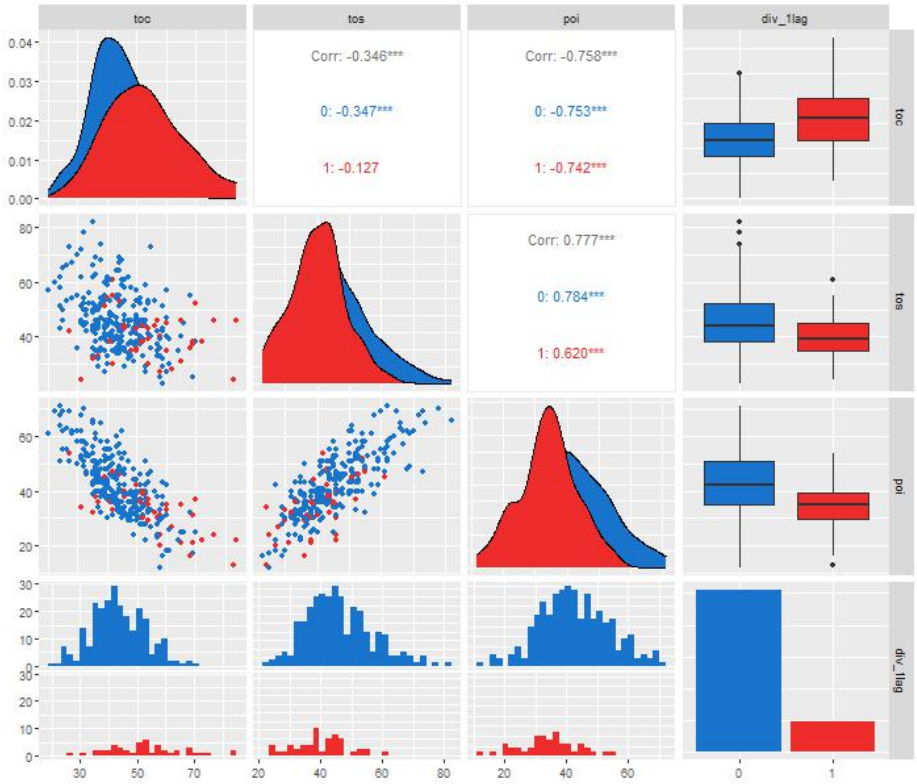


Figure 3.2.3: Pairs plot of the scored goals, conceded goals, and points at the end of the season (promoted teams in red).

3.2.1 Promoted Teams

On the previous page, there are two pairs plots of the most central quantitative predictors. Also included were the categorical predictors separating the newly promoted teams from the teams that played in Eliteserien last year. These plots underline the big difference between such teams, both their differences coming into the season and their performance in the current season.

Promoted team's performance in the previous season When a team promotes to Eliteserien, they have delivered a season with many goals, points, etc. in the lower division. Looking at Figure 3.2.2, on average, the promoted teams perform better with a much lower variance. This suggests that we must have some way to separate them; otherwise, we would assume that the newly promoted teams were going to archive the same results as the top Eliteserien teams.

Promoted team's performance in the current season It is not radical to say that promoted teams, on average, perform worse than the other teams. Figure 3.2.3 shows that they generally score fewer goals. However, their conceded goals vary considerably, suggesting that limiting the number of conceded goals might be crucial to avoiding relegation.

3.2.2 Levels of Categorical Variables

To obtain an overview of the categorical variables in the dataset, we plot the percentages of their 0/1 levels in Figure 3.2.4.

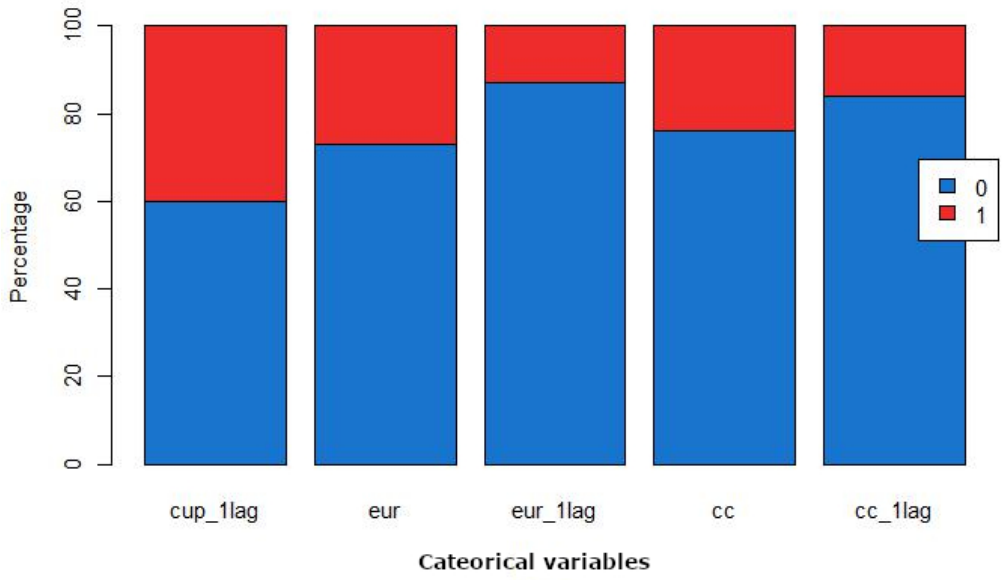


Figure 3.2.4: Bar plot showing the distribution of the categorical variables.

Looking at this plot, the categorical variables are mostly 0. For the European parameters, the distribution is as expected. With around four teams (25%) playing

qualifiers to Europe, around two teams (12.5%) actually qualify. The cup parameter could also be justified with eight quarter-final slots for the (now) 16 teams; the theoretical maximum should be 50%. Assuming that there are some early round clash-ups and "cupbomber" in the earlier rounds, 40% seems reasonable.

Looking at the coach changes, these numbers are quite interesting, as 16% of the teams changed their coaches during the last season. This equates to 2.5 clubs having a coach change during each Eliteserien season. With some seasons feeling like a kamikaze of coach sackings, this might feel like a low number, but this varies significantly from season to season. Looking at the 2018 season in Table 3.2.1, five clubs changed their coach during that season.

Table 3.2.1: Changes of coaches during Eliteserien 2018 [21]. (*) Temporary

Club	**Outgoing Head Coach**	**Reason for Departure**	**Date of Departure**
Sandefjord	Magnus Powell	Lack of results	25. April
Start	Mark Dempsey	Lack of results	18. May
Sandefjord	Geir Ludvig Fevang*	End of temporary period	31. May
Start	Mick Priest*	End of temporary period	1. June
Strømsgodset	Tor Ole Skullerud	Resigned	6. June
Stabæk	Toni Ordinas	Lack of results	27. June
Lillestrøm	Arne Erlandsen	Lack of results	29. June
Stabæk	Jan Peder Jalland*	End of temporary period	4. July
Lillestrøm	Arild Sundgot*	End of temporary period	13. July

Comparing this to the 2022 season where no coaches were sacked, the numbers start to make sense. The feeling of many changes could also come from many clubs hiring temporary coaches, often for a short period, but in some cases, they are permanently hired. When a potential new coach is hired, the club has gone through two coach changes, but it is only registered as one club doing a coach change. By the minute of writing this paragraph, it was announced that the temporary coach of Aalesunds FK Marius Bøe got an extension until the summer. This underlines the unknown future of these temporary coaches. As you read, he may be a permanent coach, assistant, or sacked. Who knows.

Changes during the off-season Approximately one in four clubs (24%) changed their coaches during the off-season. It might seem a bit high when formulated like that, but with this being the same as saying that a club changes its coach every 4th off-season, it might seem more reasonable.

Methods

After all the formalities, we finally get to predict the results of Eliteserien.

Our method has two main objectives. First, we attempt to find two models that are easy for a normal football fan to interpret while still performing at a respectable level. For the third model, we attempt to find a model that predicts the results as accurately as possible, ignoring the interpretability of the model.

4.1 Performance Measures

Having already discussed AIC, BIC, RMSE, and MAE, it should not come as a surprise for the reader that we are going to use them as performance measures in our model selection. For each model selection, we plot all four measures to obtain an overview of the performance of different model sizes. How we obtain the best models of each size, and ultimately how we select the final model, will vary based on different approaches.

1 year-fold validation A variant of k-fold cross-validation was used to calculate the RMSE and MAE. Using our model, cross-validation is used, leaving one year out at that time. For the remaining years, the model is trained and the errors are computed using the left-out year. This step is repeated for all years. In the end, the errors are averaged like in the normal k-fold cross-validation. With our model trying to predict a year given other years, this is believed to be a solid way to validate the error.

4.2 The Simple Approach

4.2.1 Response and Regression model

For the simple model, we have to ask the question, "Which variables does the normal football fan find most intuitive?". The answer to this question is easy; they mostly care about placement and points. To prove this concept, the author asked his die-hard football friend to state the placement and points of his favorite team. A question that he could answer on the spot. However, when asked about the goals scored and conceded, he could not give a precise answer. Combining this with the fact that placement is a function of points, we chose points as the response variable.

The next step is to choose the type of regression model to use. We would like to specify that there is no evidence that the points follow a Poisson distribution. From Figure 3.2.3 and the central limit theorem, it is reasonable to assume that the points follow an approximately normal distribution. Therefore, it seems logical to use an MLR model. Since it is one of the easiest models to interpret.

