HR Analytics Employee Attrition & Performance

Big Data and Databases Group Project

Group 18

Fran Stilinović, Giulia Paggini, Nikola Nenkov,
Nina Rubeša, Pavle Lalić, Pierluigi Mancinelli

OUTLINE

1. INTRODUCTION
2. UNIVARIATE ANALYSIS
3. BIVARIATE ANALYSIS
4. DATA PREPARATION
5. DATA ANALYSIS
6. OUTCOMES
7. MANAGERIAL IMPLICATIONS

INTRODUCTION

PREDICTION OF EMPLOYEE ATTRITION

DATASET SUMMARY

The dataset used for this project, IBM Employee Attrition (https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset), contains information on whether an employee left the company or not, along with information on potential factors that could lead to attrition such as age, distance from home, work-life balance and others.

The dataset consists of 1470 observations and 35 columns. The target is a binary variable, Attrition, which is strongly unbalanced: 1233 “No” against only 237 “Yes”.

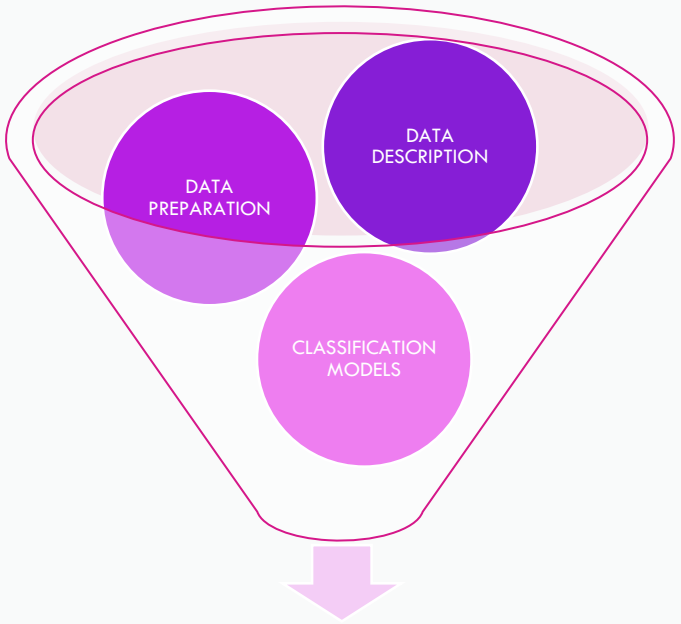
RESEARCH OBJECTIVE

The mission of our “company” is to predict future employee attrition and determine the factors that might cause it, and hence to recognize potential improvements that could be implemented by a company in order to maintain its workforce.

Reduced attrition enables a company to have more employees that possess knowledge of internal processes of the company. In addition, a dose of stability is present if employees are satisfied and are not leaving the company.

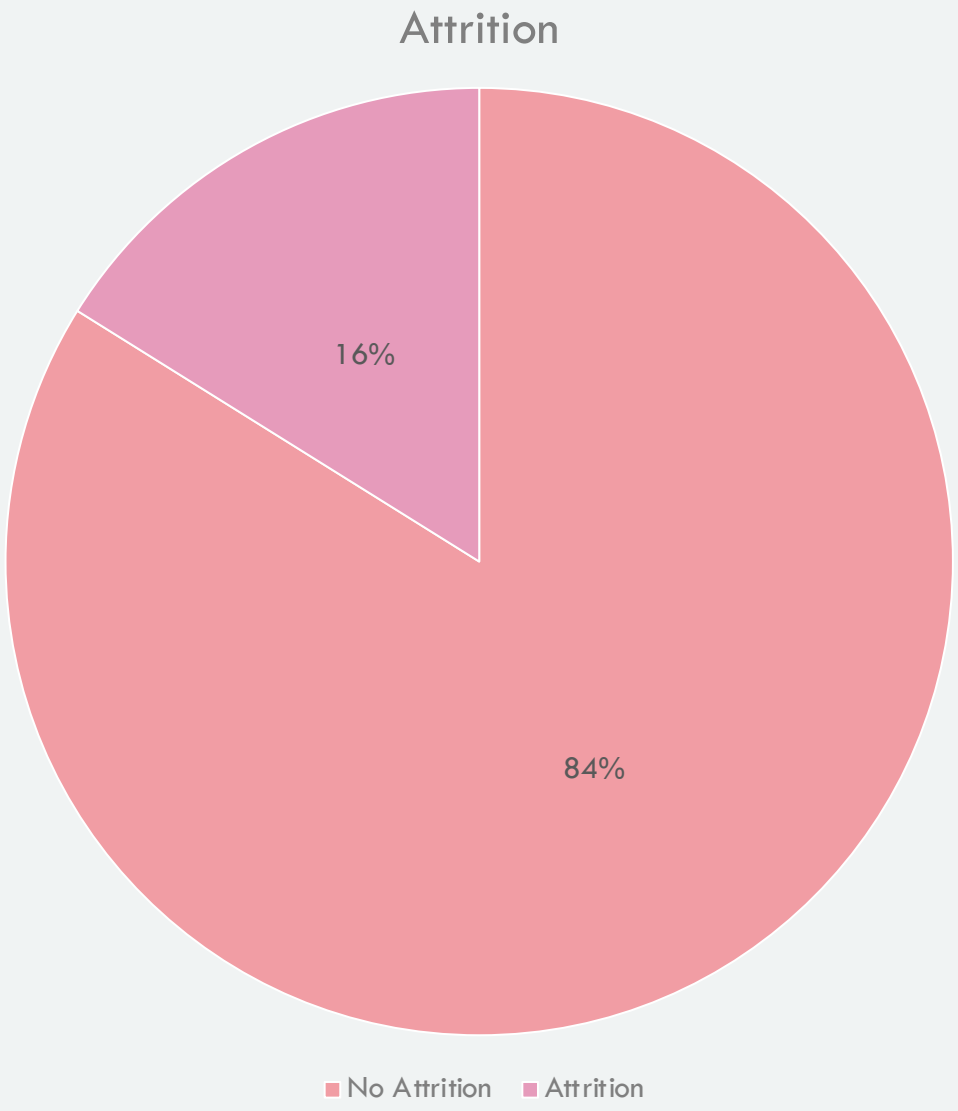
On the contrary, higher attrition leads to a higher turnover of new candidates that need to be selected and trained to work in the company, and thus to reduced efficiency. Furthermore, selection and onboarding processes impose higher costs on the company.

PIPELINE



After data description and preparation, Logistic Regression, Random Forest and Gradient Boosting models are implemented for the binary classification task described.

UNIVARIATE ANALYSIS



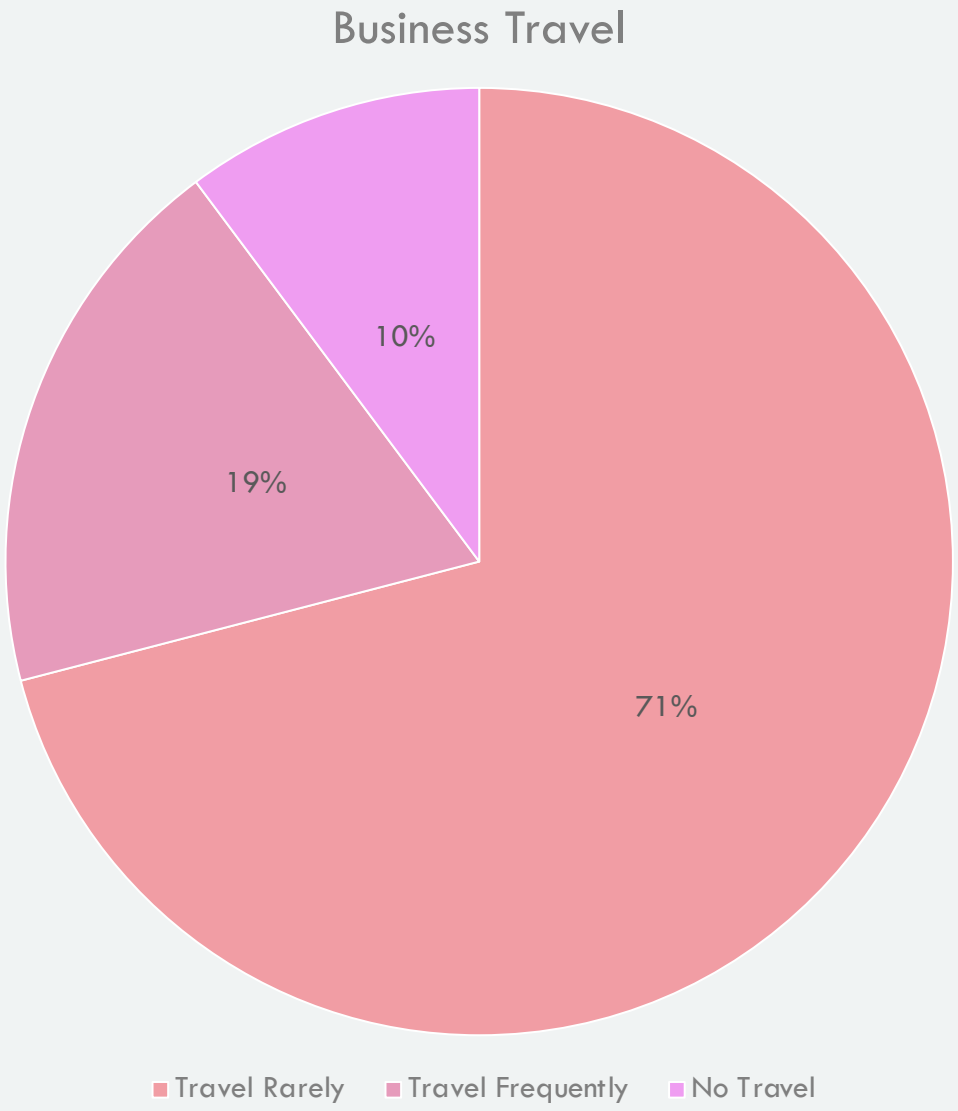
Attrition

Nature: Categorical, nominal

Description: Whether employees have left the company or not

Categories: Attrition, No Attrition

Insights: Strongly unbalanced sample, with 84% of the employees remaining at the company



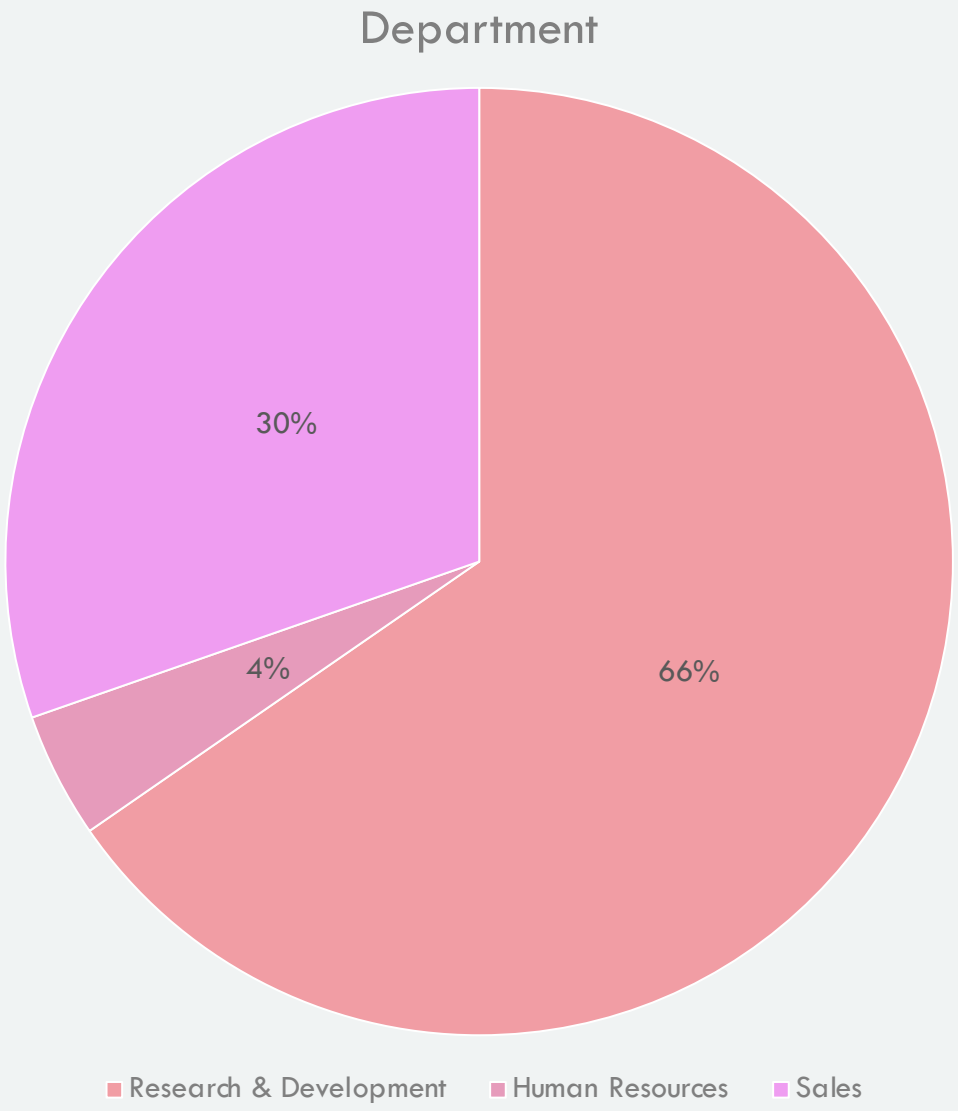
Business Travel

Nature: Categorical, nominal

Description: How often do employees travel for business

Categories: Travel Rarely, Travel Frequently, No Travel

Insights: 90% of employees do travel for business; however, a vast majority of them travels rarely



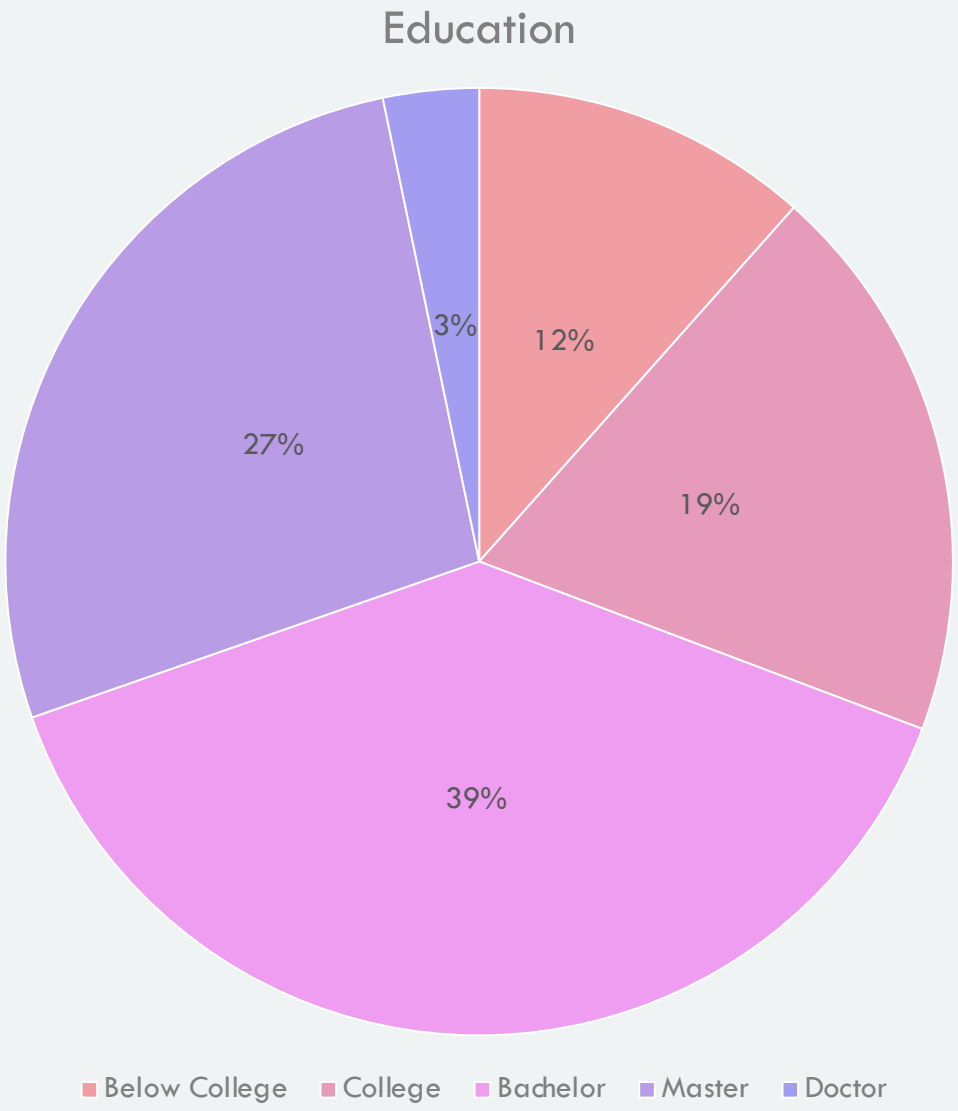
Department

Nature: Categorical, nominal

Description: Departments employees are working in

Categories: Research & Development, Human Resources, Sales

Insights: Unbalanced representation with two thirds of employees being in Research & Development department



