# Introduction

Finvia is an asset management firm based in Germany. Established in 2019 it is a growing, family-led company with now roughly 100 employees. To best allocate recourses in the marketing and sales department, Finvia needs to predict the number of leads arriving through their website and personal contact via telephone or office visit.

In marketing, a lead is any potential costumer, who comes into a first contact with the company and shows interest (Heinrich, 2020, S. 208). The subsequent steps are acquisition and retention (Kumar & Reinartz, 2018, S. 31). Both steps are recourse intensive. An accurate and secure prediction is therefore very helpful for any asset management firm.

The paper is structured in five chapters: Firstly, the data will be presented, broadly and with an exploratory data analysis. Afterwards data preparation includes the steps of data cleaning, wrangling, and merging. Then, the modeling will be explained and lastly our results, variable importance, and the performance metrices will be evaluated.

# Data Understanding

The data is divided into four separate tables: *Leads*, *Ads*, *WebsiteTraffic*, and *Macro* Data, each with a common *Date* column. The beginnings of the tables vary from the 1st of January 2020 (Macro) to the 14th of June 2021 (Ads). All tables last until the 31st of March 2023. The dependent variable *NextDayLeads* is stored in the *Leads* table.

In the table *Ads* numeric values concerning advertising expenditure (*Spend)* and conversion rate (*Impressions* and *Clicks*) are stored. Both variables are split based on platform and funnel. The project is partly based on the assumption that these values have a (lagging) influence on the number of leads.

The table *WebsiteTraffic* focuses on finvia.fo's website traffic, measuring the number of visitors and total time spent on the website. Again, both variables are assumed to have a positive influence on the number of leads by the company. All values until the 15th of August 2021 are equal to zero. Since the company was founded in 2019, a measurement error is assumed (FINVIA, 2023).

The *Macro* table stores macroeconomic data in 14 variables, like the DAX, the electricity and gold price. It only contains 824 variables in the span of 2958 days. This is because all values are only available for banking days.

# Exploratory Data Analysis

An exploratory data analysis was conducted to better understand the data and its structure. Graphs support this progress.

A provided graph showing *NextDayLeads* over time revealed some extreme outliers, for both the *manual* and the *website* type. Since the clear and timely assignment of leads is difficult (Heinrich, 2020), these outliers suggest measurement errors. Figure 1 shows the weekly sum of *NextDayLeads* after filtering the outliers. This filtering will be described in Chapter 3. Figure 1 clearly suggests that the number of leads is rising over time. This trend shows the increasing importance of this project.

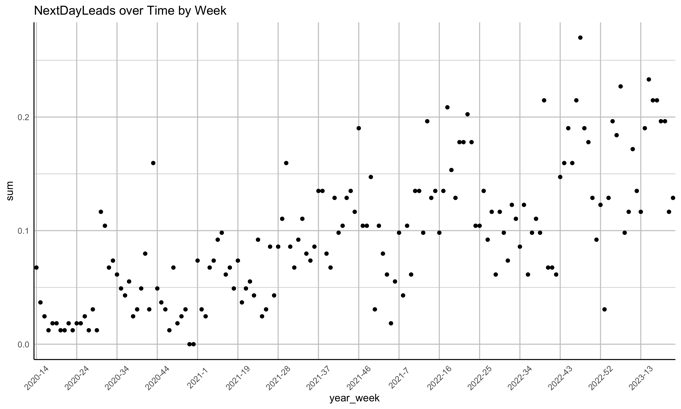


Figure 1: Leads over Time Aggregated by Week, after deleting Outliers.

The analysis of macro-economic data and its relation to leads has found little correlation between the variables. Two variables, the German Electricity price and the US Economy Uncertainty Index stand out. Figure 2 suggests a positive correlation between the German Electricity Price and leads and a negative correlation between the US Economy Uncertainty Index and leads. These correlations are based on the assumptions, that low electricity prices increase fiscal flexibility for consumers and that a low uncertainty gives consumers trust in the financial markets. On the other hand, the colors suggest that these correlations might rather be explained on the time component, which correlates with all three variables (see Figure 1).