4.2.2 Model Selection

Having chosen to use an MLR with points as a response, a best subset selection using exhaustive search be performed. With our four performance measures, models that are easy to interpret and have respectable performance can be chosen. As a rule of thumb, we should look for a model that can easily be calculated in our heads, given the predictors, and another model that would require pen and paper. Because we are dealing with MLR, the AIC and C_p are proportional. It is easier to use C_p over the AIC, with `lm()` using it as its default.

4.2.3 Model Evaluation

To evaluate the model, R^2 provides a good clue on how well the model fits our data. By combining this with MAE and RMSE, we obtain an idea of the model performance. To check if the assumption about the residuals being normally distributed, we can use both the residual and Q-Q plots.

4.3 The Goal Approach

4.3.1 Response and Regression Model

Time to keep the tongue straight in the mouth. For the next approach, three regression models will be combined to hopefully obtain a better result than the simple models. First, the total number of scored goals is predicted, followed by the total number of conceded goals. Finally, the predicted values of these two models are used to predict the total number of points through MLR.

Scored and conceded goals As discussed in the theory, there is solid reason to expect goals to follow a Poisson distribution. Furthermore, for larger λ's, the distribution can be estimated using a normal distribution. Therefore, it can be argued that the linear model is sufficient. However, by choosing Poisson regression, the models should theoretically perform better. It is worth noting that, because it has a log-link function, it is difficult to interpret. However, when passing these predicted values into another model, the interpretability would be lost regardless.

Points from goals Next, we must find a way to predict points from our already predicted scored and conceded goals. In Figure 3.2.3 it is seen that there is some linearity between goals and points. Thus, a simple MLR

$$\hat{y}_{poi} = \hat{\beta}_{tos} x_{tos} + \hat{\beta}_{toc} x_{toc}$$

should capture the observed linearity.

4.3.2 Model Selection

When doing model selection for MLR, subset selection with exhaustive search was used. In many cases, this is impossible with the GLM because the fitting of these models is more complex. Therefore, the `bestglm()` function only allows $p = 14$ predictors when performing subset selection with exhaustive search. Unfortunately, that is one less than our 15 predictors. To get around this, a predictor is removed. Having already fitted a linear model earlier, one of the predictors that did not appear as important in that model can be removed. With the subset selection in place, the 4 performance measures can be used to select the best-suited model.

4.3.3 Model Evaluation

For the two GLMs fitted in the approach, the MAE and RMSE can be used to obtain an idea of how the model performs. Even though GLM does not have good measures for evaluating the fit, such as R^2, we can use the Pearson χ^2 goodness-of-fit test to obtain some idea of the fit. To evaluate our residuals, we use a Pearson residual plot and a Q-Q plot against the deviance residuals. For the last MLR used to determine the points, we evaluated the model as in Section 4.2.3.

Analysis and Results

Time to face the brutal reality of the predictions.

5.1 The Simple Approach

5.1.1 Model Selection

With the two easily interpretable models in mind, a variable inclusion plot from the `regsubsets()` method in the `leaps` package may be useful. Again refereeing to Appendix A for the variable explanation.

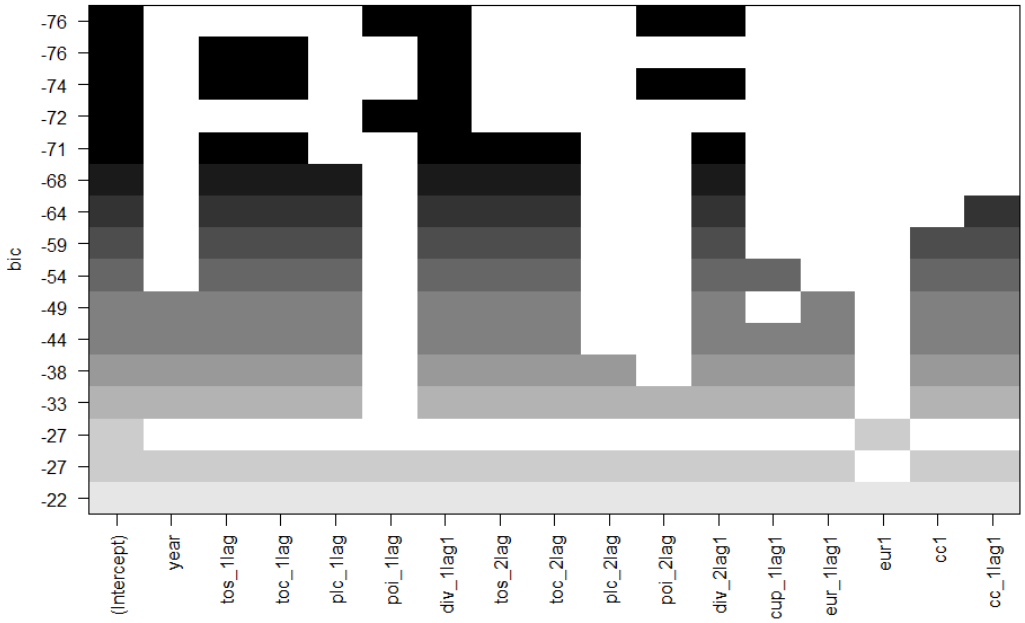


Figure 5.1.1: Variable inclusion plot. Each row represents the variables included in the best model of each model size.

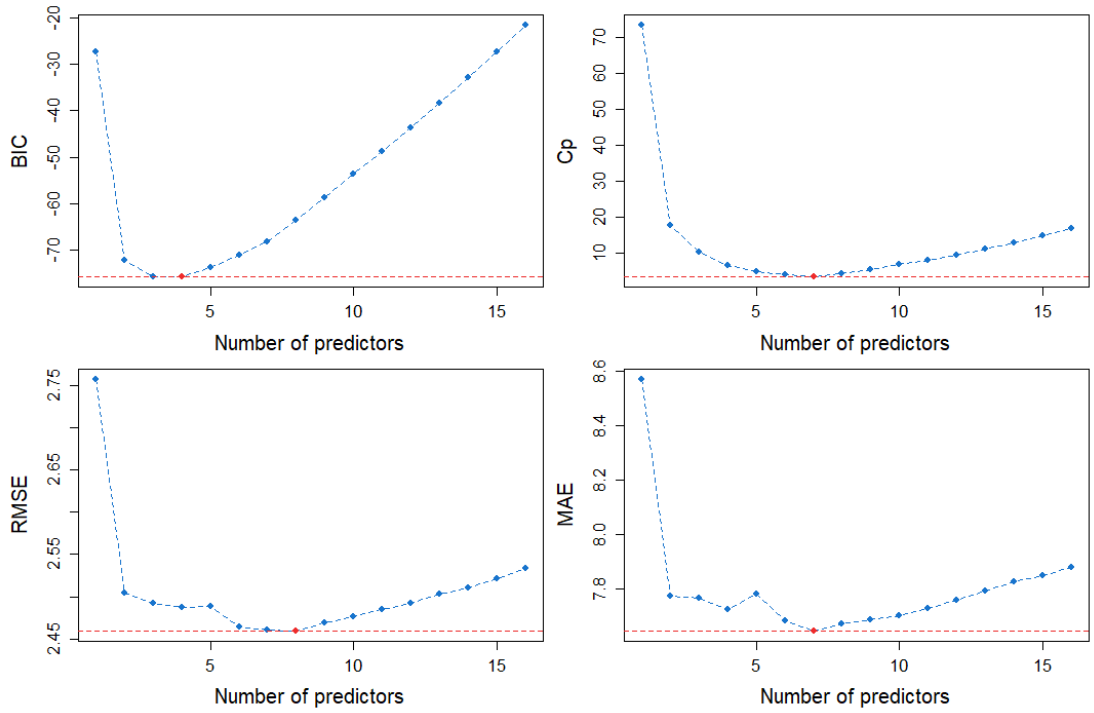


Figure 5.1.2: Error plots for the points model.

A Simple model Looking at the plotted errors in Figure 5.1.2, all measures except BIC suggest that we choose a model with 7-8 predictors. If this was a hunt for the best predicting model, it is easy to argue that a model involving 7 predictors would be sufficient. Keeping in mind that this should be an easy-to-interpret model, it should not have more than five variables. Setting a cap of 5 predictors, both our test metrics and BIC suggests that 4 predictors are the most optimal. Reading the row with 4 predictors in the variable inclusion plot in Figure 5.1.1, we obtain the following model.

$$\begin{aligned}\hat{y}_{poi} = \hat{\beta}_{intecept} + \hat{\beta}_{poi\_1lag}x_{poi\_1lag} + \hat{\beta}_{div\_1lag}x_{div\_1lag} \\ +\hat{\beta}_{poi\_2lag}x_{poi\_2lag} + \hat{\beta}_{div\_2lag}x_{div\_2lag}.\end{aligned} \tag{5.1}$$

It looks quite hard to compute with pen and paper, but with both division lags being categorical predictors, they are a constant in our model. Therefore, we only have two multiplications. This becomes clearer when we fit the model. Overall it is not too complicated to compute by hand, and we do not loose too much prediction accuracy.