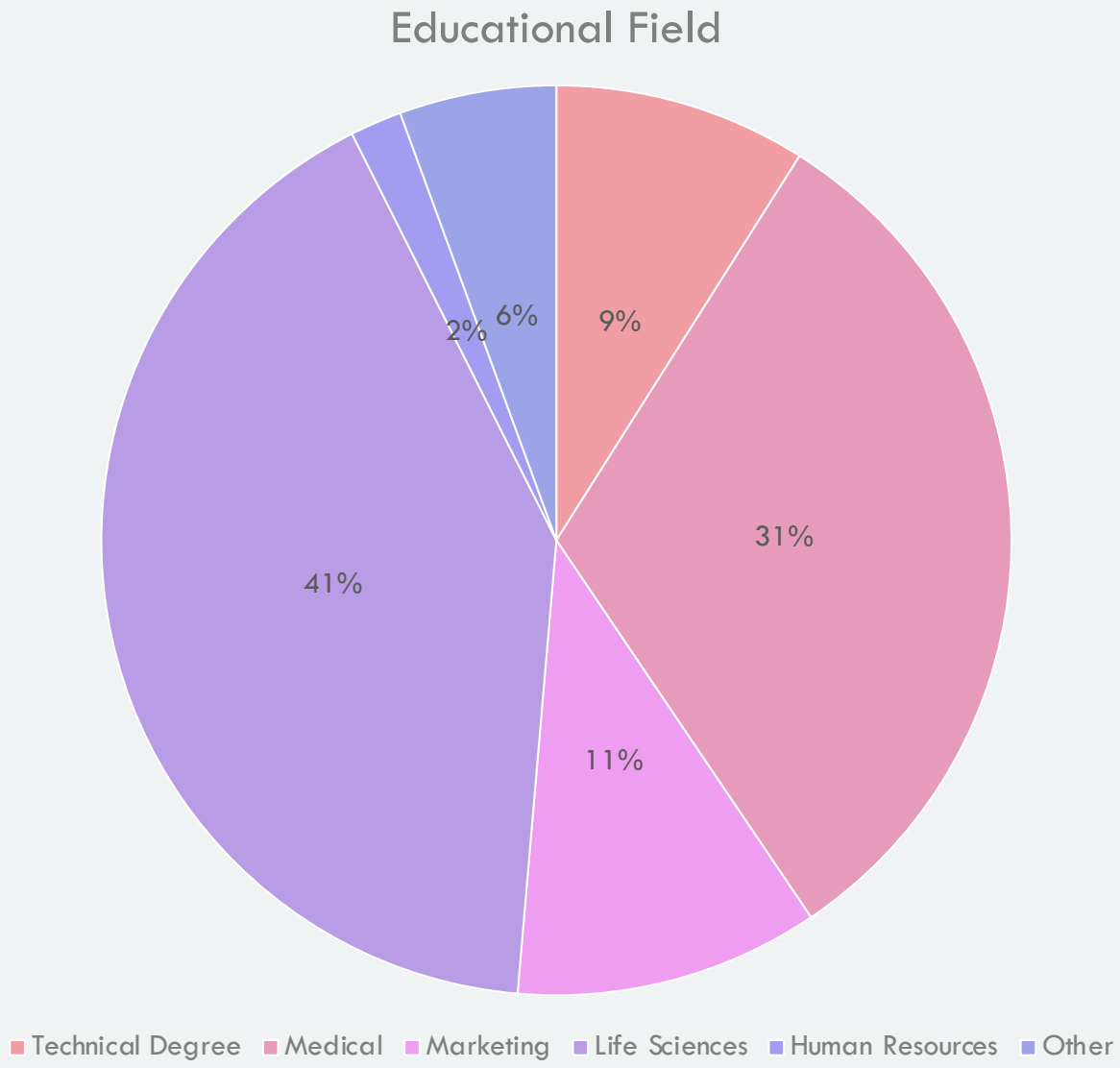
Education

Nature: Categorical, ordinal

Description: Level of employee's education

Categories: Below College (1), College (2), Bachelor (3), Master (4), Doctor (5)

Insights: We have kind of a normal distribution of the level of education, with most of individuals having either College, Bachelor or Master degree



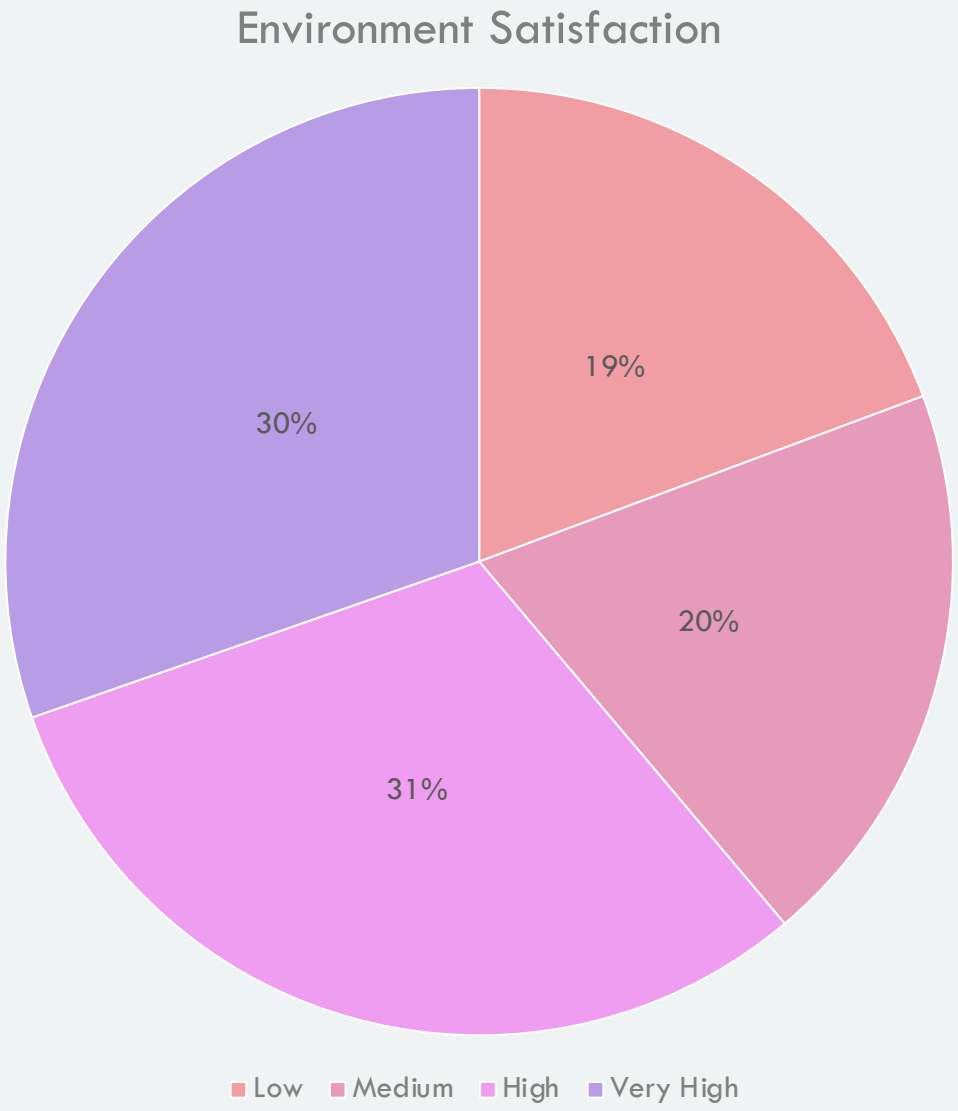
Education Field

Nature: Categorical, nominal

Description: Field in which employees pursued their education

Categories: Technical Degree, Medical, Marketing, Life Sciences, Human Resources, Other

Insights: The greatest number of employees pursued a degree in Life Sciences and Medicine, while other fields account for smaller percentages



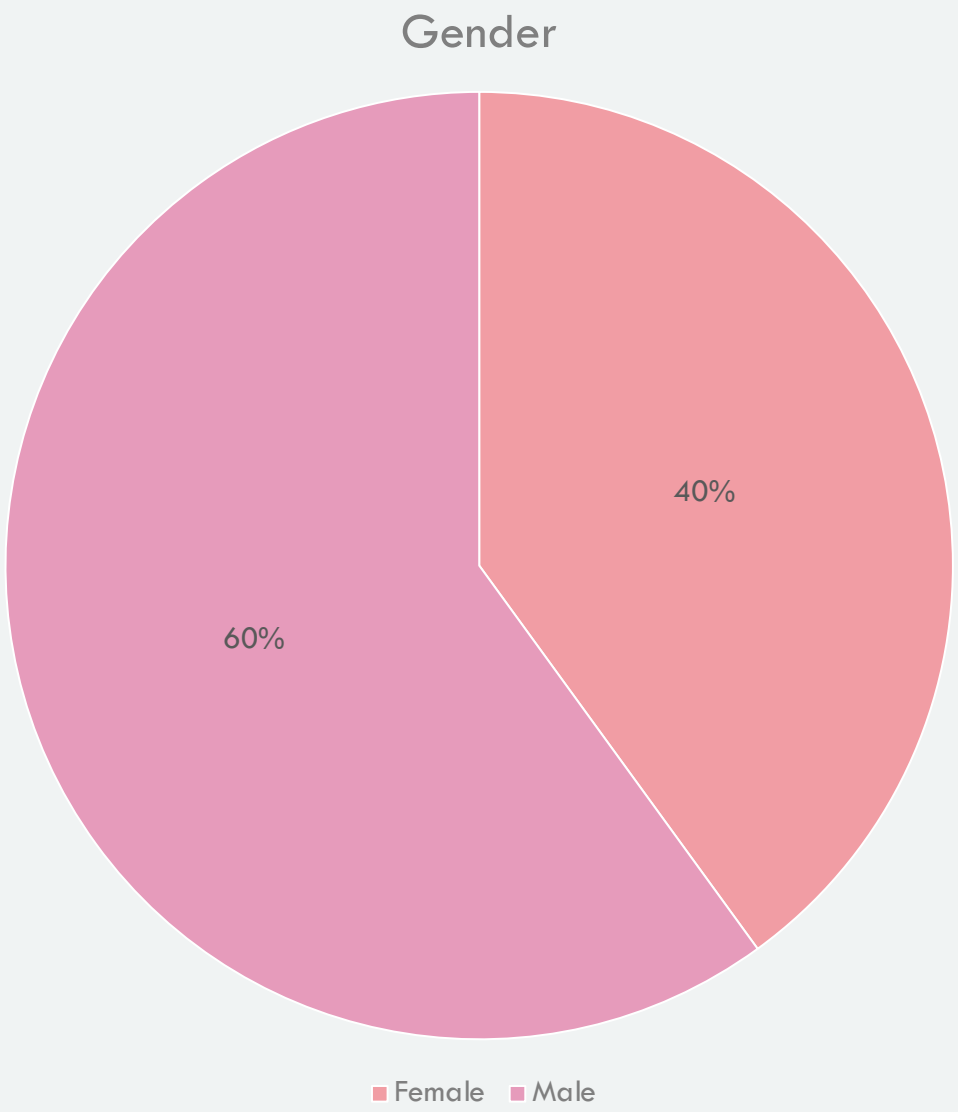
Environment Satisfaction

Nature: Categorical, ordinal

Description: How employees are satisfied with their working environment

Categories: Low (1), Medium (2), High (3), Very High (4)

Insights: We have quite a balanced sample when it comes to environment satisfaction



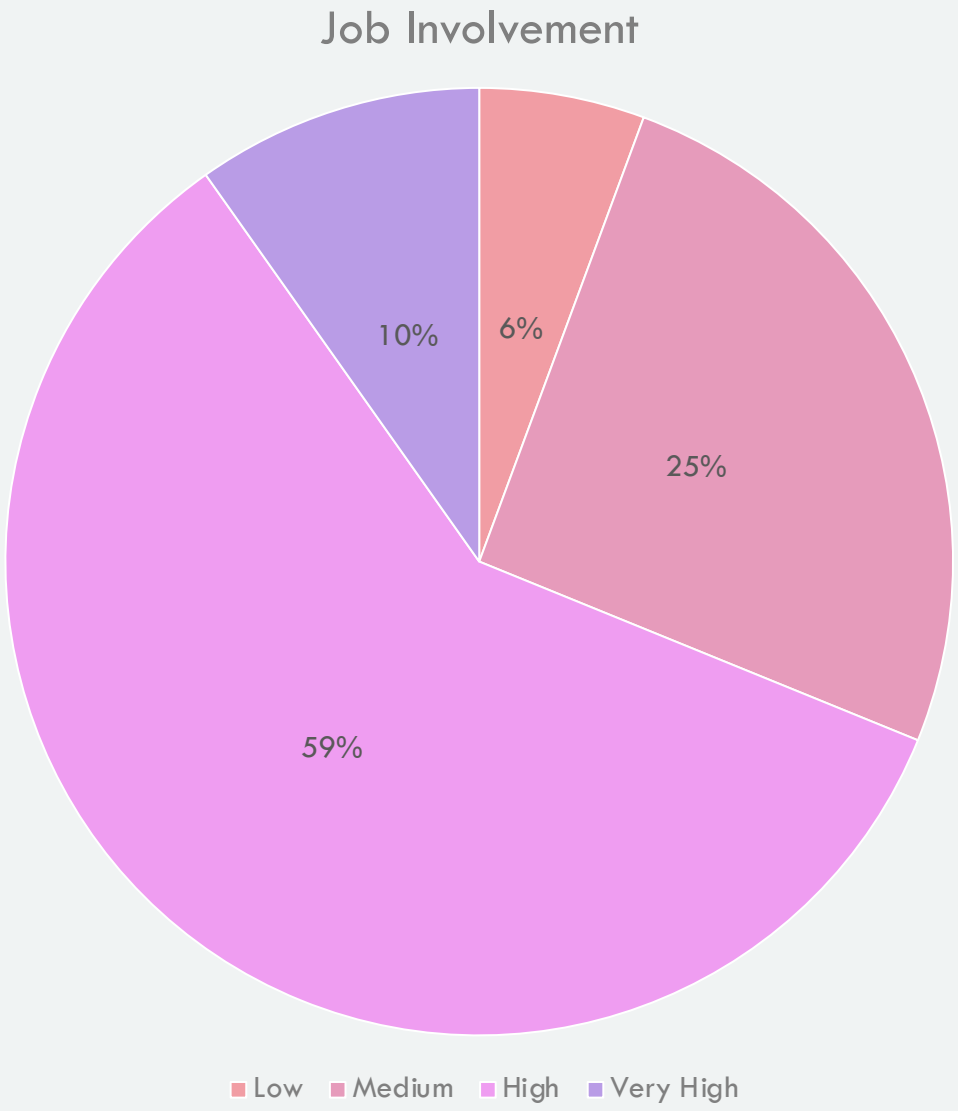
Gender

Nature: Categorical, nominal

Description: Employee's gender

Categories: Female, Male

Insights: Slightly unbalanced dataset, with sixty percent of male employees



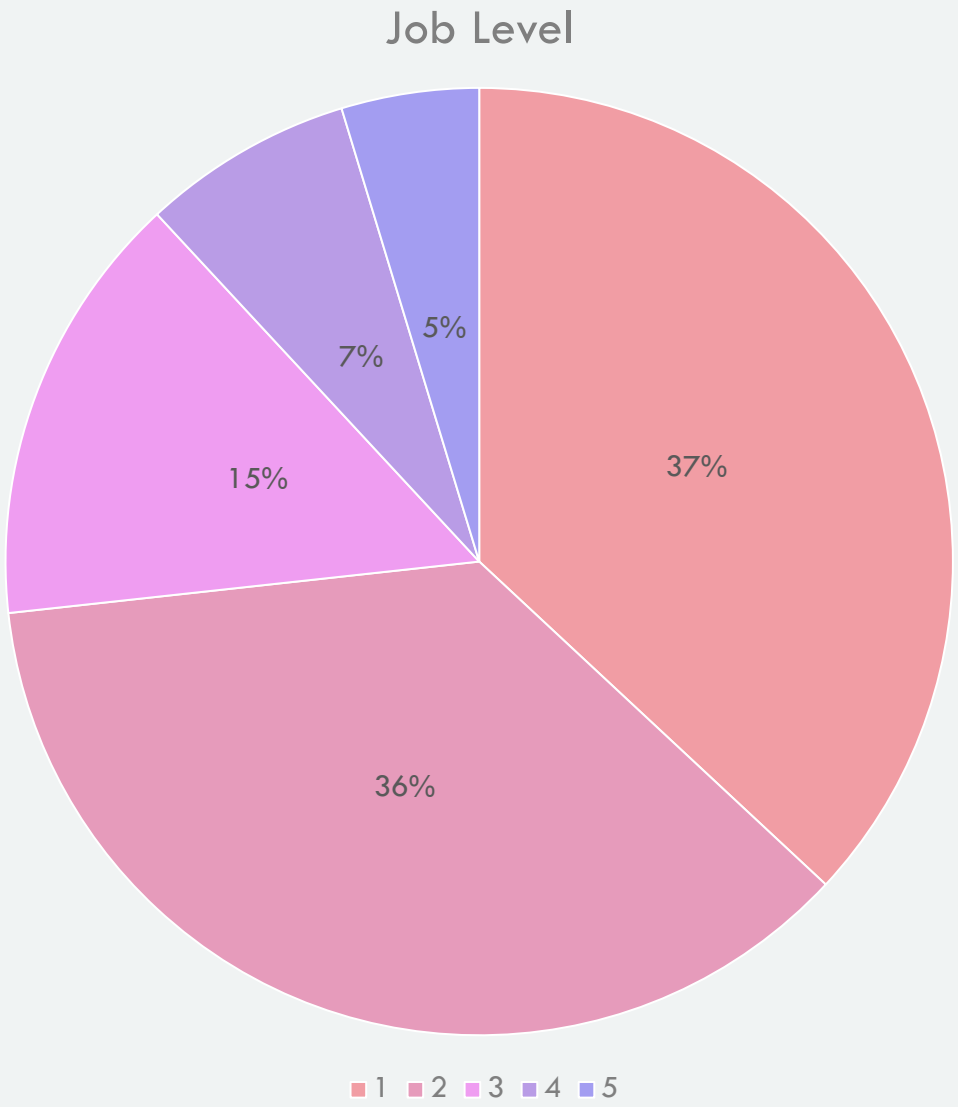
Job Involvement

Nature: Categorical, ordinal

Description: Level of involvement of employees in everyday tasks

Categories: Low (1), Medium (2), High (3), Very High (4)