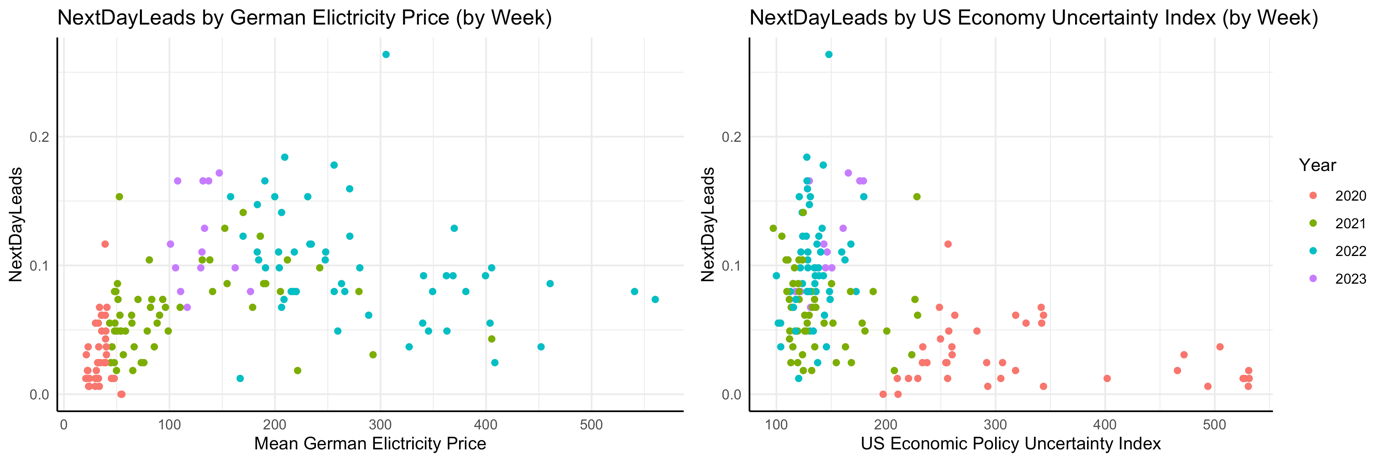


Figure 2: Leads by Week compared to Electricity Price and US Uncertainty Index.

Moreover, relations between the advertisement and traffic data have been examined. While correlations between *Visits* and *TimeSpent* and between *Clicks* and *Impressions* seem to exist, no significant correlation with *NextDayLeads* exists. The highest such correlation is between *Visits* and *NextDayLeads* at around 0.3. Just like the macro-economic data, the code includes graphs to show these values.

# Data Preparation

The goal for data preparation is to have two final data frames, one for manual and one for website leads. These data frames should only include the dependent variable and all the predictors. It should also not include any empty values.

As mentioned in the EDA, the outliers originating from measurement errors are filtered out. An outlier is defined as lying four times the standard deviation away from the mean. This combination of mean and standard deviation is a common approach (Acuña & Rodriguez, 2004). Typically, the multiplier factor is 2 or 3. Here, a higher value was chosen to not delete to many observations. To take the increasing number of outliers into account, both measures are calculated by year. This way over the whole period 18 observations are filtered out.

All website traffic data prior to the 15th of August 2020 had to be deleted as they display wrong values. Also, for the macro-economic data, the observations from weekends and holidays must be filled out. It was decided to take the values from the previous day, since technically the stock index value, for example, on a Saturday is still the value from Friday. To increase readability, all spaces and special characters from the *Macro* data frame are deleted.

The columns *Impressions*, *Clicks* and *Spend* from the *Ads* data frame were aggregated by day. All values from all funnels and platforms were summed up to create one single observation per day. Afterwards new features based on the *Ads* and *Traffic* and *Leads* data were created. The *rollapply* function was used to calculate averages over the past days. This function works like a rolling window, adjusting the upper and lower bound for every observation. These mean values were calculated for *Impressions*, *Clicks*, *Spend*, *Visits* and *TimeSpent* and *NextDayLeads*. In the last case, a lagging function was necessary to avoid the influence of future values to influence the predictors.

Other features created are *month* and *isNextDayWorkDay*. The first stores the month as a factor and functions as a proxy for season. The latter is a binary variable describing whether the next day is a workday or not, meaning weekend or holiday. This variable was created with a specific *holidays* package. The country code for Germany was specified.

To minimize missing data beyond the joining process, an inner join approach was adopted. This strategy entails retaining only those data rows that are present in all data frames. Through this method, the integrity of the dataset is preserved by excluding rows with incomplete information. Keeping all leads observations would have resulted in almost half of observations having incomplete data. This significant proportion harms the modelling process.