A Simpler model Having found a model that can be executed on pen and paper, we still want to find a model that can be computed in our head. Looking back at our plots, we can look at the model with 3 predictors

$$\hat{y}_{poi} = \hat{\beta}_{intecept} + \hat{\beta}_{tos\_1lag}x_{tos\_1lag} + \hat{\beta}_{toc\_1lag}x_{toc\_1lag} + \hat{\beta}_{div\_1lag}x_{div\_1lag}.$$

As stated earlier, a normal football fan does not remember the scored and conceded goals. Combining this with the model would be a bit difficult to compute in our heads, and it is not a great candidate for our simpler model. However, it

is interesting to see that with these few parameters accessible, the goals are more important than points and placement.

Decreasing the number of predictors down to 2, we get the model

$$\hat{y}_{poi} = \hat{\beta}_{intecept} + \hat{\beta}_{poi\_1lag} x_{poi\_1lag} + \hat{\beta}_{div\_1lag} x_{div\_1lag}. \tag{5.2}$$

In short, this model states that a team takes a percentage of its points from the previous year and obtains a constant number of points added depending on which division they played in the previous year. With this involving one multiplication and the addition of a constant, we can finally argue that this model is easy to compute in our head.

For our observant readers, you might have seen that our first model was just a two-year extension of the last model. Thus our simple models can be divided into a 1-year model and a 2-year model.

Table 5.1.1: Simple models

Model	Predictors
Simple 1 year model	`poi_1lag, div_1lag`
Simple 2 year model	`poi_1lag, div_1lag, poi_2lag, div_2lag`

5.1.2 Model Interpretation and Evaluation

We begin by fitting our selected models in R using the `lm()` function.

```
Residuals:
     Min       1Q   Median       3Q      Max
-28.2871  -6.2534  -0.7351   7.3929  28.5849

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  19.00725    2.82482   6.729 8.77e-11 ***
poi_1lag      0.53200    0.06219   8.554 6.36e-16 ***
div_1lag1   -17.48171    1.88124  -9.293  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.871 on 298 degrees of freedom
Multiple R-squared:  0.2571,    Adjusted R-squared:  0.2522
F-statistic: 51.58 on 2 and 298 DF,  p-value: < 2.2e-16
```

Figure 5.1.3: Summary of `lm()`
Simple 1-year model.

```
Residuals:
     Min       1Q   Median       3Q      Max
-29.1115  -6.1395  -0.9393   7.2301  25.3551

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 13.27629    3.38819   3.918 0.000111 ***
poi_1lag     0.40643    0.06902   5.888 1.06e-08 ***
div_1lag1   -9.93845    2.68016  -3.708 0.000249 ***
poi_2lag     0.25700    0.06911   3.719 0.000239 ***
div_2lag1   -6.70203    2.03625  -3.291 0.001118 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.661 on 296 degrees of freedom
Multiple R-squared:  0.2933,    Adjusted R-squared:  0.2837
F-statistic: 30.71 on 4 and 296 DF,  p-value: < 2.2e-16
```

Figure 5.1.4: Summary of `lm()`
Simple 2-year model.

Starting with the simple 1 year model in Figure 5.1.3, the model states that each team receives about 53% of their points last season, plus 19 points if you played in Eliteserien the previous year, or 2 points if you just promoted. Inserting our estimated coefficients into the model in Equation 5.2, we get the estimate model

$$\hat{y}_{poi} = 19 + 0.53 \cdot x_{poi\_1lag} - 17.50 \cdot x_{div\_1lag}. \tag{5.3}$$

All the coefficients are significant. $R^2 = 0.26$, which is far from a good model fit. From the model selection, we obtain $MAE = 7.78$ and $RMSE = 2.50$. Reading from the MAE, it missed an average of 7.78 points when predicting the points of each team. This is a long way away from being able to predict anything with great confidence.

Moving to the 2-year model in Figure 5.1.4, it is time to keep our tongues straight in our mouths. This model states that a team receives about 40% of their last year's points, plus 25% of their points two years ago. You also got 13 points added if you played in Eliteserien in the last two seasons, with a deduction of 10 and 7 points if you played a level down 1 or 2 years back, respectively. Inserting our fitted coefficients into the model in Equation 5.1 and flooring some values for simplicity, we get the estimated model

$$\begin{aligned} \hat{y}_{poi} = 13 &+ 0.40 \cdot x_{poi\_1lag} + 10 \cdot x_{div\_1lag} \\ &+ 0.25 \cdot x_{poi\_2lag} - 7 \cdot x_{div\_2lag}. \end{aligned} \tag{5.4}$$

Again, we have significant coefficients and an increase of $R^2 = 0.29$. Despite this increase, R^2 is far from being good. With our $MAE = 7.72$ and $RMSE = 2.49$, we have some decrease, but it is a long way from being something to cheer about.

Residuals Finally, we checked our residuals to determine whether our model assumptions were met. As discussed earlier, this can be achieved through a residual and Q-Q plot. We obtain the plots through the `plot()` function in R.

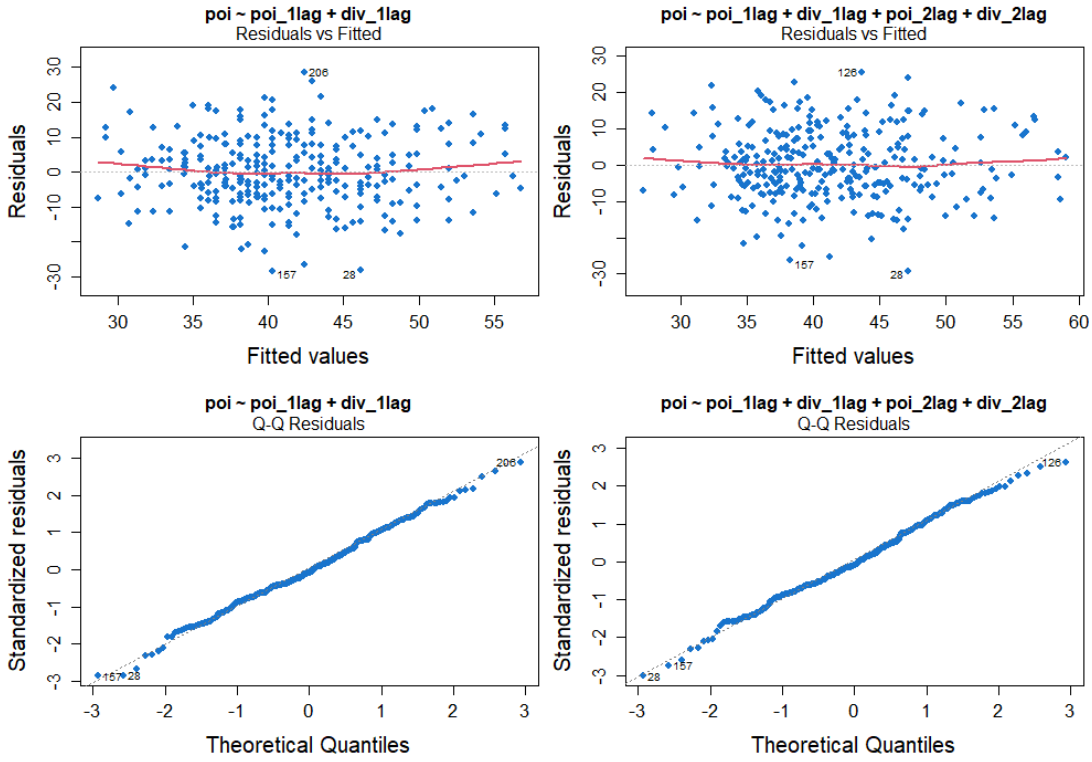


Figure 5.1.5: Residual and Q-Q plot for both models.

Looking at the residual plots in Figure 5.1.5, there is no reason to claim a severe lack of fit. Our Q-Q plots also support the models, with most points falling on the line $y = x$, indicating that the residuals are approximately normally distributed. Appendix refnotableprovides a full list of outliers found during our analysis is found in Appendix B.

5.2 The Poisson Approach

Section 4.3.1 argues that there is a reason to believe that goals follow a Poisson distribution. Thus a Poisson regression for both models involving scoring seems reasonable.

5.2.1 Model Selection

Total Scored Model We start with our model for total scored goals and use the `bestglm()` with `family = poisson(link="log")` in the `bestglm` package.