Insights: The biggest portion of employees has a medium or high level of job involvement



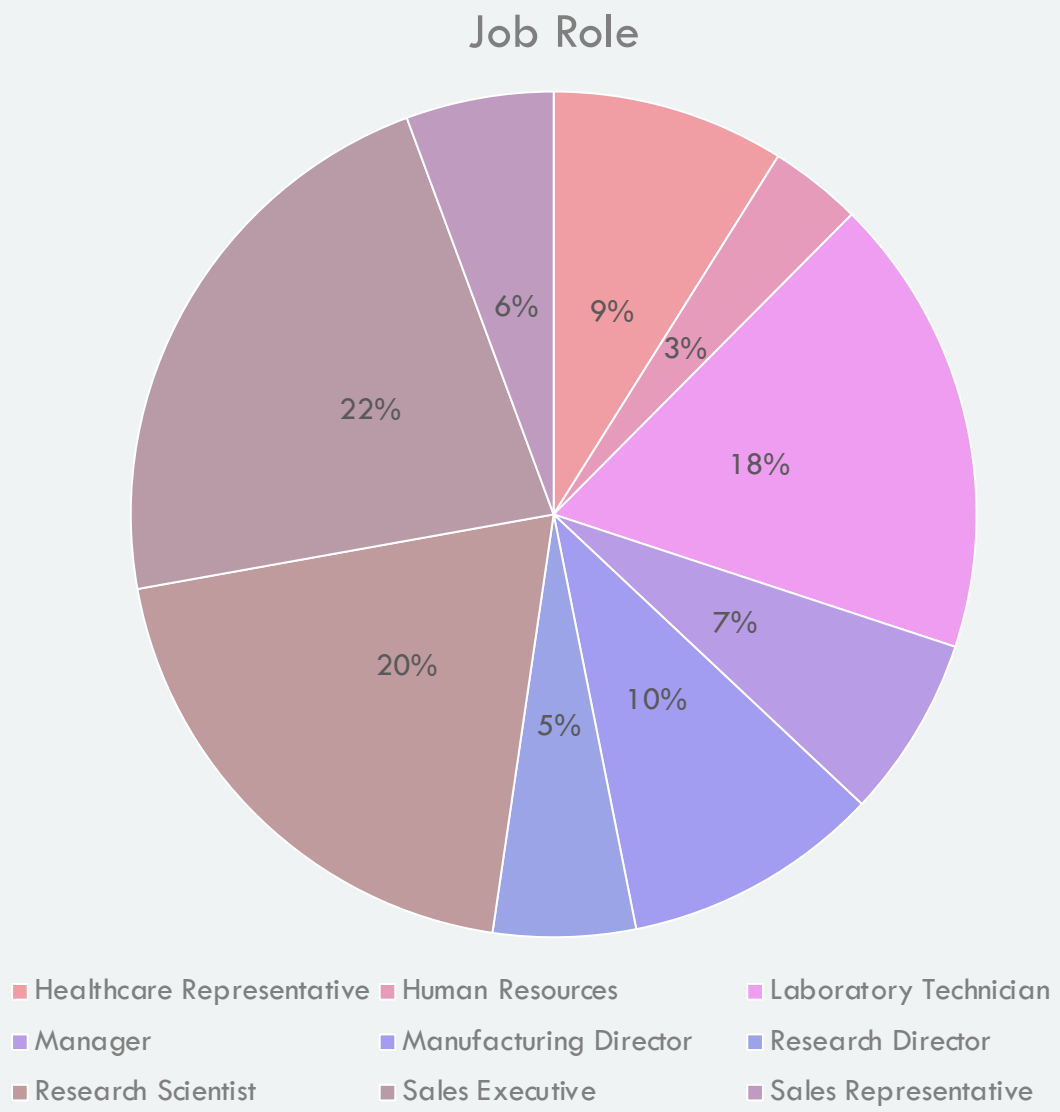
Job Level

Nature: Categorical, ordinal

Description: Employees' job level – meaning is not clear

Categories: 1, 2, 3, 4, 5

Insights: Unbalanced representation with majority of employees being at low job levels, i.e. 1 and 2



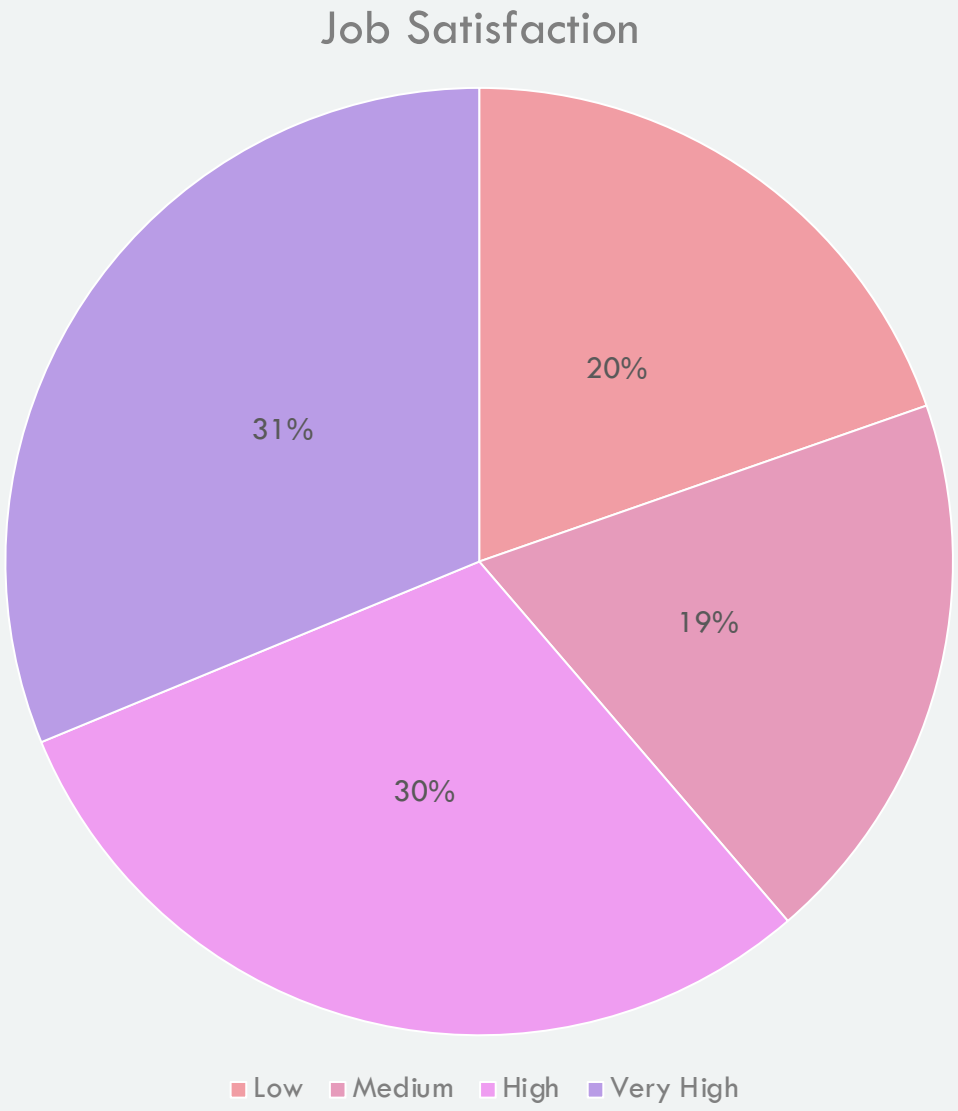
Job Role

Nature: Categorical, nominal

Description: Job roles of employees

Categories: Healthcare Representative, Human Resources, Laboratory Technician, Manager, Manufacturing Director, Research Director, Research Scientist, Sales Executive, Sales Representative

Insights: Unbalanced representation with most of the employees being Research Scientists, Sales Executives, and Laboratory Technicians



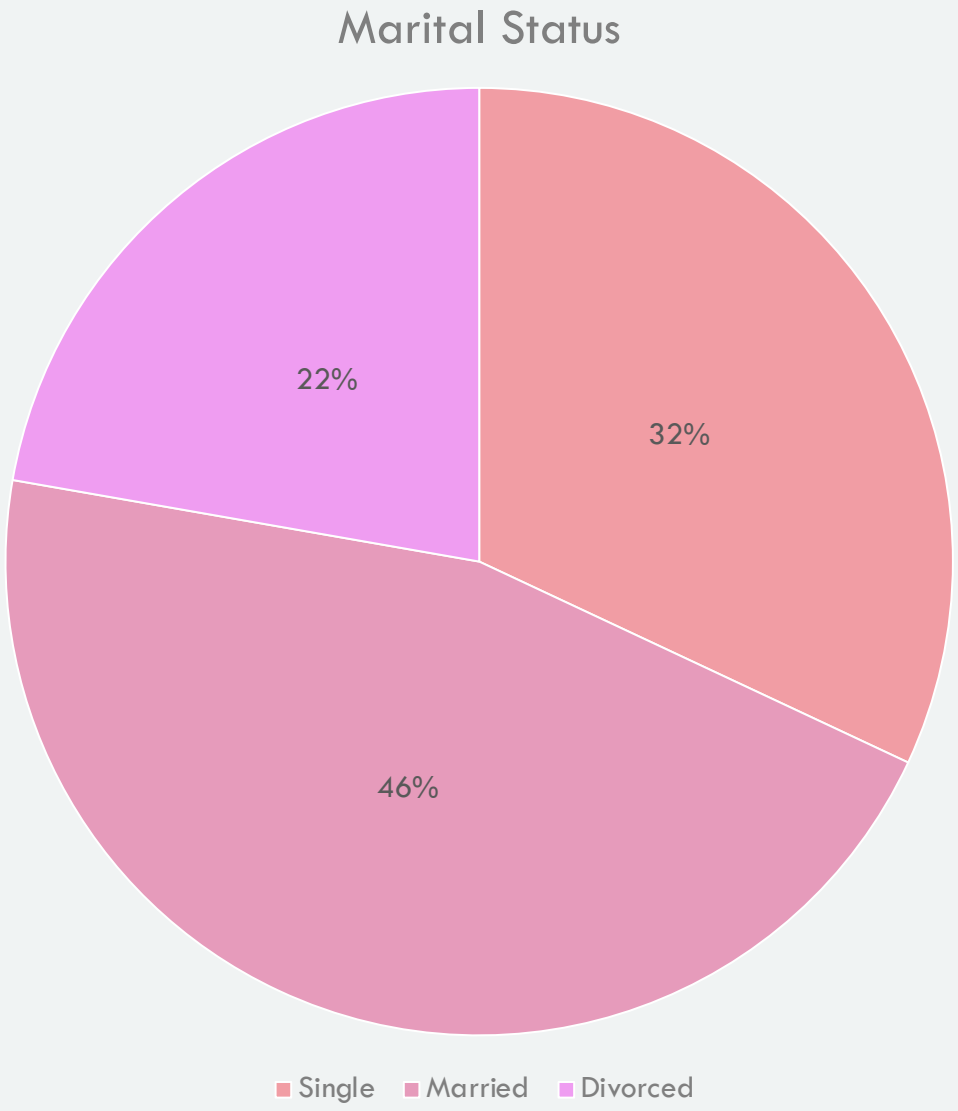
Job Satisfaction

Nature: Categorical, ordinal

Description: Level of satisfaction of employees with the jobs they do

Categories: Low (1), Medium (2), High (3), Very High (4)

Insights: We have a balanced representation of all four categories



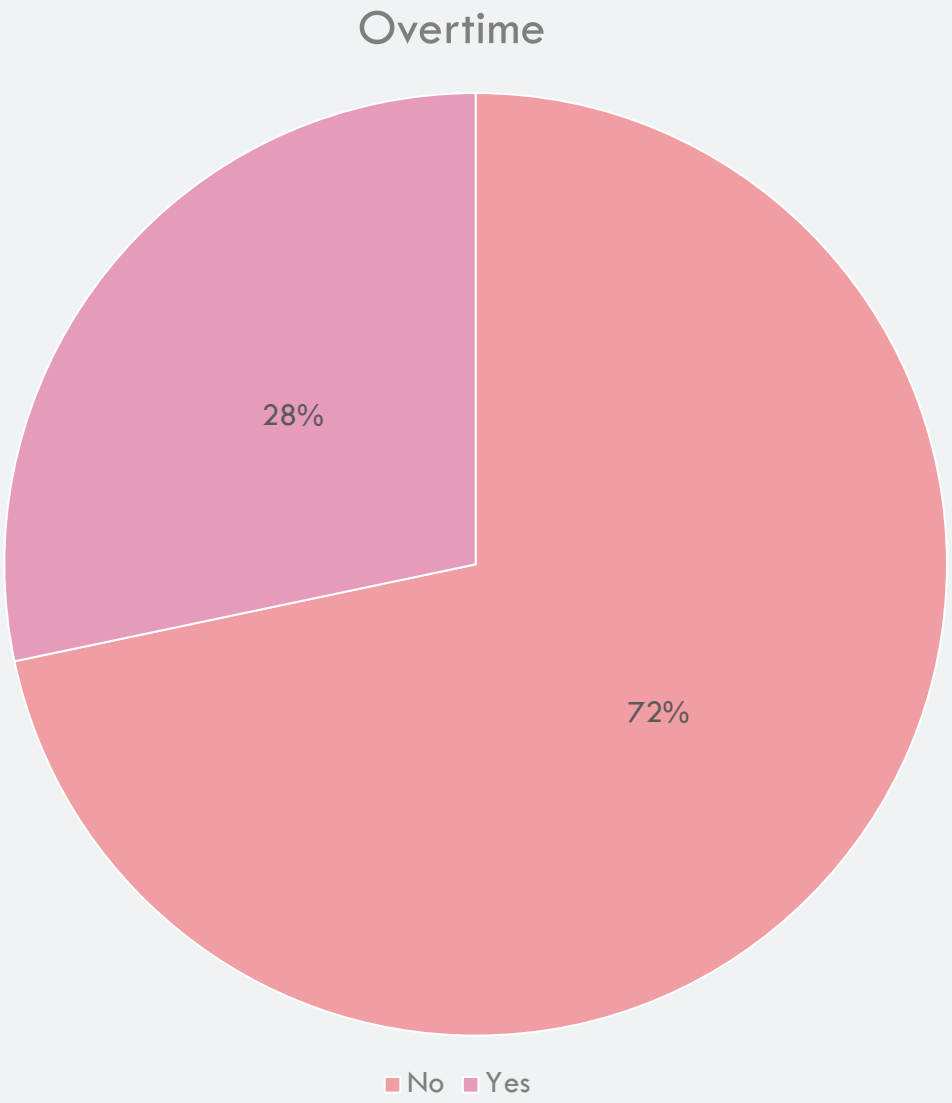
Marital Status

Nature: Categorical, nominal

Description: Employee's marital status

Categories: Single, Married, Divorced

Insights: Unbalanced representation of the three categories, with slightly less than a half of employees being married