Finally, the data frame has been duplicated, providing separate data frames for both manual and website leads. All empty values, resulting from filtering the outliers, from the dependent variable were deleted. Additionally, the respective other NextDayLeads was eliminated, considering their unavailability in practical scenarios, making predictions impractical. The final data frames comprise 30 columns with 588 and 583 observations for manual and website leads, respectively.

# Modeling

The use case at hand is a regression problem. The goal is to predict the numeric variable *NextDaysLeads*. A random Forest model from the *randomForest* package was chosen since a random Forest provides easy interpretability results and clear insights into feature importance. This model also benefits from its ability to show complex and nonlinear relations.

For the Random Forest, three hyperparameters had to be tuned: the features per tree, the number of trees and the node size. Also, the width argument of the *rollapply* function had to be optimized. The features per tree were set to the square root of the number of predictors. This is a common rule of thumb (James, Witten, Hastie, & Tibshirani, 2013, S. 324)

With an increasing number of trees, random Forest cannot overfit. Therefore, the prediction does not get worse and the higher the number of trees, the better. The tradeoff concerns computational efficiency. The right of Figure 3 shows the RMSE with an increasing number of trees. Notably, the test RMSE has a few extreme values with a low number of trees. The higher the number, the less volatile the distribution gets.

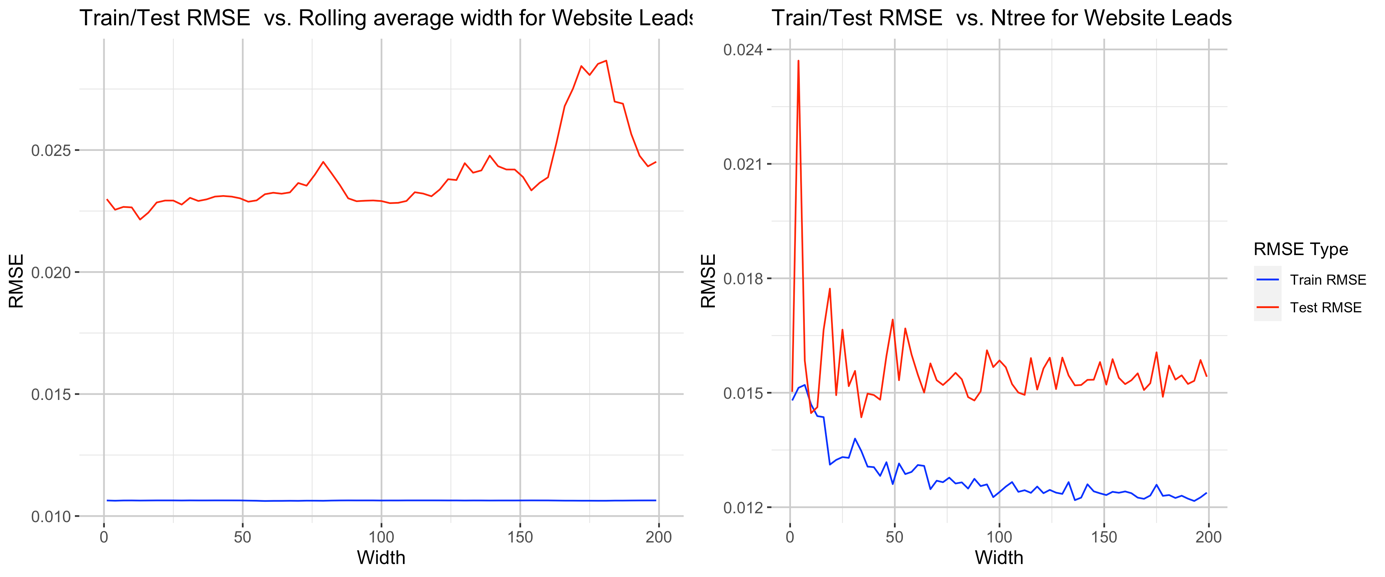


Figure 3: Hyperparameter Tuning of Width Argument and Ntree Setting.

The width argument was optimized in the same way. The left graph in Figure 3 shows the RMSE with an increasing range of the *rollapply* function. The graph is based on a simpler linear regression and suggests that rather the more recent observations are suitable to predict *NextDayLeads*. A closer analysis of lower values, available in the code, supports this conclusion. The width-argument was therefore set to three. A second predictor with a width of 10 was also created. Because of the relative big difference in these variables, the correlation is relatively small and they potential displays different aspects of the lagging effect.

# Evaluation