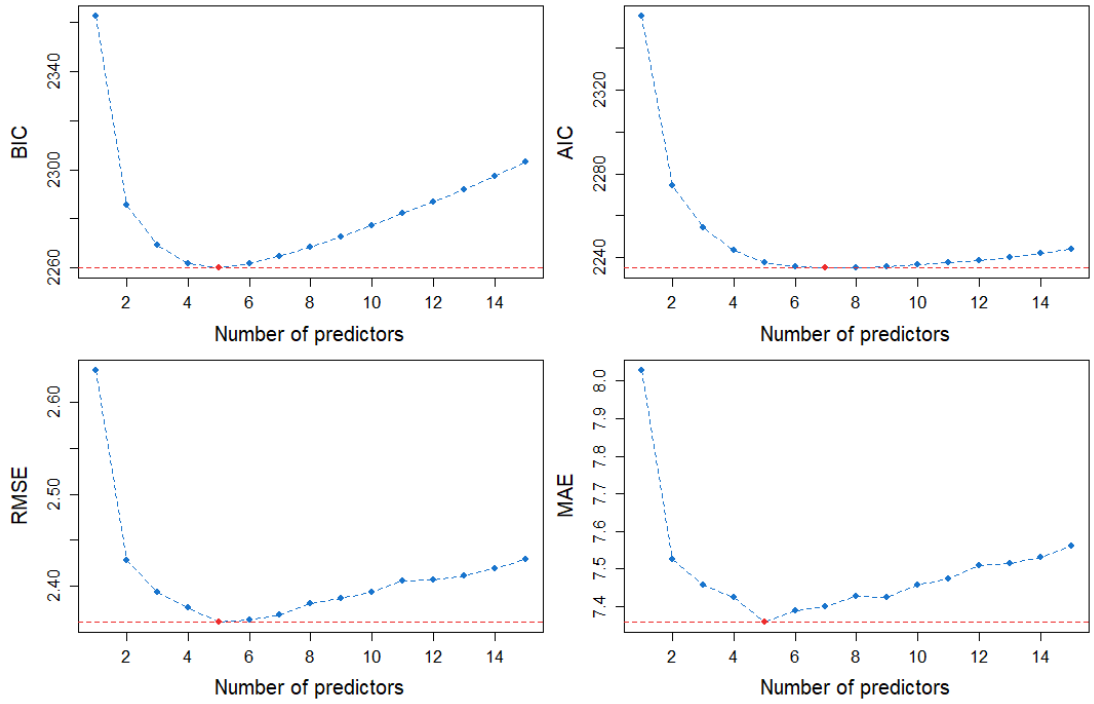


Figure 5.2.1: Error plots for the total scored goals.

Looking at the plotted errors in Figure 5.2.1 and ignoring any attempt to interpret the model, we claim that we should choose the model with 5 predictors. Both RMSE and MAE support this, and are also supported by the conservative BIC.

Total Conceded Model For conceded goals, the same process was repeated.

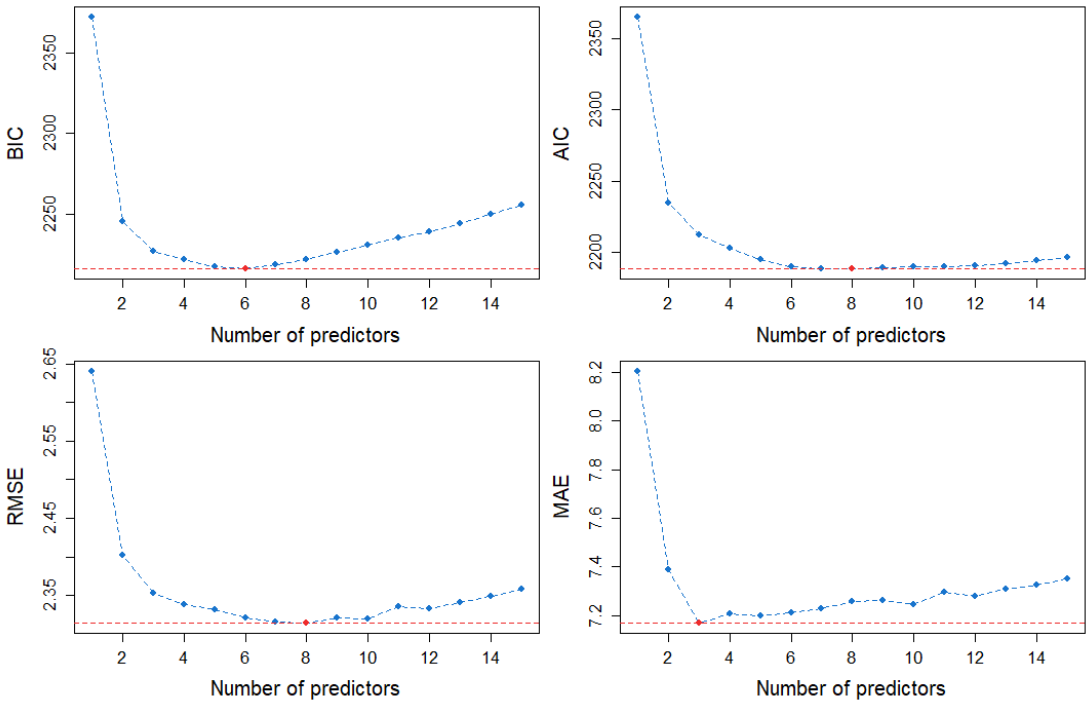


Figure 5.2.2: Error plots for the total conceded goals.

Looking at the plots in Figure 5.2.2, the different measures do not seem to agree with the number of predictors to be included. Looking at the MAE and RMSE, one suggests 3 and the other 8 predictors. Therefore, we are left with a decision on which option to prioritize. Do we want a model which on average predicts slightly better values, or do we want a model which heavier penalizes many predictions where we miss by a greater margin? With us ultimately wanting to predict the final table of a season, we argue that a model where we do not miss by bigger margins is more optimal. Further evaluation of the RMSE and the AIC, show that we could reduce the number of predictors down to 6 without losing much test error. With BIC also supporting 6 predictors, we ultimately argue to select the model with 6 predictors.

5.2.2 Model Interpretation and Evaluation

Total scored and conceded goals We fit our selected GLM models in R using the `GLM()` function.

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.2742190  0.0536221  61.061  < 2e-16 ***
tos_1lag     0.0040637  0.0011499   3.534  0.00041 ***
poi_1lag     0.0051406  0.0012883   3.990 6.60e-05 ***
div_1lag1   -0.2472313  0.0317889  -7.777 7.41e-15 ***
tos_2lag     0.0021682  0.0007757   2.795  0.00519 **
eur_1lag1    0.0803274  0.0262865   3.056  0.00224 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 726.97  on 300  degrees of freedom
Residual deviance: 534.11  on 295  degrees of freedom
AIC: 2237.6

Number of Fisher Scoring iterations: 4
```

Figure 5.2.3: Summary for `GLM` for goals scored.

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) 12.135062   3.231892   3.755 0.000173 ***
year        -0.004412   0.001601  -2.756 0.005853 **
toc_1lag     0.004791   0.001303   3.677 0.000236 ***
plc_1lag     0.016378   0.003308   4.951 7.38e-07 ***
div_1lag1    0.266172   0.033890   7.854 4.03e-15 ***
toc_2lag     0.003871   0.001055   3.670 0.000243 ***
div_2lag1    0.099331   0.027156   3.658 0.000254 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 744.88  on 300  degrees of freedom
Residual deviance: 484.84  on 294  degrees of freedom
AIC: 2189.6

Number of Fisher Scoring iterations: 4
```

Figure 5.2.4: Summary for `GLM` for goals conceded.

Looking at the total-scored model in Figure 5.2.3, all our coefficients are significant. The Pearson χ^2 to can be used to evaluate the goodness-of-fit. With the help of equation 2.8

$$\chi^2 = \sum_{i=1}^{n} r_i^2 = 532.9 \geq \chi^2_{0.95,296} = 337.1.$$

Therefore, we reject H_0, indicating that the model does not sufficiently fit the data.

Not giving up on our analysis, we repeat the process for the conceded goals. Looking at Figure 5.2.4, we again find that our coefficients are significant. Checking the fit, we again brutally reject H_0 with

$$\chi^2 = \sum_{i=1}^{n} r_i^2 = 477.9 \geq \chi^2_{0.95,295} = 336.1.$$

Continuing with the residuals using the `plot()` function in R.

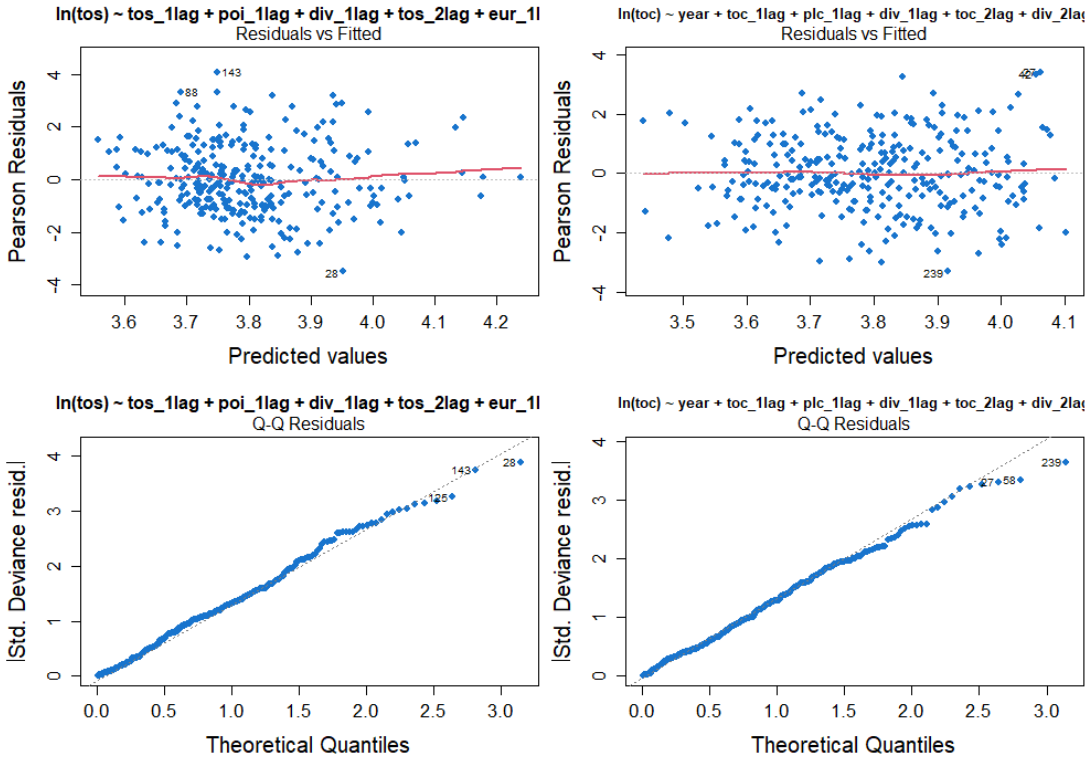


Figure 5.2.5: Residual and Q-Q plot for both scored and conceded goal GLM.