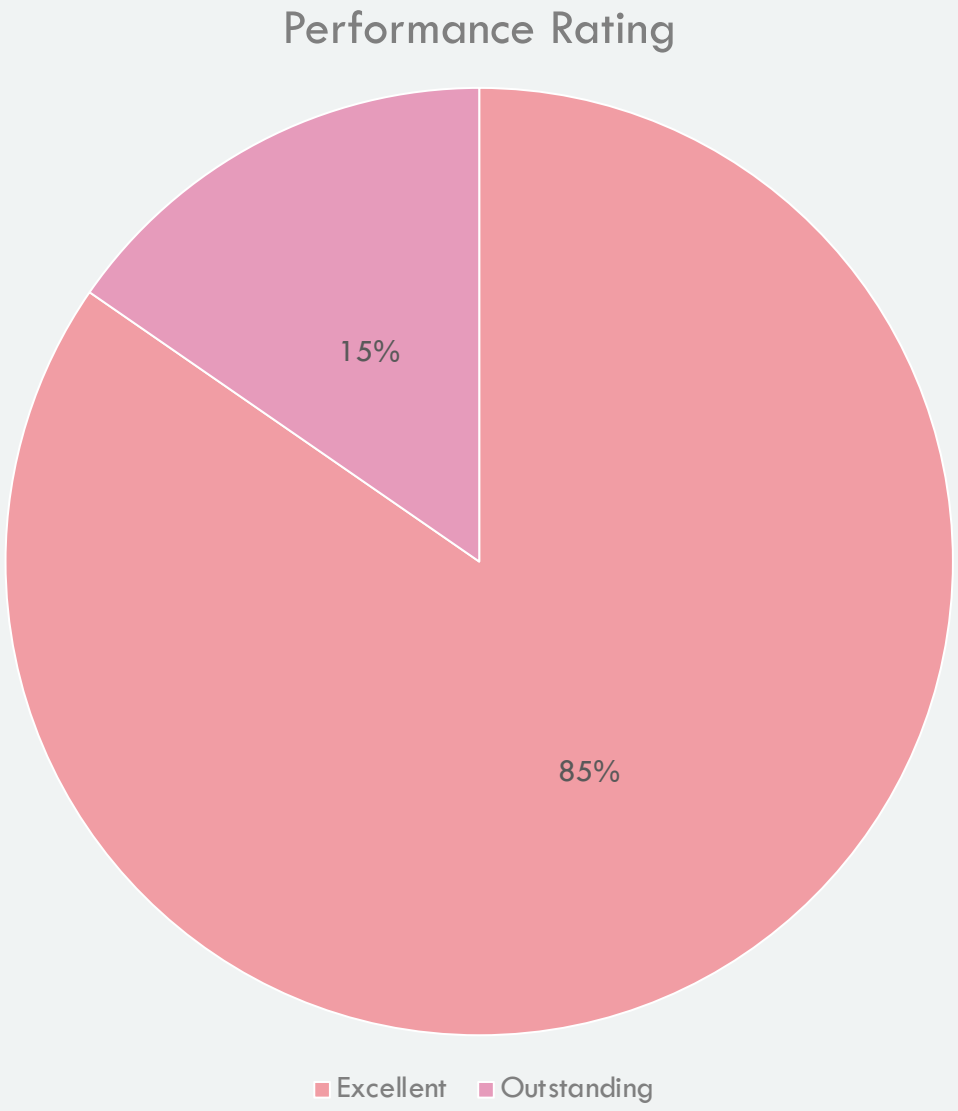
Overtime

Nature: Categorical, nominal

Description: Whether employees work overtime or not

Categories: Yes, No

Insights: Unbalanced representation with most of the employees not working overtime



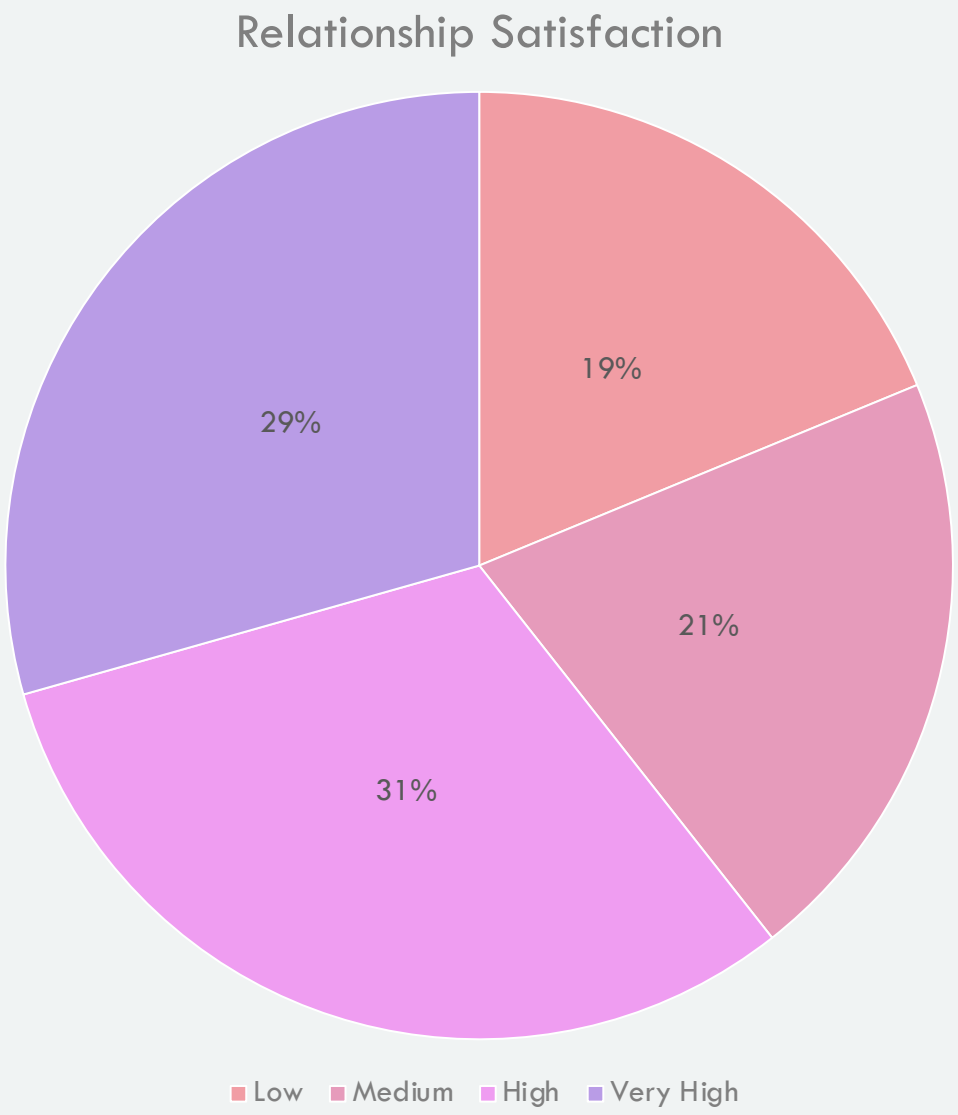
Performance Rating

Nature: Categorical, ordinal

Description: How employees performed in previous periods

Categories: Low (1), Good (2), Excellent (3), Outstanding (4)

Insights: Unbalanced representation with majority of employees having excellent, and none having low or good ratings



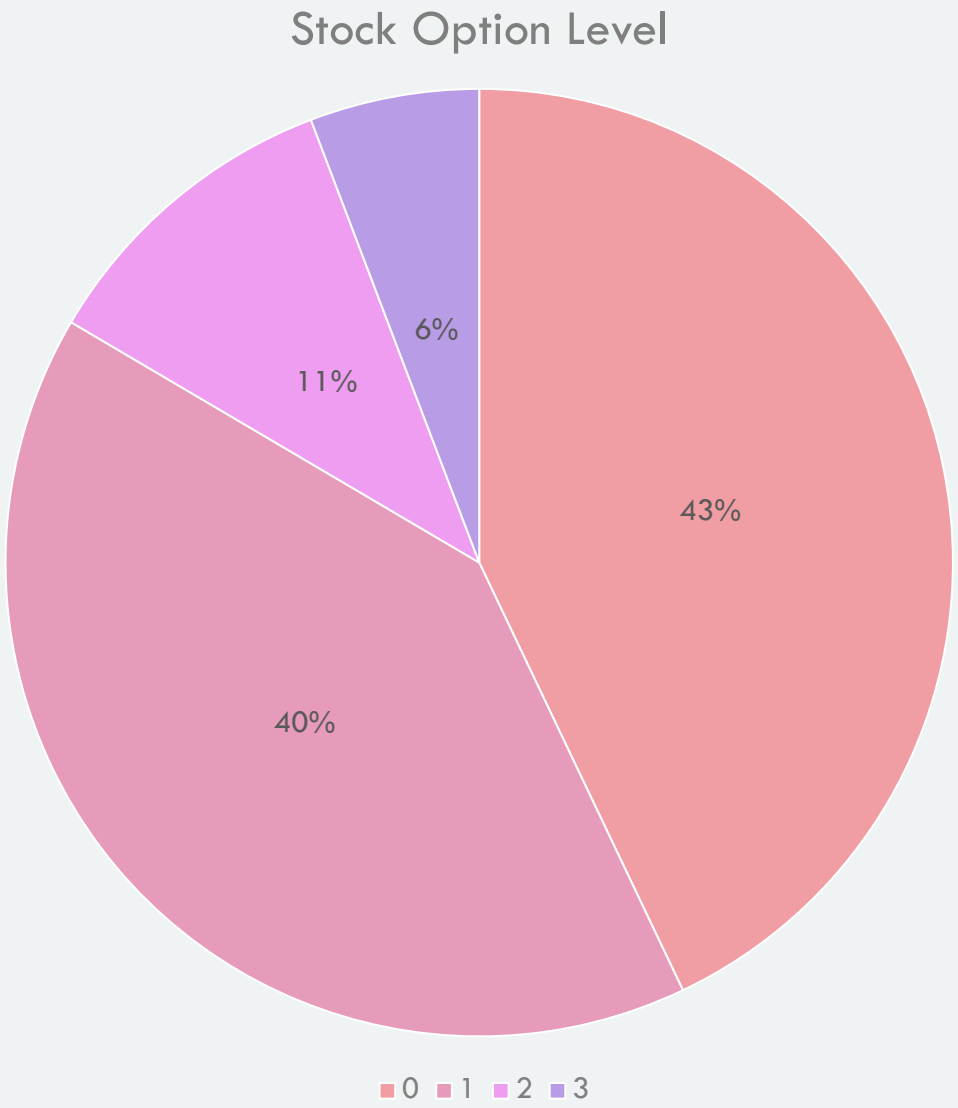
Relationship Satisfaction

Nature: Categorical, ordinal

Description: How satisfied employees are with relationships inside the company

Categories: Low (1), Medium (2), High (3), Very High (4)

Insights: Quite balanced representation of all four categories, with slightly more employees with higher levels of satisfaction



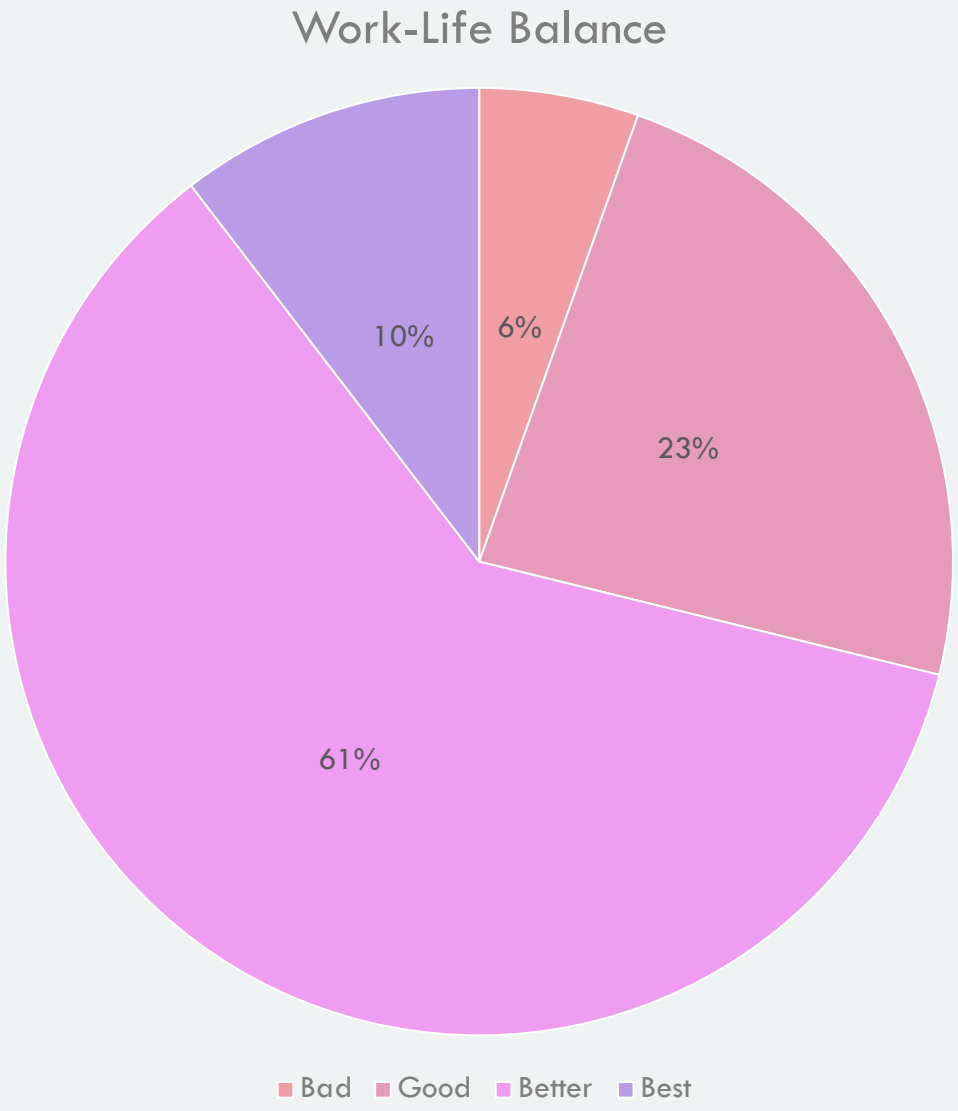
Stock Option Level

Nature: Categorical, ordinal

Description: What level of stock options are offered to employees

Categories: 0, 1, 2, 3

Insights: Unbalanced representation with majority of employees having lower stock options



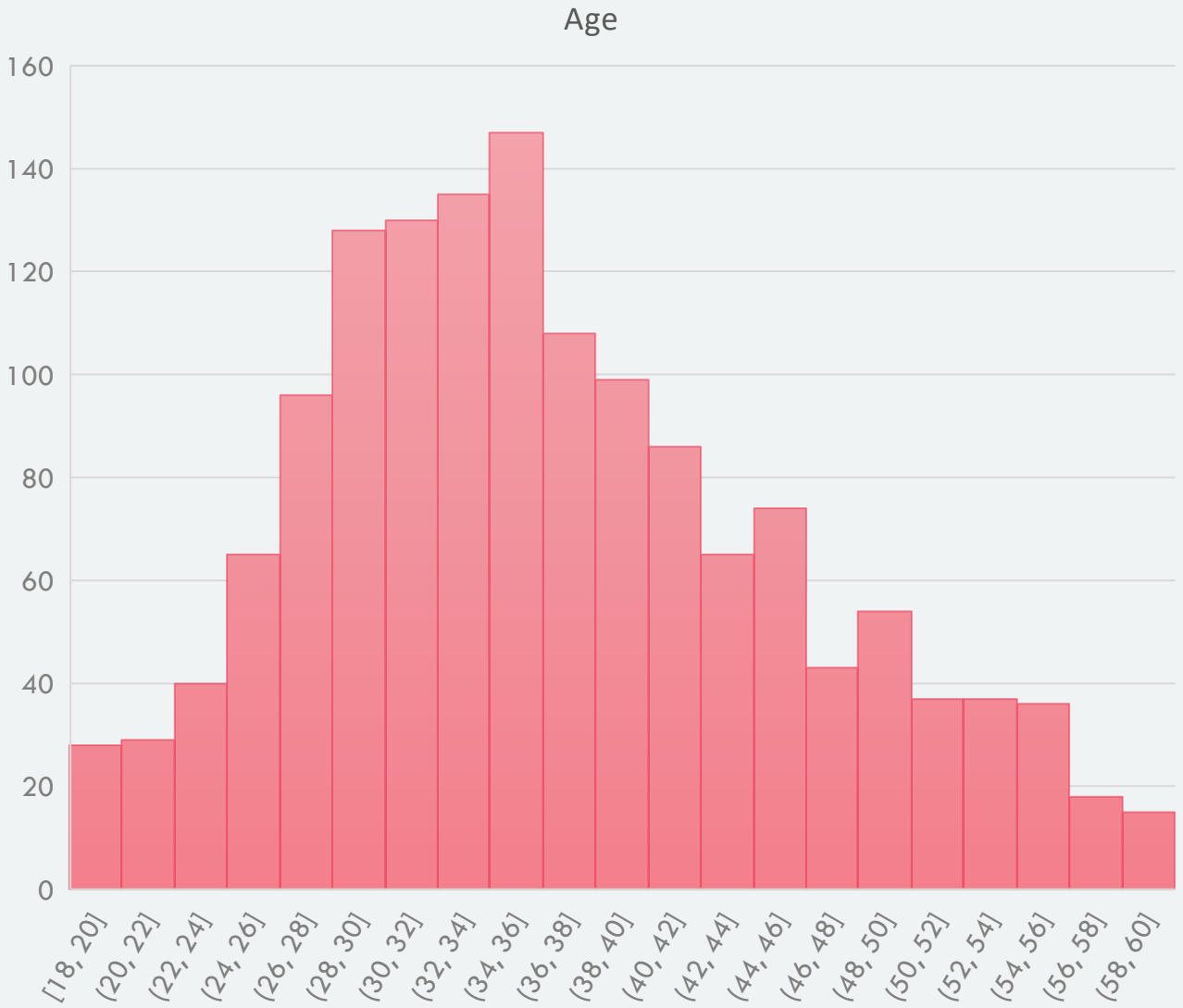
Work-Life Balance

Nature: Categorical, ordinal

Description: How employees rate their work-life balance

Categories: Bad (1), Good (2), Better (3), Best (4)

Insights: Unbalanced representation with great majority of employees in category Better



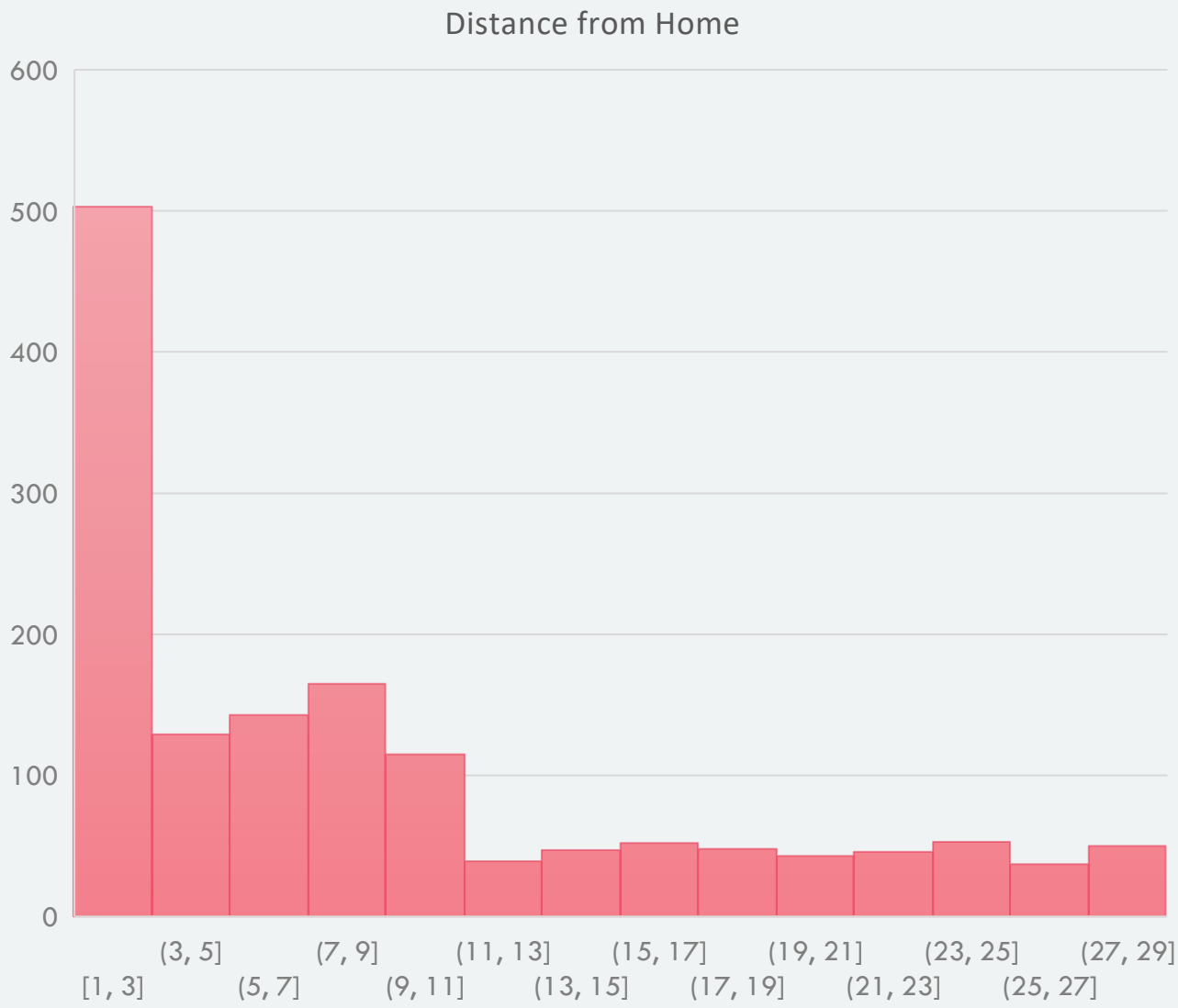
Age

Nature: Numerical, Discrete

Description: Age of the employees

Range: From 18 to 60

Insights: Mean and median of this variable are equal to 36, while standard deviation equals 9; distribution is slightly skewed to the left and close to normal



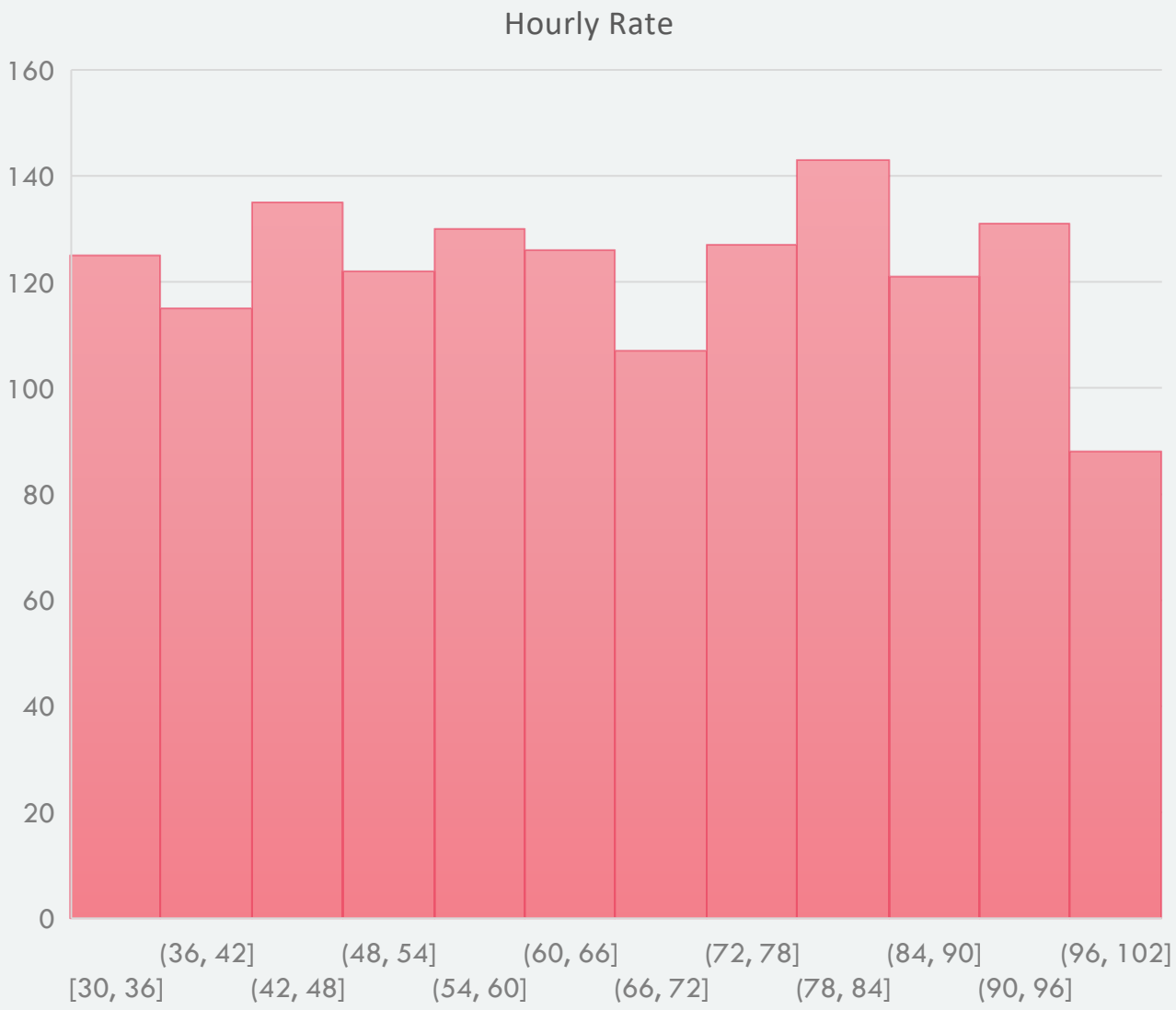
Distance From Home

Nature: Numerical, Discrete

Description: Distance between employees' offices and their homes

Range: From 1 to 29

Insights: Mean of the variable equals 9, while median equals 7; standard deviation equals 8; as it can be seen, most of the employees live close to the office



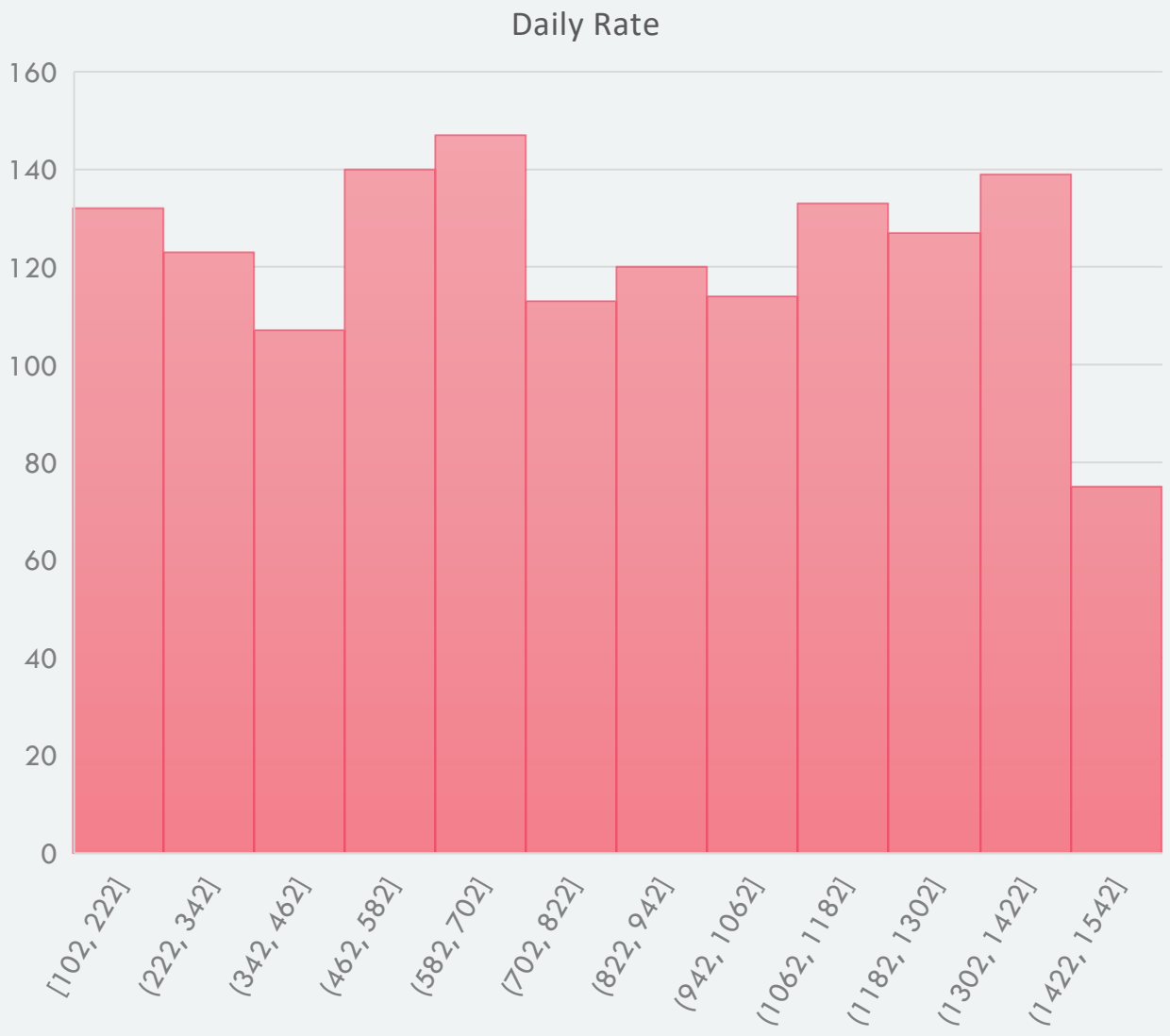
Hourly Rate

Nature: Numerical, discrete

Description: Count of the employees with hourly rate in specific range intervals

Range: From 30 to 102

Insights: Mean and median of this variable equals 66, and standard deviation equals 20; chart shows similar distribution across intervals with smaller frequencies around mean and maximum



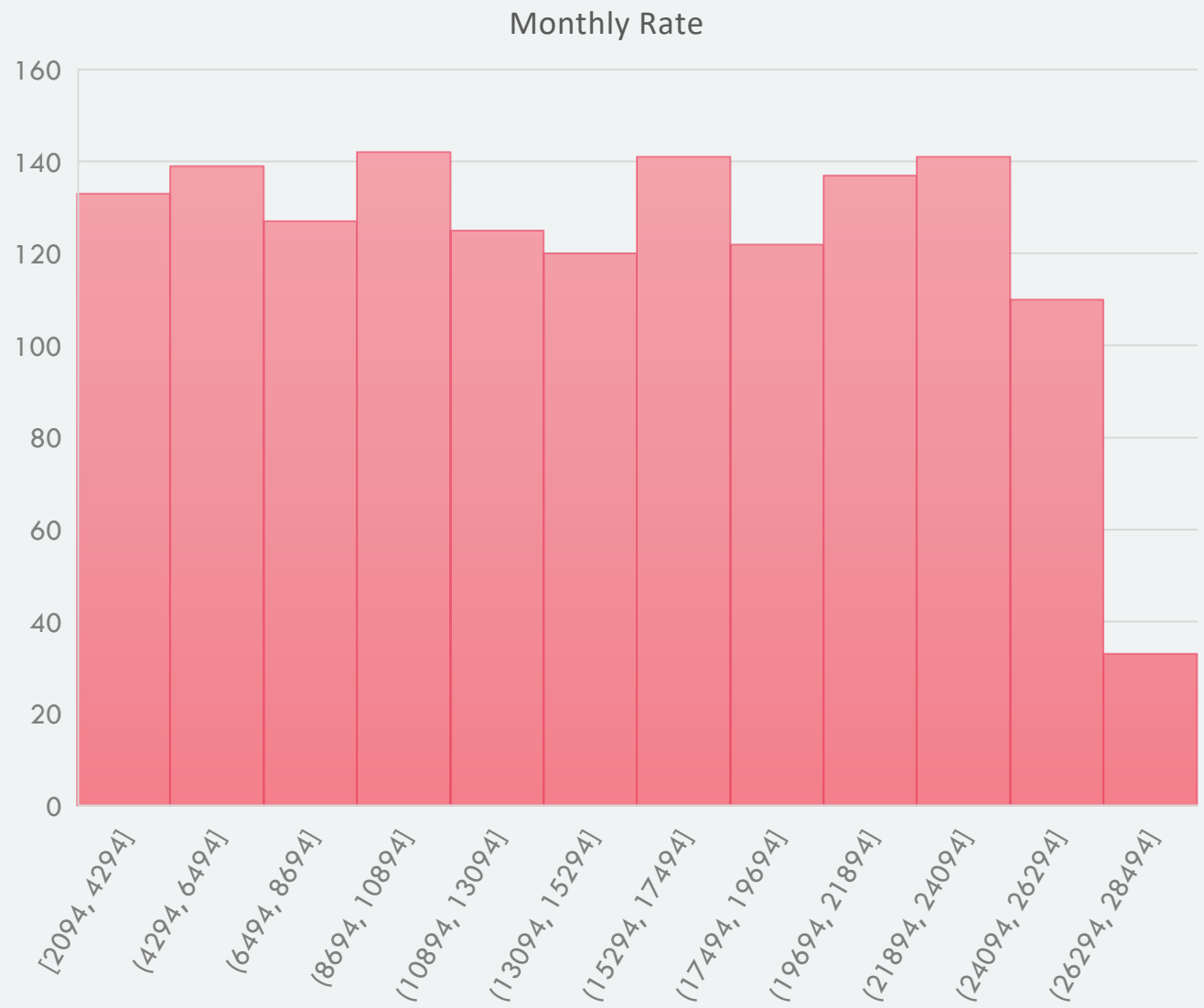
Daily Rate

Nature: Numerical, discrete

Description: Count of the employees with daily rate in specific range intervals

Range: From 102 to 1542

Insights: Mean of the variable is 802; frequencies between intervals are relatively similar, with the lowest frequency in the highest interval between 1422 and 1542