Looking at both the Pearson residual plot and the Q-Q plot with standardized deviance residuals, there is no reason to claim a severe lack of fit. Some might argue that the large theoretical quantiles fall slightly short of the line $y = x$, but not dramatically much.

Points from goals Continuing with the MLR to convert scored and conceded goals into points. Once again, we use the `plot()` function.

```
Residuals:
     Min       1Q   Median       3Q      Max
-11.9473  -2.8334   0.1399   2.6753  13.3736

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 39.62298    1.73860   22.79   <2e-16 ***
tos          0.63269    0.02347   26.96   <2e-16 ***
toc         -0.59733    0.02334  -25.60   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.027 on 298 degrees of freedom
Multiple R-squared:  0.8764,    Adjusted R-squared:  0.8755
F-statistic:  1056 on 2 and 298 DF,  p-value: < 2.2e-16
```

Figure 5.2.6: Summary for goals to points model.

For the first time, we have a model with solid $R^2 = 0.88$. We also found significant coefficients. Looking at the MAE = 3.2 and RMSE = 1.03, the MAE tells us that if we have a perfect prediction for teams scored and conceded goals, we on average still will miss by 3.2 points in our prediction.

The residual and Q-Q plots for our MLR are shown in Figure 5.2.7. Again there is no reason to claim a lack of fit.

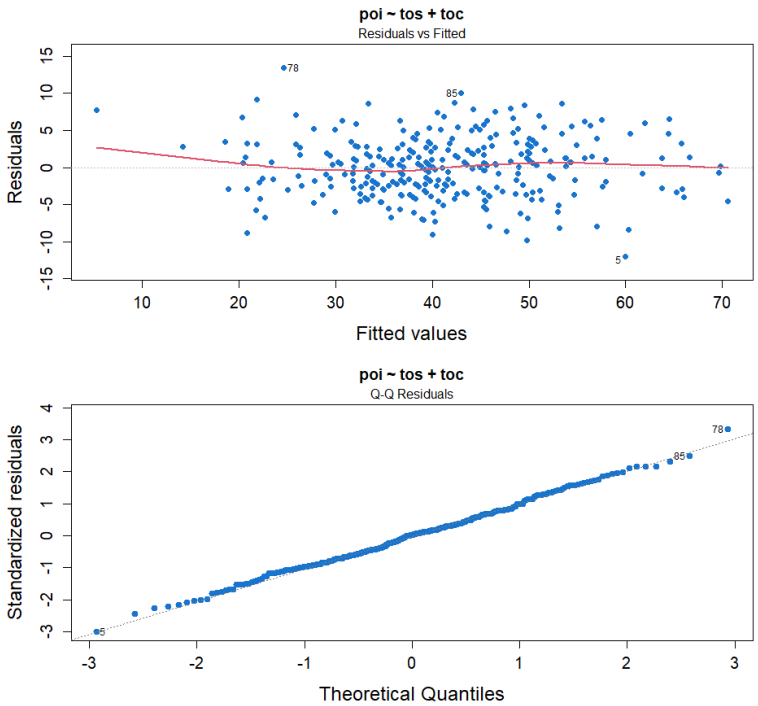


Figure 5.2.7: Residual and Q-Q plot for goals to points model.

Putting the model together To get an overview, we put the model together

$$\begin{aligned}\ln(\hat{\mu}_{tos}) = \hat{\beta}_{intecept} + \hat{\beta}_{tos\_1lag}x_{tos\_1lag} + \hat{\beta}_{poi\_1lag}x_{poi\_1lag} \\ +\hat{\beta}_{div\_1lag}x_{div\_1lag} + \hat{\beta}_{tos\_2lag}x_{tos\_2lag} + \hat{\beta}_{eur\_1lag}x_{eur\_1lag}.\end{aligned}$$

$$\begin{aligned}\ln(\hat{\mu}_{toc}) = \hat{\beta}_{intecept} + \hat{\beta}_{tos\_1lag}x_{tos\_1lag} + \hat{\beta}_{plc\_1lag}x_{plc\_1lag} \\ +\hat{\beta}_{div\_1lag}x_{div\_1lag} + \hat{\beta}_{toc\_2lag}x_{toc\_2lag} + \hat{\beta}_{div\_2lag}x_{div\_2lag}.\end{aligned}$$

$$\hat{y}_{poi} = \hat{\beta}_{tos}\hat{\mu}_{tos} + \hat{\beta}_{toc}\hat{\mu}_{toc}.$$

The first two models can be used to predict the total scored goals $\hat{\mu}_{tos}$ and conceded goals $\hat{\mu}_{toc}$. The final points prediction $\hat{y}_{poi}$ is obtained by passing these into the last model. Errors are obtained with cross-validation, ending up with MAE = 7.60 and RMSE = 2.45.

5.3 Models Overview

From Table 5.3.1, two points become clear. The predictions are far from good, and we do not gain much model precision for the more complex models. Looking at the difference in MAE, we reduce the average error by 0.12, which in the context of points is nothing. However, it is clear that our Poisson model performs better than the linear models, indicating that the Poisson model is theoretically better for our problem.

Table 5.3.1: Error of all models

Model	MAE	RMSE
Simple 1 year model	7.78	2.50
Simple 2 year model	7.72	2.49
Poisson model	7.60	2.45

5.4 Eliteserien 2023 Predictions

To complete this thesis, we are going to do something doomed to go wrong. The ridiculous task of trying to predict Eliteserien. With the odds being stacked against us, we are setting this up for people in the near future to make fun of us.

Although our models have a marginal difference in error, this does not mean that they will predict the same final standing. To illustrate this, we can predict the 2023 Eliteserien using all 3 models and compare their differences. We predict the final standing by simply predicting the points of all teams and sorting them in descending order. Ending up with the predictions on the next page.

1	Molde
2	Bodø/Glimt
3	Rosenborg
4	Lillestrøm
5	Brann
6	Vålerenga
7	Tromsø
8	Odd
9	Sarpsborg 08
10	Aalesund
11	Haugesund
12	Viking
13	Strømsgodset
14	HamKam
15	Stabæk
16	Sandefjord

1	Molde
2	Bodø/Glimt
3	Rosenborg
4	Lillestrøm
5	Brann
6	Vålerenga
7	Viking
8	Odd
9	Sarpsborg 08
10	Tromsø
11	Haugesund
12	Aalesund
13	Hamkam
14	Strømsgodset
15	Stabæk
16	Sandefjord

1	Molde
2	Bodø/Glimt
3	Rosenborg
4	Lillestrøm
5	Brann
6	Vålerenga
7	Odd
8	Sarpsborg 08
9	Tromsø
10	Haugesund
11	Viking
12	Aalesund
13	Hamkam
14	Strømsgodset
15	Stabæk
16	Sandefjord

Figure 5.4.1: Eliteserien 2023 predicted by the simple 1-year model, the simple 2-year model, and the Poisson model respectively.

The first thing worth noting is the fact that all models seem to agree on the top 5 and the relegated teams With Molde, Bodø/Glimt, and Rosenborg taking home the medals, and Stabæk and Sandefjord being relegated. With the odds of a team either overperforming or underperforming, at least one of these predictions or going to be laughed at the end of the season. For the mid-table teams, the models can not quite agree. For teams, such as Viking and Tromsø, we see that their predictions vary in some places.

For the simple models, this may be related to their varying performance over the last two seasons; hence, when looking one year back, they do not get credit for their season two years back. Having talked about the simple models, the Poisson model is believed to predict the best table. If we had to predict one table for our readers, we would choose the Poisson model as our guess. Having realized how difficult it is to predict the final table at the start of the season, we would like to lock our answer and see where this is going.

Retrospective With the thesis being written in May 2023 and having already seen a third of Eliteserien 2023, there is already some clear evidence that our predictions will be horribly wrong. With both Rosenborg and Molde having opened their season as if they were going for relegation, their way to the top 3 is going to be long. At the other end of the table, Aalesund managed to set a new record of games played without scoring, setting up for the greatest comeback of the century.

Conclusions and Remarks

Having done all this, the conclusion is quite clear. The models do not perform on a level where it is worth betting away your student scholarship. With the best model missing an average of 7.6 points for each team, there is a margin of 15.2 points that a team can place while still being in the predicted range. With this in some cases being the difference between battling relegation to battling for a medal, it is fair to say that the model's prediction error is too big.

If one wants to use the models to predict future results, the recommendation would be to use the simple 1-year model. With it only missing by 0.18 points more than the Poisson model, while only having two parameters. This model is relatively good and easy to use.

Why do we not see better results? There are many factors that affect the results of this prediction. One can argue that a larger dataset with more predictors is required. Things such as players sold, players bought, and team budget are predictors that could have improved the results of the analysis. We could also argue that we should have tried using other methods; however, this is speculative.