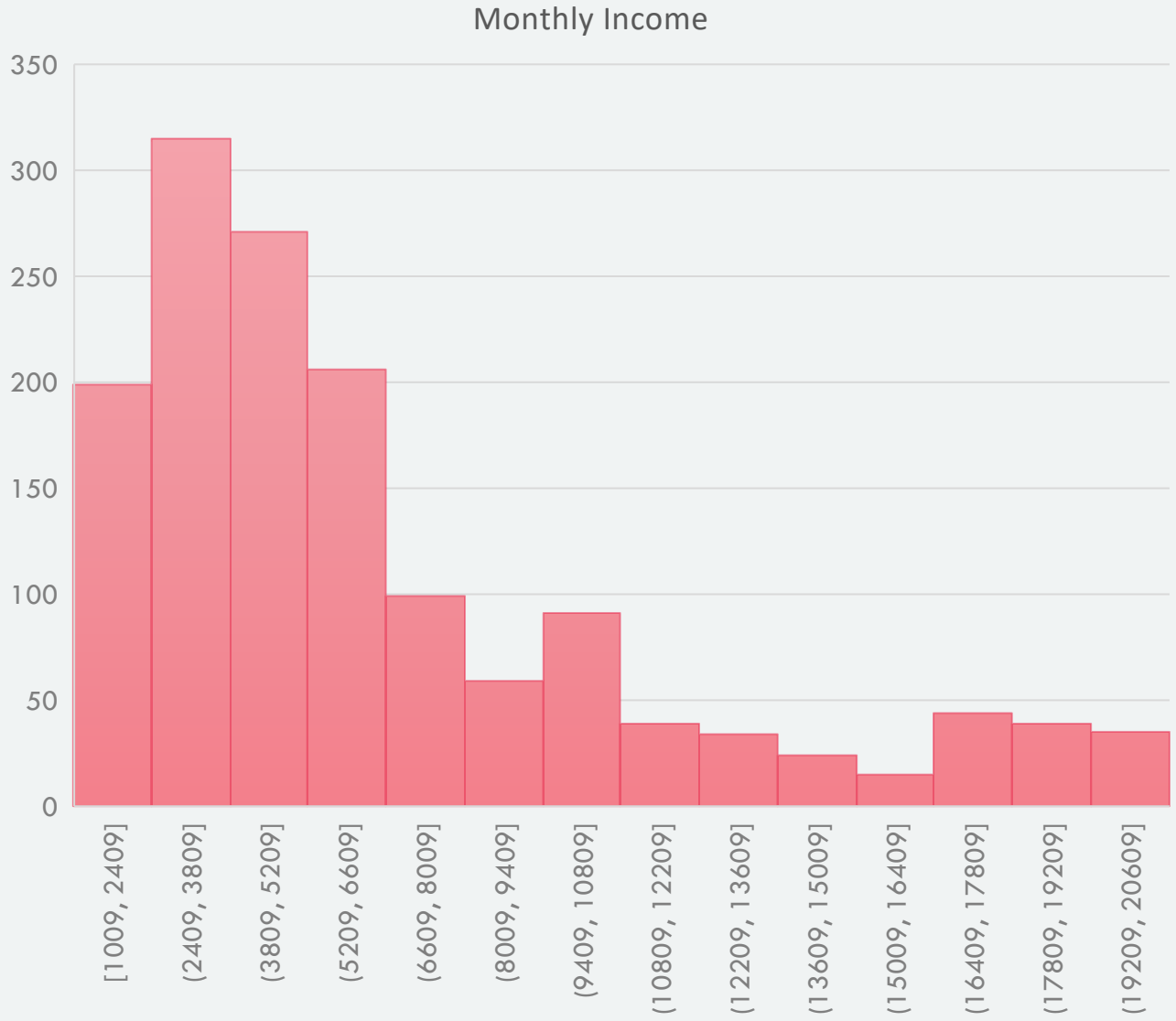
Monthly Rate

Nature: Numerical, discrete

Description: Count of the employees with monthly rate in specific range intervals

Range: From 2,094 to 28,494

Insights: Mean of this variable is 14,313 and median is 14,235; standard deviation is 7,118; distribution is close to uniform, with smaller frequencies among the highest intervals



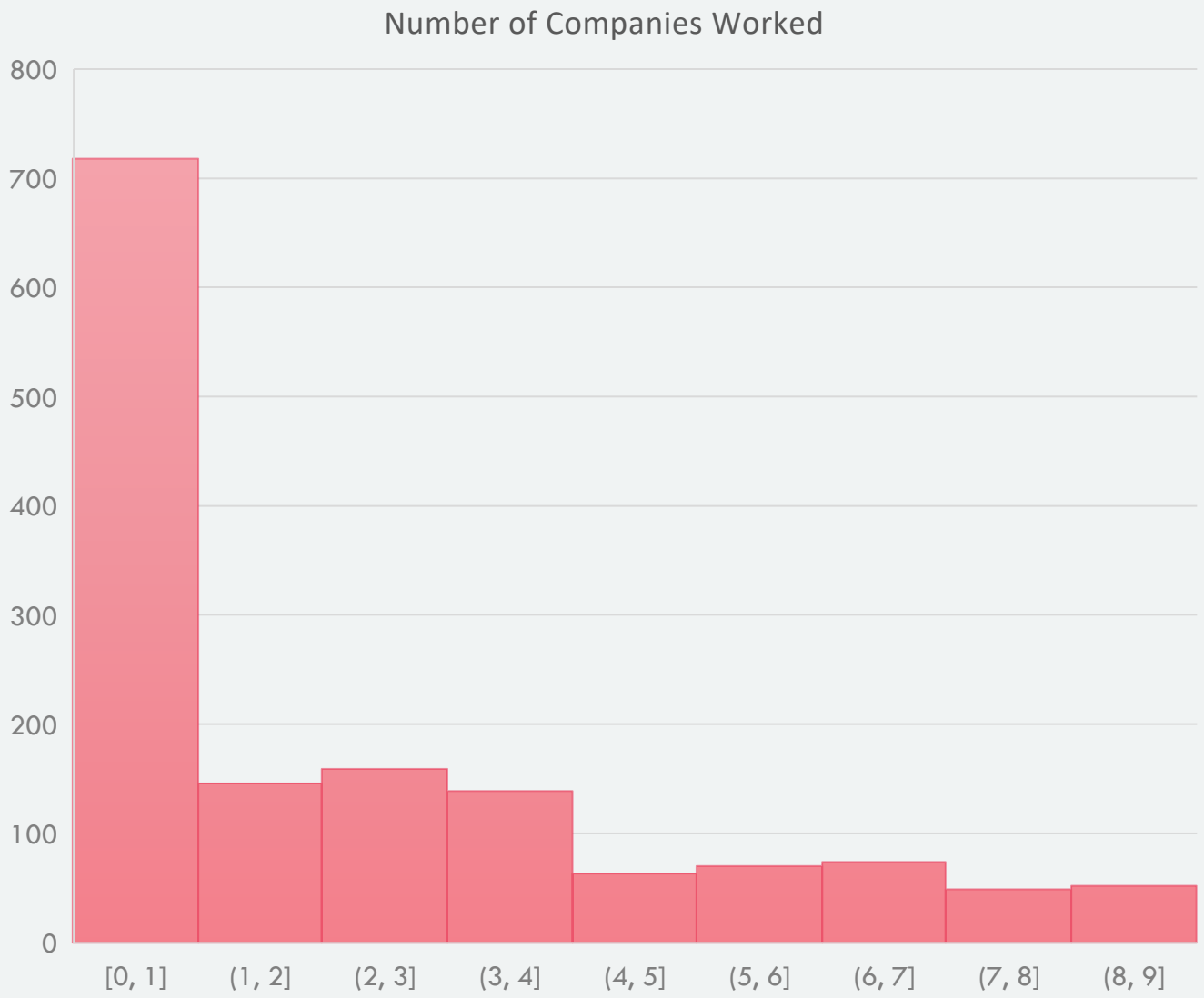
Monthly Income

Nature: Numerical, discrete

Description: How much money employees earn per month

Range: From 1,009 to 20,609

Insights: Mean of the variable is 6,503, and median 4,919; standard deviation is 4,708; distribution is skewed to the left and most of the employees can be found in interval between 2,409 and 5,209



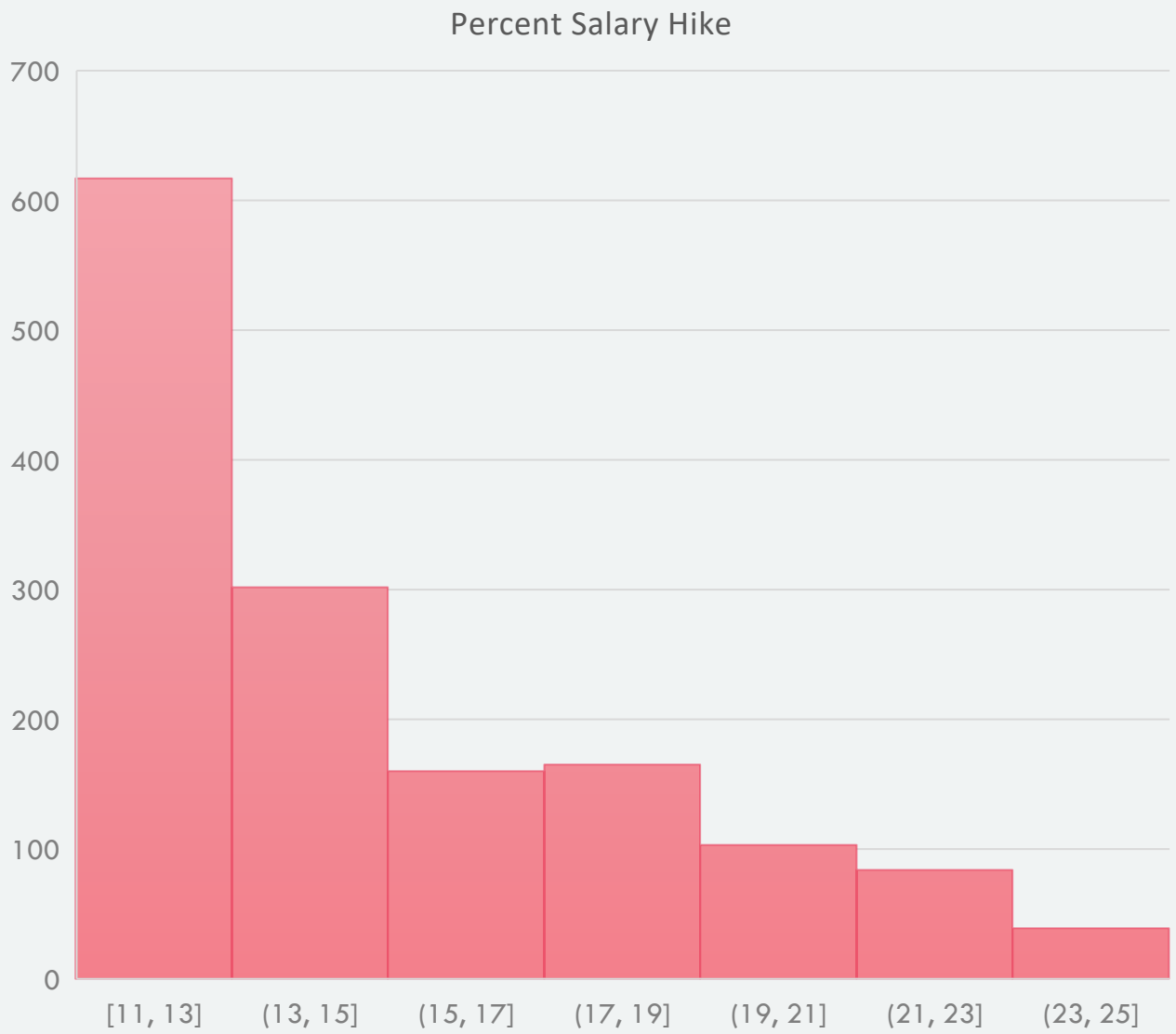
Number of Companies Worked

Nature: Numerical, discrete

Description: For how many companies employees worked, or in other words, how many times they changed their company

Range: From 0 to 9

Insights: Mean of the variable equals 2.7, while median equals 2; standard deviation equals 2.5; distribution is positively skewed, with most of the employees working for just one company in their careers



Percent Salary Hike

Nature: Numerical, discrete

Description: By how many percent grew salaries of employees over previous period

Range: From 11 to 25

Insights: Mean of the variable is 15.2 and median is 14; standard deviation is 3.7; distribution is skewed to the left and there is a steady decrease in frequency as percentage is increased, with the exception of sharp decrease between the first and the second interval

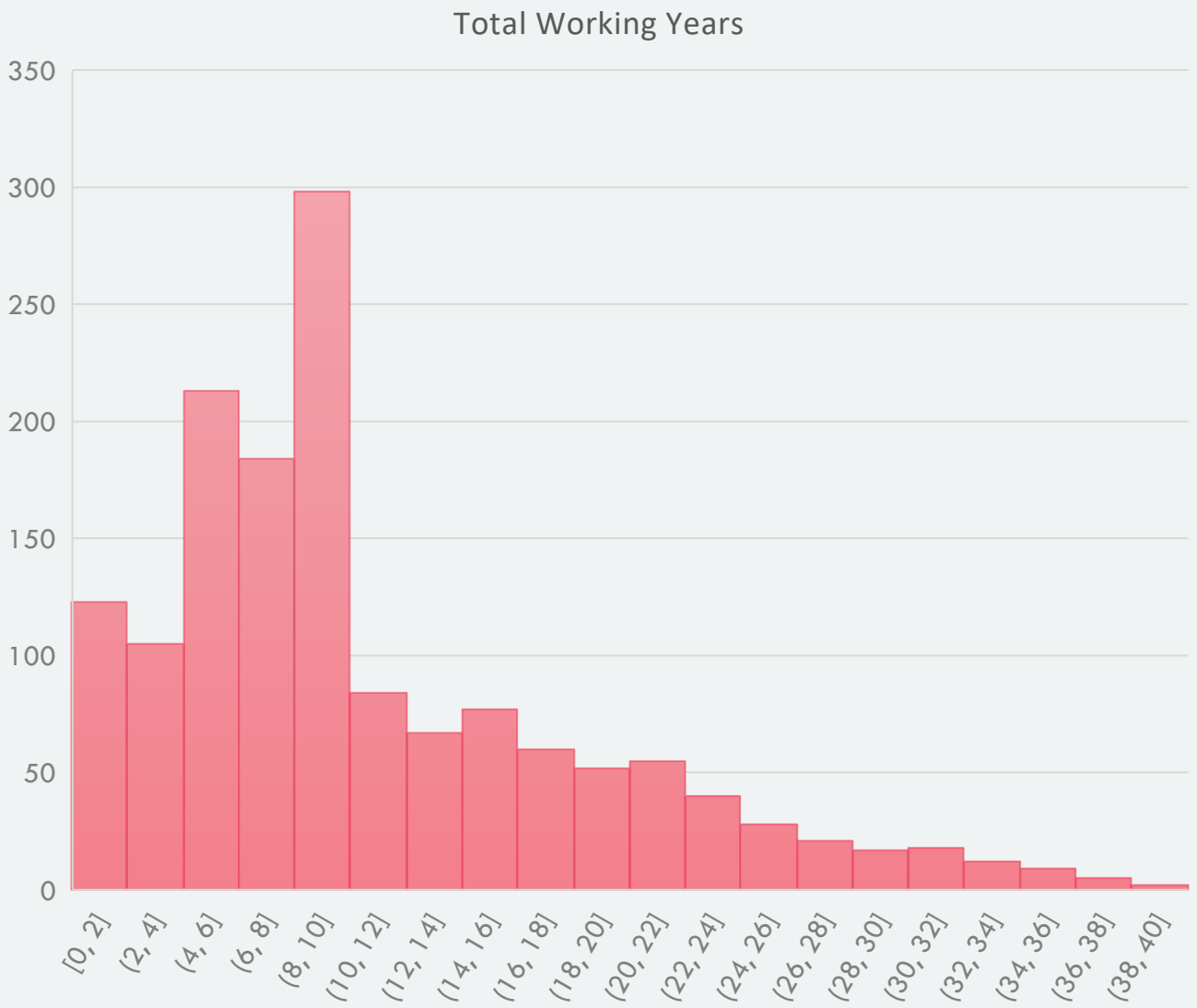
Standard Hours

Nature: Numerical, discrete

Description: Standard hours of employees

Range: 80

Insights: Variable equals 80 for all the employees



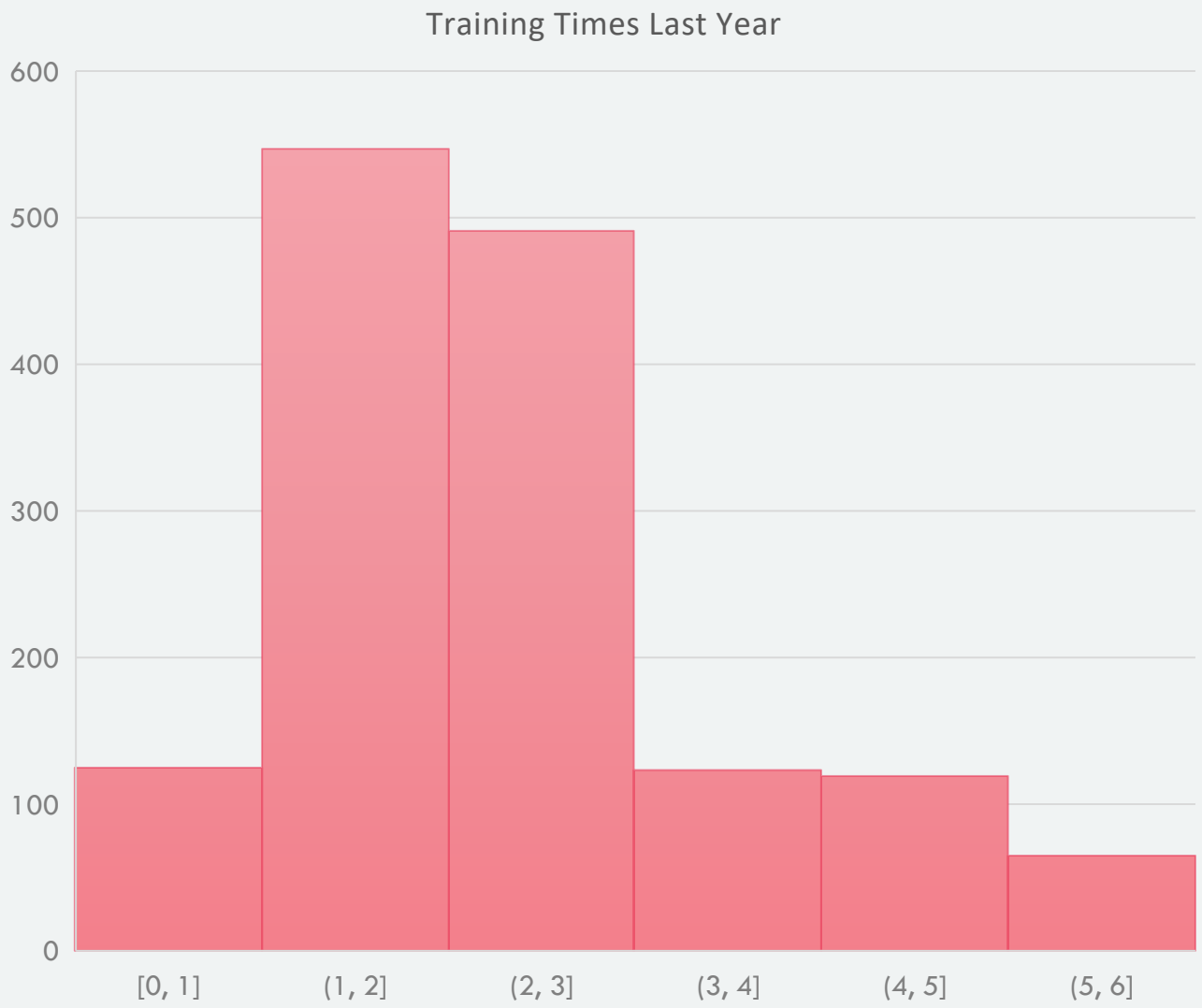
Total Working Years

Nature: Numerical, discrete

Description: For how long do employees work

Range: From 0 to 40

Insights: Mean of this variable is 11.3 and median is 10; standard deviation is 7.8; distribution is positively skewed, with highest frequencies in interval from 4 to 10; after this, there is a steady decrease



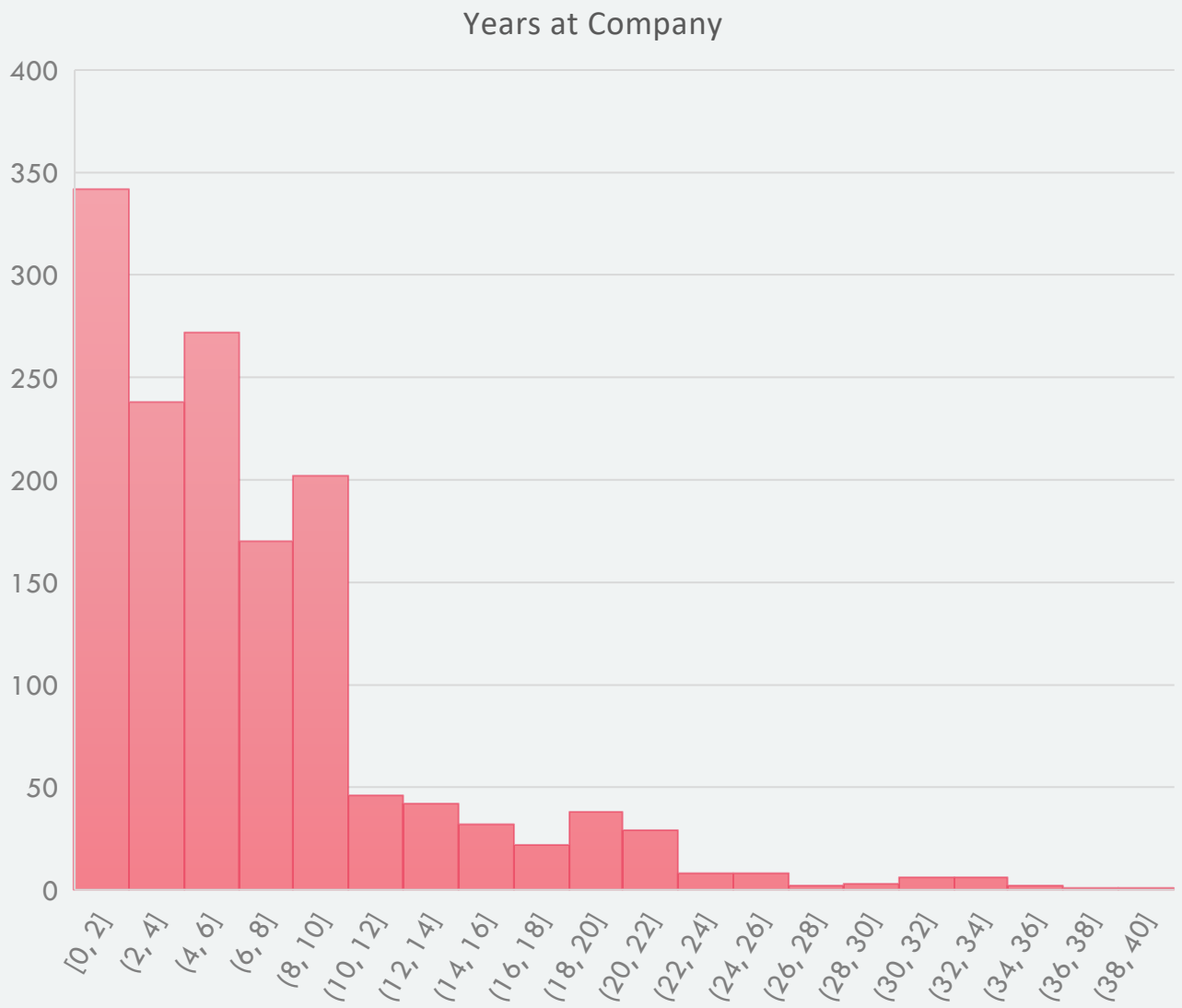
Training Times Last Year

Nature: Numerical, discrete

Description: How many times employees participated in company trainings last year

Range: From 0 to 6

Insights: Mean of this variable equals 2.8, while median equals 3; standard deviation equals 1.3; majority of the employees had 2 or 3 trainings last year, while other intervals have similar frequencies



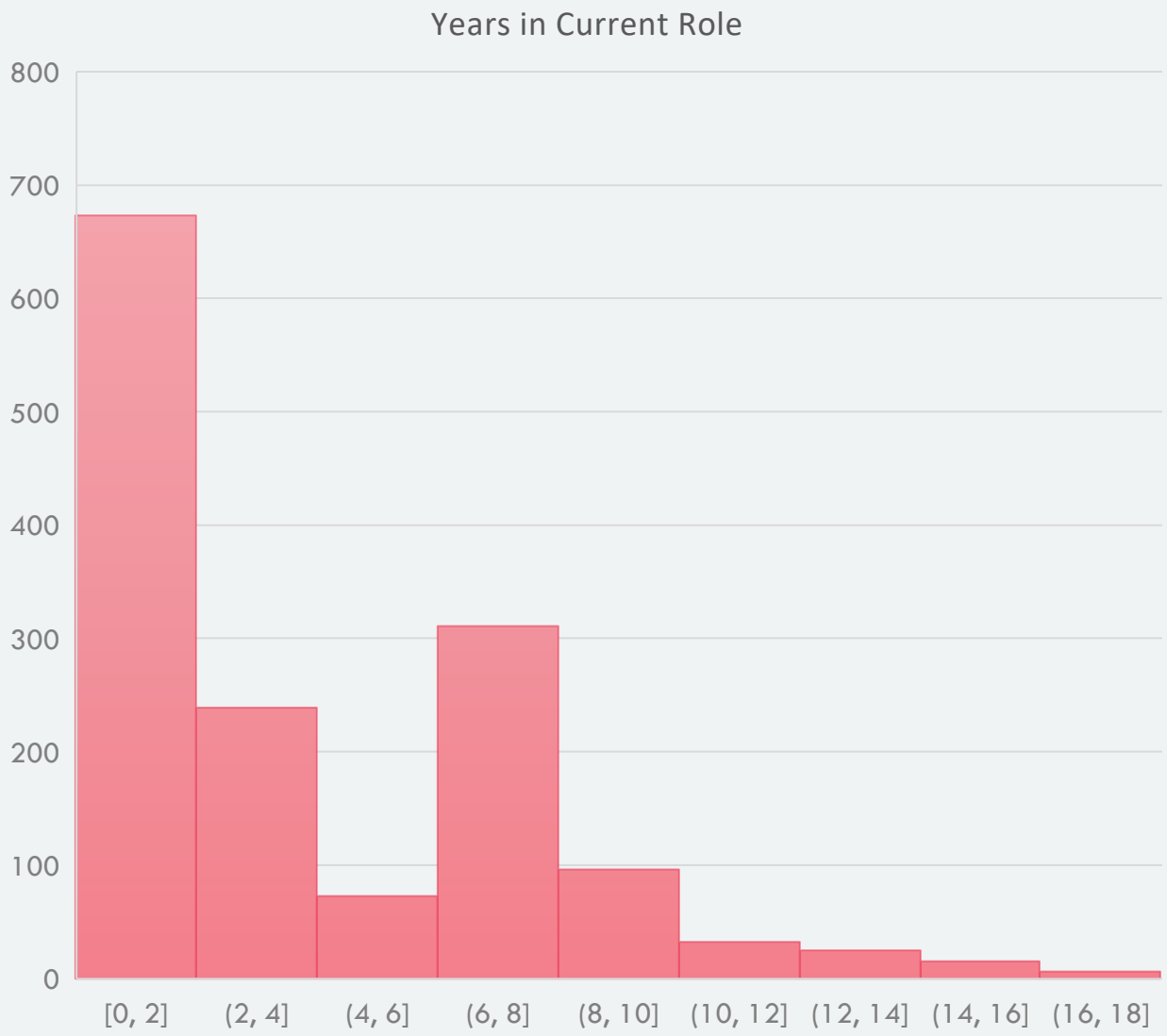
Years At Company

Nature: Numerical, discrete

Description: For how many years do employees work for the company

Range: From 0 to 40

Insights: Mean of this variable is 7, and median is 5; standard deviation is 6.1; distribution is skewed to the left, with a great majority of employees working for the company between 0 and 10 years



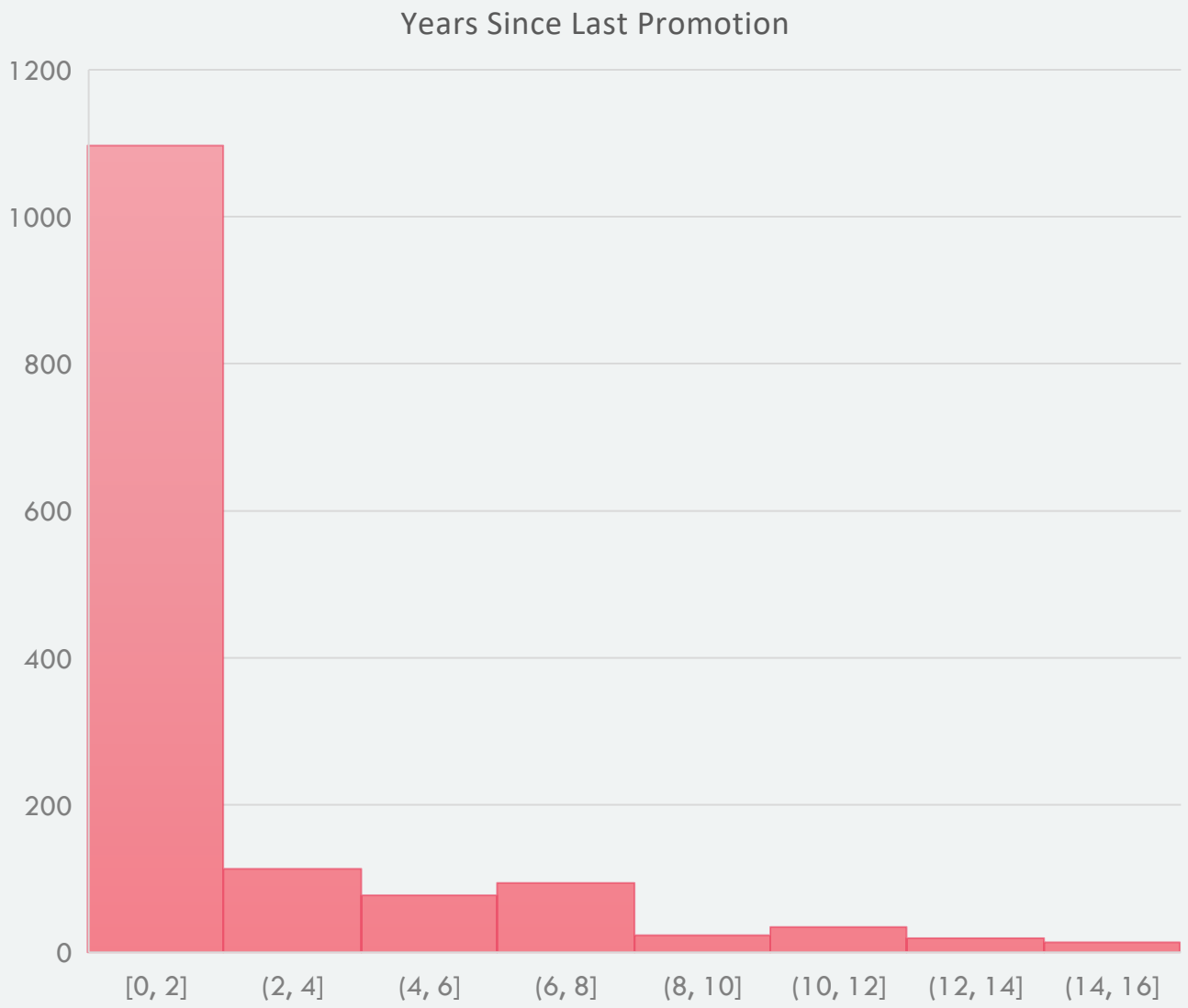
Years In Current Role

Nature: Numerical, discrete

Description: For how many years do employees work at a certain position

Range: From 0 to 18

Insights: Mean of this variable is 4.2, while median is 3; standard deviation is equal to 3.6; distribution is positively skewed, with the general trend of decreasing frequencies as number of years at a position increases



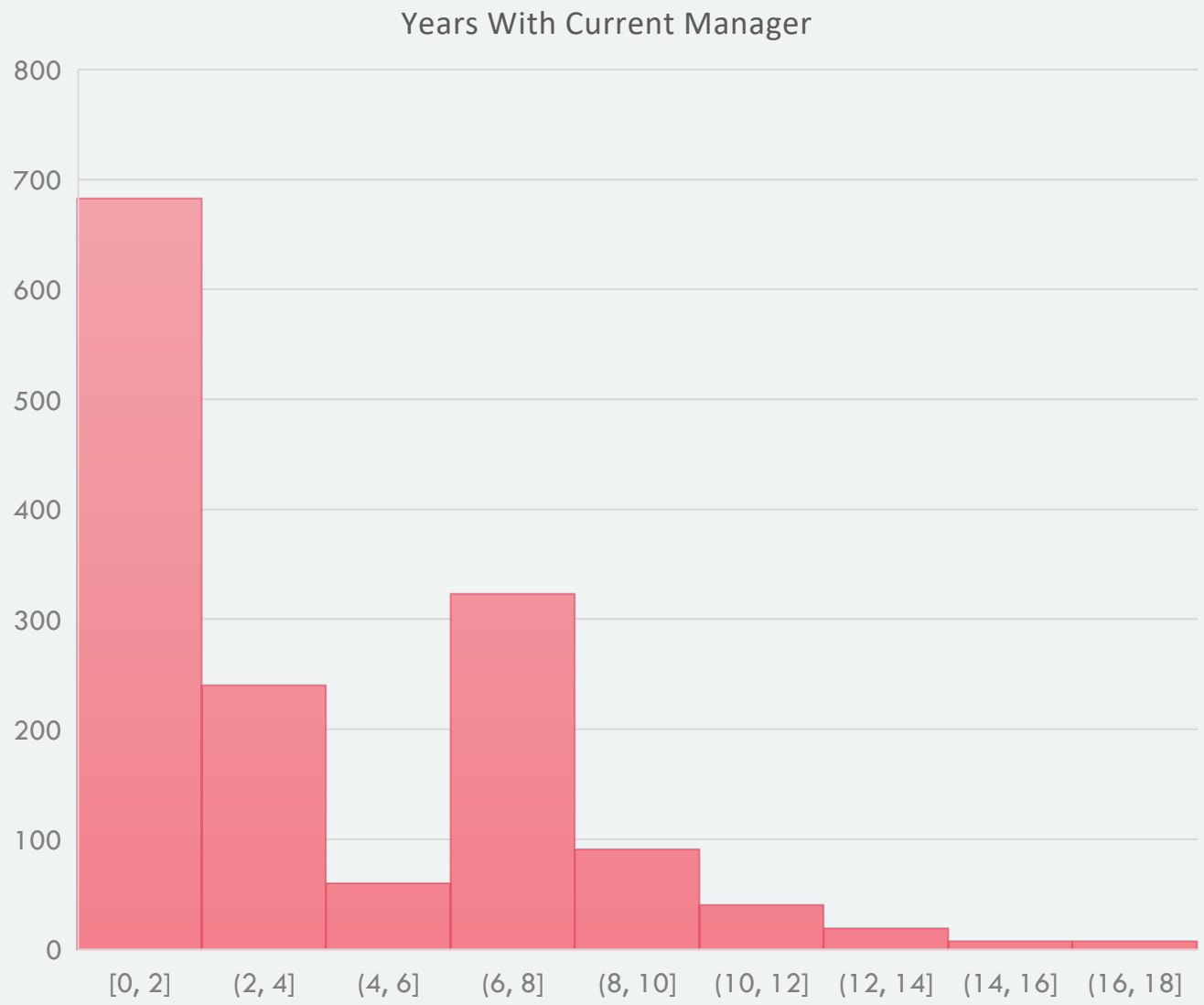
Years Since Last Promotion

Nature: Numerical, discrete

Description: How many years have passed since employees were promoted last time

Range: From 0 to 16

Insights: Mean of the variable equals 2.2, and median equals 1; standard deviation is 3.2; distribution is positively skewed, with most of the employees being promoted recently