We could also claim that predicting Eliteserien is even more complicated than that of larger leagues because teams are more unstable. As in most leagues, there are the risks of players and coaches leaving if a team delivers a relatively bad season. However, in Eliteserien, there is also the risk of the opposite affecting a team in the same way. If a team delivers a solid season, there is a risk of both players and coaches leaving for bigger clubs, hence affecting the teams in the same way as if they played badly.

Lastly, football is and will always be unpredictable, with too many uncontrollable factors. This is the reason why the author and all the football experts will be bullied for their table predictions at the end of each season.

Coaches As stated multiple times throughout the thesis, the dataset on coaches was the most tedious. With our analysis not finding these changes important, the author wasted much of his time, and there is no reason to believe that coach changes affect a team. More research into this should be conducted, and it is left as an exercise for the reader.

References

[1] Norges Fotballforbund. *Finn kamper*. 2023. URL: https://www.fotball.no/turneringer/ (visited on 04/02/2023).

[2] Lars Aarhus and RSSSF Norway. *RSSSF Norwegian Football Archive*. 2023. URL: https://www.rsssf.no/archive.html (visited on 04/02/2023).

[3] *den som ler sist ler best oversetter engelsk*. 2023-05-16. URL: https://www.google.com/search?q=den+som+ler+sist+ler+best+oversetter+engelsk.

[4] UEFA. *UEFA.com*. 2023. URL: https://www.uefa.com/ (visited on 04/02/2023).

[5] Axel Springer SE. *transfermarkt.com*. 2023. URL: https://www.transfermarkt.com/ (visited on 04/02/2023).

[6] Øyvind Salvesen. "Statistical Models in Ice Hockey". PhD thesis. Trondheim: Norwegian University of Science and Technology, Nov. 2011.

[7] Quang Nguyen. *Predictive Models of English Premier League Goal Scoring*. 2020. URL: https://bookdown.org/theqdata/honors_thesis/ (visited on 04/15/2023).

[8] *Linear regression*. [Online; accessed 2023-05-24]. URL: https://en.wikipedia.org/wiki/Linear_regression.

[9] Douglas C. Montgomery and George G. Runger. *Applied Statistics and Probability for Engineers*. Arizona State University: Wiley, 2023.

[10] Ronald E. Walpole, Raymond H. Myers, and Sharon L. Myers. *Probability and Statistics for Engineers and Scientists*. New Jersey: Prentice hall international, INC, 1998.

[11] Marta García Ben and Víctor J Yohai. "Quantile–Quantile Plot for Deviance Residuals in the Generalized Linear Mode". In: *Journal of Computational and Graphical Statistics*. Taylor & Francis, Ltd., 2004, pp. 36–47.

[12] R.H Myers, D.C. Montgomery, and G.G. Vining. *Generalized Linear Models – With Applications in Engineering and the Sciences*. New York: Wiley, 2002, p. 131.149.

[13] Wikipedia. *Poisson regression*. [Online; accessed 2023-05-19]. URL: https://en.wikipedia.org/wiki/Poisson_regression.

[14] P. McCullagh. *Generalized Linear Models*. Routledge, 1989.

[15] David E. Bock, Paul F. Velleman, and Richard D. De Veaux. *Stats, Modeling the World*. Boston: Pearson Addison Wesley, 2007.

[16] *Akaike information criterion*. [Online; accessed 2023-05-19]. URL: https://en.wikipedia.org/wiki/Akaike_information_criterion.

[17] *Bayesian information criterion*. [Online; accessed 2023-05-19]. URL: https://en.wikipedia.org/wiki/Bayesian_information_criterion.

[18] Thomas Lumley based on Fortran code by Alan Miller. *Regression Subset Selection*. [Online; accessed 2023-05-22]. 2020. URL: https://cran.r-project.org/web/packages/leaps/leaps.pdf.

[19] A.I. McLeod, Changjiang Xu, and Yuanhao Lai. *Best Subset GLM and Regression Utilities*. [Online; accessed 2023-05-22]. 2020. URL: https://cran.r-project.org/web/packages/bestglm/bestglm.pdf.

[20] Wikipedia contributors. *SQL*. [Online; accessed 2023-04-02]. 2023. URL: https://en.wikipedia.org/wiki/SQL.

[21] *Eliteserien i fotball for menn 2018*. [Online; accessed 2023-05-24]. URL: https://no.wikipedia.org/wiki/Eliteserien_i_fotball_for_menn_2018.

Appendix

A Variable Explanation

- **year**: the year of the season the data comes from.
- **tos**: the total amount of goals scored that season.
- **toc**: the total amount of goals conceded that season.
- **plc**: the placement that season.
- **poi**: the total amount of points taken that season.
- **tos_1lag**: the total amount of goals scored the previous season.
- **toc_1lag**: the total amount of goals conceded the previous season.
- **plc_1lag**: the placement the previous season.
- **poi_1lag**: the total amount of points taken the previous season.
- **div_1lag**: If the team played in the lower division last season 1, else 0.
- **tos_2lag**: the total amount of goals scored the second to last season.
- **toc_2lag**: the total amount of goals conceded second to last season ago.
- **plc_2lag**: the placement the two seasons ago.
- **poi_2lag**: the total amount of points taken the second to last season.
- **div_2lag**: If the team played in the lower division second to last season 1, else 0.
- **cup_1lag** : If the team is at least came through to the quarter-final in the domestic last season 1, else 0.
- **eur_1lag**: If the team qualified for the group stage in a European competition 1, else 0.
- **eur**: If a team is playing qualification for a European competition during the season 1, else 0.
- **cc**: If a team got a new coach in pre-season 1, else 0.
- **cc_1lag**: If a team changed coach during the last season 1, else 0

B Noteable Obersvations

- **5**: Stabæk 2000 (5.)
- **15**: Rosenborg 2001 (1.)
- **16**: Lillestrøm 2001 (2.)
- **27**: Strømsgodset 2001(13.)
- **28**: Tromsø 2001 (14.)
- **42**: Start 2002 (14.)
- **78**: Odd 2005 (9.)
- **85**: Brann 2006 (2.)
- **88**: Stabæk 2006 (5.)
- **125**: HamKam 2008 (14.)
- **126**: Rosenborg 2009 (1.)
- **143**: Vålerenga 2010 (2.)
- **157**: Sandefjord 2010 (16.)
- **206**: Molde 2014 (1.)

C Proofs

It is rather intriguing to devote an entire section to the elucidation of a singular, albeit significant, proof.

C.1 Proof Least-squares Estimation

Proof. Starting we have

$$\boldsymbol{\epsilon}'\boldsymbol{\epsilon} = (\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) = \mathbf{y}'\mathbf{y}-\boldsymbol{\beta}'\mathbf{X}'\mathbf{y}-\mathbf{y}'\mathbf{X}\boldsymbol{\beta}+\boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta} = \mathbf{y}'\mathbf{y}-2\boldsymbol{\beta}'\mathbf{X}'\mathbf{y}+\boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta}.$$

The last step follows from

$$\mathbf{y}'\mathbf{X}\boldsymbol{\beta} = (\boldsymbol{\beta}'\mathbf{X}'\mathbf{y})' = \boldsymbol{\beta}'\mathbf{X}'\mathbf{y}$$

since $\boldsymbol{\beta}'\mathbf{X}'\mathbf{y}$ is a scalar. $\boldsymbol{\epsilon}'\boldsymbol{\epsilon}$ is minimized where

$$\frac{\partial}{\partial\boldsymbol{\beta}}\boldsymbol{\epsilon}'\boldsymbol{\epsilon} = -2\mathbf{X}'\mathbf{y} + 2\mathbf{X}'\mathbf{X}\boldsymbol{\beta} = 0.$$

Thus

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}.$$

□

D R code

D.1 Data Frame Setup

```
library(bestglm)
col <- c("dodgerblue3","firebrick2")
# set colors, the most important thing
source("methods.R") #import helper methods
df = setup_data("res.json") #load data
#define dataframes with different responses
df_no_pred = subset(df,select=-c(poi,tos,toc,plc))
poi = df$poi
df_poi = cbind(df_no_pred,poi)
tos = df$tos
df_tos = cbind(df_no_pred,tos)
toc = df$toc
df_toc = cbind(df_no_pred,toc)
```

D.2 R Code for Section 5.1

```
par(mfrow=c(1,1))
regsubsets_poi = regsubsets(poi ~ . ,data=df_poi,nvmax = 20)
plot(regsubsets_poi)
summary_regsubsets_poi = summary(regsubsets_poi)
bic_poi = summary_regsubsets_poi$bic
cp_poi = summary_regsubsets_poi$cp
cross_rmse = cross_validation(regsubsets_poi,"RMSE")
cross_mae = cross_validation(regsubsets_poi,"MAE")
par(mfrow=c(2,2),mar = c(5,5,0,0), oma = c(0,0,1,1))
error_plot(bic_poi,"Number of predictors","BIC")
error_plot(cp_poi,"Number of predictors","Cp")
error_plot(cross_rmse$error,"Number of predictors","RMSE")
error_plot(cross_mae$error,"Number of predictors","MAE")
easy_poi1 = 2
easy_poi2 = 4
easy_model_poi1 = cross_mae$model(easy_poi1)
easy_model_poi2 = cross_mae$model(easy_poi2)
summary(easy_model_poi1)
summary(easy_model_poi2)
pred_string1 = paste(filter_variables(
names(easy_model_poi1$coefficients)), collapse = " + ")
pred_string2 <- paste(filter_variables(
names(easy_model_poi2$coefficients)), collapse = " + ")
par(mfrow=c(2,2),mar = c(4,5,4,0), oma = c(0,0,0,1))
plot(easy_model_poi1, which = c(1),cex.lab=1.5,cex.axis=1.25,main
=paste("poi~",pred_string1),lwd=2,pch=19,col=c(col[1]))
plot(easy_model_poi2, which = c(1),cex.lab=1.5,cex.axis=1.25,main
=paste("poi~",pred_string2),lwd=2,pch=19,col=col[1])
plot(easy_model_poi1, which = c(2),cex.lab=1.5,cex.axis=1.25
,main=paste("poi ~",pred_string1),pch=19,col=col[1])
plot(easy_model_poi2, which = c(2),cex.lab=1.5,cex.axis=1.25
,main=paste("poi ~",pred_string2),pch=19,col=col[1])
par(mfrow=c(1,1))
```

D.3 R Code for Section 5.2

```
#Model selection tos
df_tos_cut = subset(df_tos, select=-c(eur))
bestglm_tos = bestglm(df_tos_cut,family=poisson)
measure_tos = measure_glm(bestglm_tos,df_tos_cut)
par(mfrow=c(2,2),mar = c(5,5,0,0), oma = c(0,0,1,1))
error_plot(measure_tos$BIC(),"Number of predictors","BIC")
error_plot(measure_tos$AIC(),"Number of predictors","AIC")
error_plot(measure_tos$error_rmse,"Number of predictors","RMSE")
error_plot(measure_tos$error_mae,"Number of predictors","MAE")
par(mfrow=c(1,1))
best_tos = measure_tos$model(5)
summary(best_tos)

#Pearson residual chi test
p = length(best_tos$coefficients) -1
residuals <- residuals(best_tos, type = "pearson")
pearson_chisq <- sum(residuals^2)
pearson_chisq
deg_fre <- length(residuals) - p
deg_fre
cut_off = qchisq(0.95,deg_fre)
cut_off

#model selection toc
df_toc_cut = subset(df_toc, select=-c(eur))
bestglm_toc = bestglm(df_toc_cut,family=poisson)
measure_toc = measure_glm(bestglm_toc,df_toc_cut)
par(mfrow=c(2,2),mar = c(5,5,0,0), oma = c(0,0,1,1))
error_plot(measure_toc$BIC(),"Number of predictors","BIC")
error_plot(measure_toc$AIC(),"Number of predictors","AIC")
error_plot(measure_toc$error_rmse,"Number of predictors","RMSE")
error_plot(measure_toc$error_mae,"Number of predictors","MAE")
par(mfrow=c(1,1))
best_toc = measure_toc$model(6)
summary(best_toc)

#Pearson residual chi test
p = length(best_toc$coefficients) -1
residuals <- residuals(best_toc, type = "pearson")
pearson_chisq <- sum(residuals^2)
pearson_chisq
deg_fre <- length(residuals) - p
deg_fre
cut_off = qchisq(0.95,deg_fre)
cut_off
```

```
#Pearson residual plt and Q-Q
par(mfrow=c(2,2),mar = c(4,5,4,0), oma = c(0,0,0,1))
plot(best_tos, which = c(1),cex.lab=1.5,cex.axis=1.25,main
=paste("ln(tos) ~",pred_string1),lwd=2,pch=19,col=c(col[1]))
plot(best_toc, which = c(1),cex.lab=1.5,cex.axis=1.25,cex.main=1,main
=paste("ln(toc) ~",pred_string2),lwd=2,pch=19,col=col[1])
plot(best_tos, which = c(2),cex.lab=1.5,cex.axis=1.25,main
=paste("ln(tos) ~",pred_string1),pch=19,col=col[1])
plot(best_toc, which = c(2),cex.lab=1.5,cex.axis=1.25,cex.main=1,main=
paste("ln(toc) ~",pred_string2),pch=19,col=col[1])
par(mfrow=c(1,1))

#zen #voi #biojentene #sponsetinnlegg

#Testing fit of lm and cross validation the full poisson model
pred_tos = exp(predict(best_tos,df_poi))
pred_toc = exp(predict(best_toc,df_poi))
lm_poi_goal= lm(poi ~ tos + toc,data = df)
summary(lm_poi_goal)
par(mfrow=c(2,1))
plot(lm_poi_goal, which = c(1),cex.lab=1.5,cex.axis=1.25,main=
paste("poi ~ tos + toc"),lwd=2,pch=19,col=c(col[1]))
plot(lm_poi_goal, which = c(2),cex.lab=1.5,cex.axis=1.25,main=
paste("poi ~ tos + toc"),lwd=2,pch=19,col=c(col[1]))
par(mfrow=c(1,1))
pred_poi = predict(lm_poi_goal,data.frame(tos = pred_tos, toc = pred_toc))
mae = mean(abs(pred_poi-df$poi))
rmse = sqrt(sum((pred_poi - df$poi)^2))/length(pred_poi)
error_lm = simple_lm_cross_validation(as.formula("poi ~ tos + toc"),df)
error_lm$mae
error_lm$rmse
error_full_mod = multiple_cross_validation(df,as.formula("poi ~ tos + toc"),
as.formula("tos ~ tos_1lag + poi_1lag + div_1lag + tos_2lag + eur_1lag"),
as.formula("toc ~ toc_1lag + plc_1lag + div_1lag + toc_2lag + div_2lag"))
error_full_mod$mae
error_full_mod$rmse

####Predict the data for 2023 season
df_new = setup_data("data22.json")
pred1 = predict(easy_model_poi1,df_new)
indices1 <- order(unlist(pred1),decreasing = TRUE)
prde2 = predict(easy_model_poi2,df_new)
indices2 <- order(unlist(pred2),decreasing = TRUE)
df_new
pred31 = exp(predict(best_tos,df_new))
pred32 = exp(predict(best_toc,df_new))
pred3 = predict(lm_poi_goal,data.frame(tos = pred31, toc = pred32))
indices3 <- order(unlist(pred3),decreasing = TRUE)
```

```
teams = c("Molde","Bodo Glimt","Rosenborg",
          "Lillestrom","Odd","Vaalerenga",
          "Tromso","Sarpsborg 08","Aalesund",
          "Haugesund","Viking","Stromsgodset",
          "HamKam","Sandefjord","Brann","Stabaek")
#Some crazy shit.
cbind(1:16,teams[indices1])
cbind(1:16,teams[indices2])
cbind(1:16,teams[indices3])
```

D.4 Helper Methods

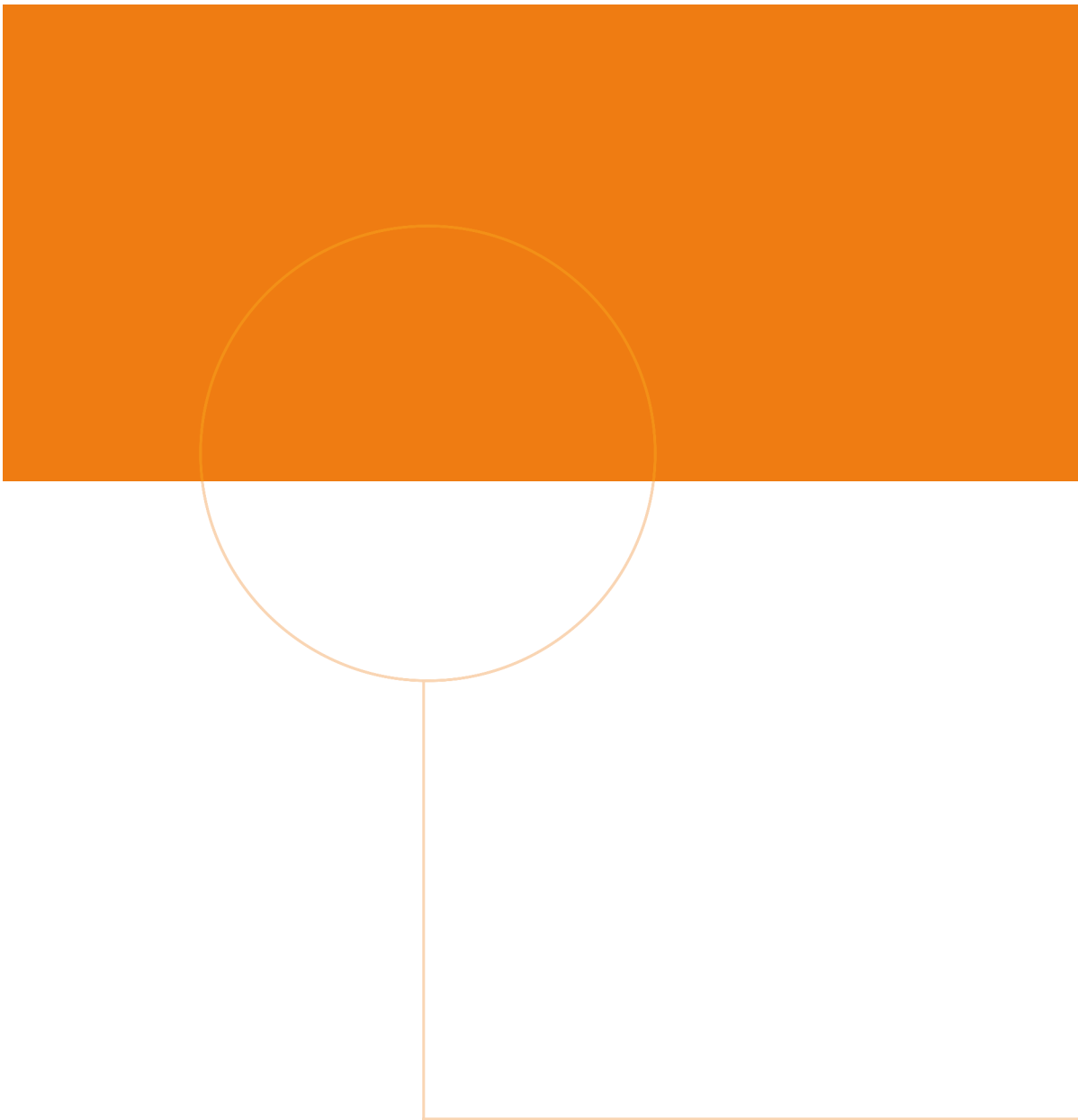
```
library("rjson")
library("ggplot2")
library(leaps)
library(dplyr)
col <- c("dodgerblue3","firebrick2")
#set colors, the most important thing

setup_data = function(file_stirng) {
  #sets up the data frame from json file
  data <- fromJSON(file= file_stirng)
  df = data.frame(data)

  df$div_1lag = as.factor(df$div_1lag)
  df$div_2lag = as.factor(df$div_2lag)
  df$cup_1lag = as.factor(df$cup_1lag)
  df$eur_1lag = as.factor(df$eur_1lag)
  df$eur= as.factor(df$eur)
  df$cc = as.factor(df$cc)
  df$cc_1lag = as.factor(df$cc_1lag)
  return(df)
}

setup_data_num = function(file_stirng {
  #sets up the data frame from json file
  data <- fromJSON(file_string)
  df = data.frame(data)
  return(df)
}

error_plot = function(list,xlab,ylab) {
  #helper function for pretty error plot
  plot(list,xlab=xlab,ylab=ylab,
    cex.lab=1.5,cex.axis=1.25,pch=19,col=col[1])
  lines(list,lty=2,col=col[1])
  points(which.min(list),min(list),pch=19,col=col[2])
  abline(min(list),0,lty=2,col=col[2])
}

filter_variables = function(var_list) {
  #to filter variable nummy numbers
  no_intercept = var_list[-1]
  pred = gsub('^(.*?)(\\d+)?$', '\\1', no_intercept)
  return(pred)
}

cross_validation = function(reg_model,meassure) {
```

```
  #cross validation for regsubsets
  response = all.vars(reg_model$call)[1]
  summary = summary(reg_model)
  nvmax = reg_model$nvmax - 1 #remove intecept
  df_string = all.vars(reg_model$call)[length(all.vars(reg_model$call))]
  df = get(df_string)
  cv_error = matrix(NA, 18, nvmax, dimnames = list(NULL, paste(1:nvmax)))
  for (j in 1:18) {
    y = 1999 + j
    data.train = dplyr::filter(df, y != year)
    data.test = dplyr::filter(df, y == year)
    for (i in 1:nvmax) {
      best_model_names = names(coef(reg_model,i))[-1]
      best_model_names = gsub('^(.*?)(\\d+)?$', '\\1', best_model_names)
      best_model_names = gsub("0", "", best_model_names)
      predictors <- paste(best_model_names, collapse = " + ")
      formula_string <- paste("poi ~", predictors)
      lm = lm(formula_string,data = data.train)
      pred = predict(lm,data.test)
      if(meassure == "RMSE") {
        error = sqrt(sum((pred - data.test[response])^2))/length(pred)
      } else if (meassure == "MAE") {
        error = sum(abs(pred - data.test[response]))/length(pred)
      }
      cv_error[j,i] = error
    }
  }
  temp = c()
  temp$error = colMeans(cv_error)
  temp$error_matrix = cv_error
  temp$best = which.min(temp$error)
  temp$model = function(index) {
    best_model_names = names(coef(reg_model,index))[-1]
    best_model_names = gsub('^(.*?)(\\d+)?$', '\\1', best_model_names)
    best_model_names = gsub("0", "", best_model_names)
    predictors <- paste(best_model_names, collapse = " + ")
    formula_string <- paste("poi ~", predictors)
    lm = lm(formula_string,data = df)
    return(lm)
  }
  return(temp)
}



measure_glm = function(reg_model,df) {
  #get all measures for glm (bestglm) (MAE,RMSE,AIC,BIC)
```

```
response = names(df)[length(df)]
nvmax = ncol(df) - 1 #remove intecept
cv_error_rmse = matrix(NA, 18, nvmax, dimnames
= list(NULL, paste(1:nvmax)))
cv_error_mae = matrix(NA, 18, nvmax, dimnames
= list(NULL, paste(1:nvmax)))
for (j in 1:18) {
  y = 1999 + j
  data.train = filter(df, y != year)
  data.test = filter(df, y == year)
  for (i in 1:nvmax) {
    row = reg_model$Subsets[i+1,]
    best_model_names = names(row)[row == TRUE][-1]
    best_model_names = gsub('^(.*?)(\\d+)?$', '\\1'
                      , best_model_names)
    best_model_names = gsub("0", "", best_model_names)
    predictors <- paste(best_model_names, collapse = " + ")
    formula_string <- paste(response," ~", predictors)
    glm = glm(formula_string,data = data.train,family="poisson")
    pred = exp(predict(glm,data.test))
    error_rmse =
    sqrt(sum((pred - data.test[response])^2))/length(pred)
    error_mae =
    sum(abs(pred - data.test[response]))/length(pred)
    cv_error_rmse[j,i] = error_rmse
    cv_error_mae[j,i] = error_mae
  }
 #Kristian Gartz
}
temp = c()
temp$error_rmse = colMeans(cv_error_rmse)
temp$error_mae = colMeans(cv_error_mae)
temp$error_matrix_rmse = cv_error_rmse
temp$error_matrix_mae = cv_error_mae
temp$best = which.min(temp$error)
temp$model = function(index) {
  row = reg_model$Subsets[index+1,]
  best_model_names = names(row)[row == TRUE][-1]
  best_model_names = gsub('^(.*?)(\\d+)?$', '\\1'
                       , best_model_names)
  best_model_names = gsub("0", "", best_model_names)
  predictors <- paste(best_model_names, collapse = " + ")
  formula_string <- paste(response," ~", predictors)
  glm = glm(formula_string,data = df,family="poisson")
  return(glm)
}
temp$AIC = function() {
  aic = c()
```

```
    for(i in 1:nvmax) {
      glm = temp$model(i)
      aic = c(aic,AIC(glm))
      #CLARA
      #ARILD
      #MARTIN
      #-----> CAM -----> Saurus
    }
    return(aic)
  }
  temp$BIC = function() {
    bic = c()
    for(i in 1:nvmax) {
      glm = temp$model(i)
      bic = c(bic,BIC(glm))
    }
    return(bic)
  }
  return(temp)
}
#zen #voi #biojentene #sponsetinnlegg
simple_lm_cross_validation = function(lm_formula,df) {
  #crossvalidation for lm()
  error_list_rmse = c()
  error_list_mae = c()
  response = all.vars(lm_formula)[1]
  for (j in 1:18) {
    y = 1999 + j
    data.train = filter(df, y != year)
    data.test = filter(df, y == year)
    lm_mod = lm(lm_formula,data = data.train)
    pred = predict(lm_mod,data.test)
    error_rmse = sqrt(sum((pred - data.test[response])^2))/length(pred)
    error_mae = sum(abs(pred - data.test[response]))/length(pred)
    error_list_mae = c(error_list_mae, error_mae)
    error_list_rmse = c(error_list_rmse, error_rmse)
  }
  return(list(mae = mean(error_list_mae), rmse = mean(error_list_rmse)))
}
multiple_cross_validation =
function(df,lm_formula,tos_formula,toc_formula) {
  #cross validation for the Poisson approach
  error_list_rmse = c()
  error_list_mae = c()
  response = all.vars(lm_formula)[1]
  for (j in 1:18) {
    y = 1999 + j
    data.train = filter(df, y != year)
```

```
    data.test = filter(df, y == year)
    glm_mod_tos = glm(tos_formula,family = poisson(),data=data.train)
    glm_mod_toc = glm(toc_formula,family = poisson(),data=data.train)
    lm_mod_poi = lm(lm_formula,data = data.train)
    pred_tos = exp(predict(glm_mod_tos,data.test))
    pred_toc = exp(predict(glm_mod_toc,data.test))
    pred_poi
    = predict(lm_mod_poi,data.frame(tos = pred_tos, toc = pred_toc))
    error_rmse
    = sqrt(sum((pred_poi - data.test[response])^2))/length(pred_poi)
    error_mae
    = sum(abs(pred_poi - data.test[response]))/length(pred_poi)
    error_list_mae = c(error_list_mae, error_mae)
    error_list_rmse = c(error_list_rmse, error_rmse)
  }
  return(list(mae = mean(error_list_mae), rmse = mean(error_list_rmse)))
}
```



NTNU

Norwegian University of
Science and Technology

SIL
1932