Years With Current Manager

Nature: Numerical, discrete

Description: For how many years do employees work with their current managers

Range: From 0 to 18

Insights: Mean of the variable is 4.1, while median is 3; standard deviation equals 3.6; distribution is positively skewed with the highest frequency in interval from 0 to 2

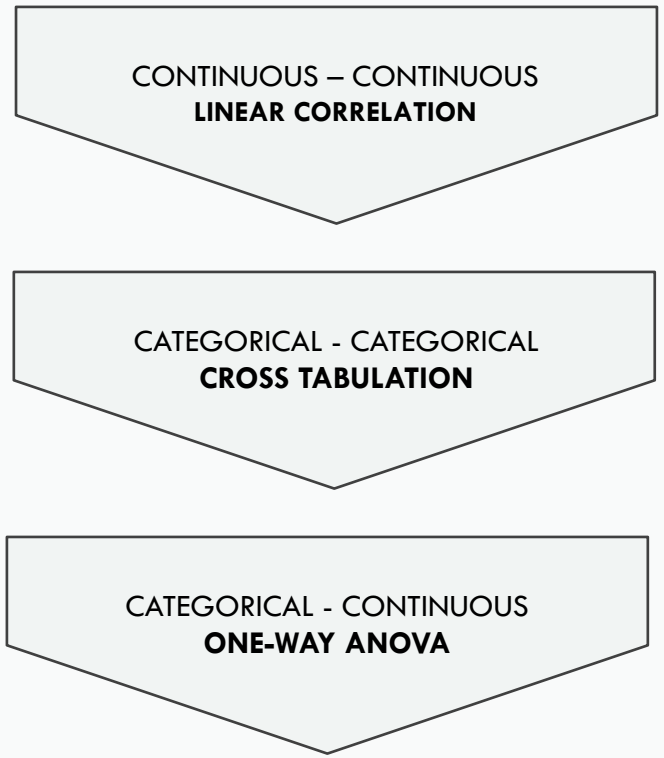
BIVARIATE ANALYSIS

OVERVIEW

Bivariate Analysis is performed to understand interactions between variables themselves.

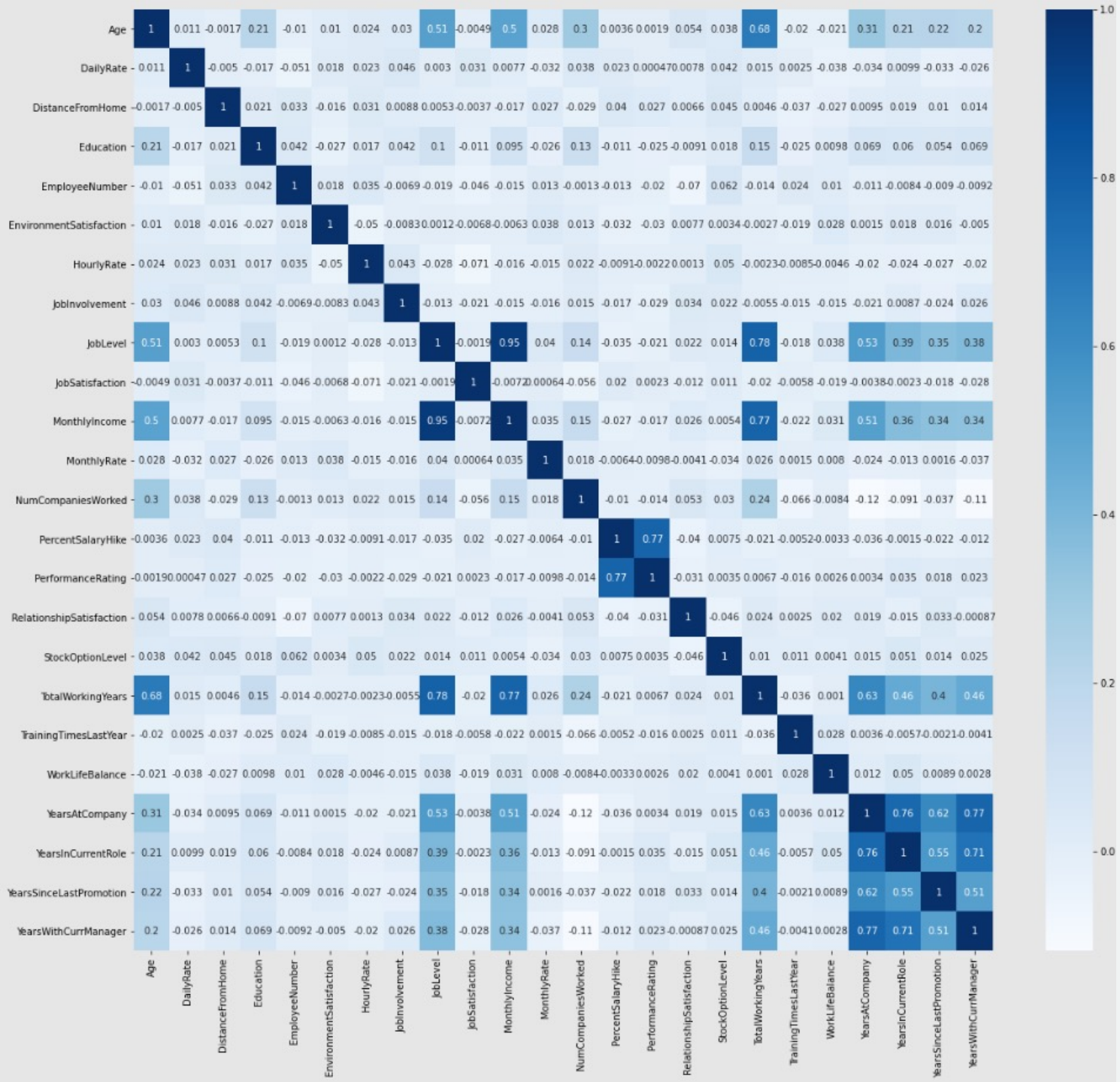
Based on their type, namely continuous or categorical, there are different techniques that can be used to extract insights.

We carry out **continuous – continuous**, **categorical – categorical** and **categorical – continuous** analysis.



CONTINUOUS VARIABLES

In this section, we use a **Linear Correlation Matrix** to understand if there are highly-correlated variables which could lead to multicollinearity issues in our predictive models



CORRELATION MATRIX

The blue square on the bottom right highlights a higher correlation among the variables related to years

However, we have to accept it for many of these variables as they are important in predicting Attrition, as shown in One-Way Anova and Crosstabs

CORRELATION MEASURE

Table "default" – Rows: 345 | Spec – Columns: 5 | Properties | Flow Variables

First column name	Second column name	Correlation value	p value
JobLevel	MonthlyIncome	0.9502999134798449	0.0
Department	JobRole	0.939392649020428	0.0
JobLevel	TotalWorkingYears	0.7822078045362727	0.0
PercentSalaryHike	PerformanceRating	0.7735499964012642	0.0
MonthlyIncome	TotalWorkingYears	0.7728932462543565	0.0
YearsAtCompany	YearsWithCurrManager	0.7692124251007015	0.0
YearsAtCompany	YearsInCurrentRole	0.7587537366134606	0.0
YearsInCurrentRole	YearsWithCurrManager	0.7143647616385879	0.0
TotalWorkingYears	YearsAtCompany	0.6281331552682494	0.0
YearsAtCompany	YearsSinceLastPromotion	0.6184088652176083	0.0
Department	EducationField	0.5904510115591987	0.0
YearsInCurrentRole	YearsSinceLastPromotion	0.5480562476995173	0.0
JobLevel	YearsAtCompany	0.5347386873756313	0.0
MonthlyIncome	YearsAtCompany	0.5142848257331962	0.0
YearsSinceLastPromotion	YearsWithCurrManager	0.5102236357788075	0.0

As shown in the table, we have four highly correlated features – JOB LEVEL and MONTHLY INCOME, as well as DEPARTMENT and JOB ROLE.

Those high correlations represent a problem, and they will be addressed in the following slides.

CATEGORICAL VARIABLES

In this section, we use **Cross Tabulation** to study the absolute joint frequencies of the observed values of two variables. To check for significance, we resort on the p-value of a Chi-Squared test and we use Cramer's V to test the strength of relationship.

Frequency Column Percent	No	Yes	Total
Adult	315	34	349
	25.5474%	14.346%	
Elder	150	23	173
	12.1655%	9.7046%	
Young	235	91	326
	19.0592%	38.3966%	
Young/Adult	533	89	622
	43.2279%	37.5527%	
Total	1,233	237	1,470

AGE BINNED - ATTRITION

Cramer's V

0.126 – Medium

We decided to bin the age to work with years' layers instead of each single age. Thanks to the crosstab we can conclude this sub-division is significant

Frequency Row Percent	No	Yes	Total
1	400	143	543
	73.6648%	26.3352%	
2	482	52	534
	90.2622%	9.7378%	
3	186	32	218
	85.3211%	14.6789%	
4	101	5	106
	95.283%	4.717%	
5	64	5	69
	92.7536%	7.2464%	
Total	1,233	237	1,470

JOB LEVEL - ATTRITION

Cramer's V

0.157 – Medium High

DEPARTMENT - ATTRITION

Cramer's V

0.06 - Low

Frequency Row Percent	No	Yes	Total
Human Resources	51	12	63
	80.9524%	19.0476%	
Research & Development	828	133	961
	86.1602%	13.8398%	
Sales	354	92	446
	79.3722%	20.6278%	
Total	1,233	237	1,470

Frequency Row Percent	No	Yes	Total
Healthcare Representative	122	9	131
	93.1298%	6.8702%	
Human Resources	40	12	52
	76.9231%	23.0769%	
Laboratory Technician	197	62	259
	76.0618%	23.9382%	
Manager	97	5	102
	95.098%	4.902%	
Manufacturing Director	135	10	145
	93.1034%	6.8966%	
Research Director	78	2	80
	97.5%	2.5%	
Research Scientist	245	47	292
	83.9041%	16.0959%	
Sales Executive	269	57	326
	82.5153%	17.4847%	
Sales Representative	50	33	83
	60.241%	39.759%	
Total	1,233	237	1,470

JOB ROLE - ATTRITION

Cramer's V

0.171 – Medium High

YEARS WITH CURRENT MANAGER - ATTRITION

Cramer's V

0.128 - Medium

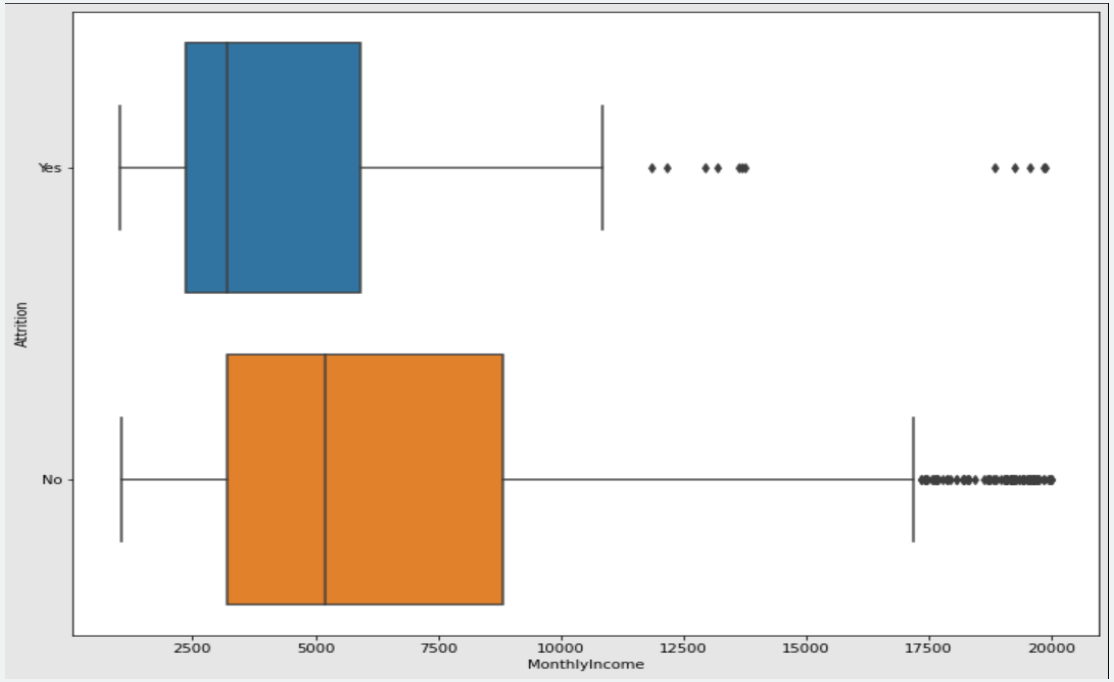
Frequency Row Percent	No	Yes	Total
Less 2	243	96	339
	71.6814%	28.3186%	
More 2	990	141	1,131
	87.5332%	12.4668%	
Total	1,233	237	1,470

CONTINUOUS – CATEGORICAL VARIABLES

Since we deal with different data types, we use the **One-Way ANOVA** which resorts on t-tests to evaluate whether the difference in means of the variable's categories is statistically significant. If so, it would mean that the distinction is needed to have a more informative view of the other variable taken into consideration, Attrition in our case.

For visualization purposes, we use box-plots.

MONTHLY INCOME – ATTRITION



One-way analysis of variance (ANOVA)

Descriptive Statistics

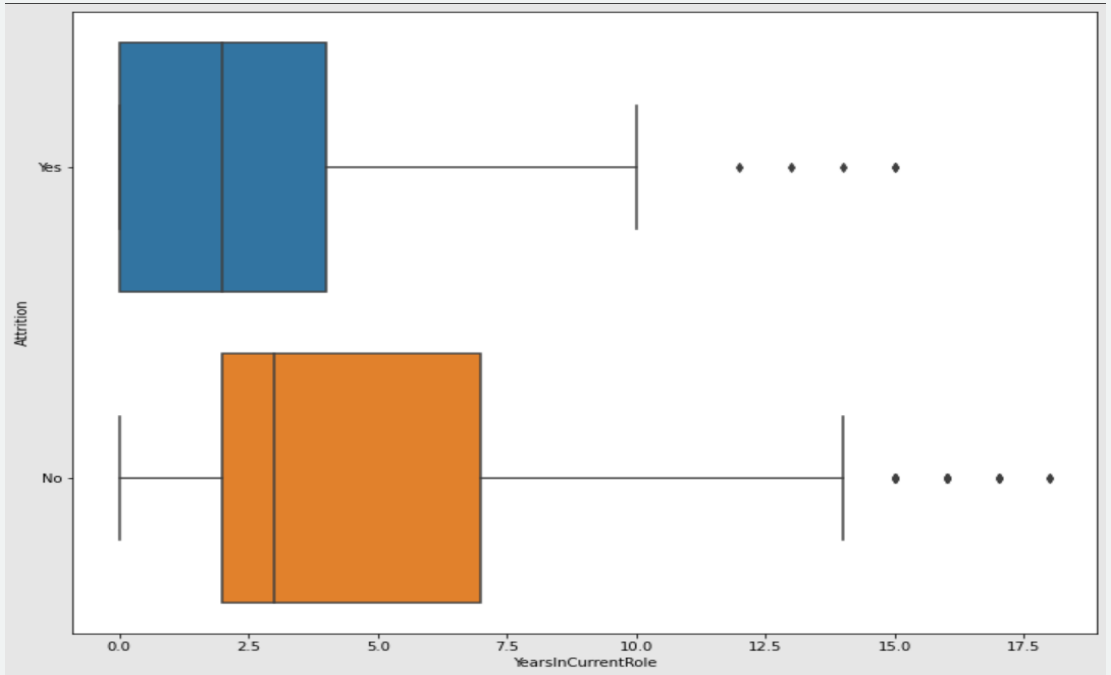
Confidence Interval (CI) Probability: 95.0%

	Group	N	Missing	Missing Group	Mean	Std. Deviation
MonthlyIncome	Yes	237	0	0	4,787.0928	3,640.2104
MonthlyIncome	No	1233	0	0	6,832.7397	4,818.208
MonthlyIncome	Total	1470	0	0	6,502.9313	4,707.9568

ANOVA

	Source	Sum of Squares	df	Mean Square	F	p-value
MonthlyIncome	Between Groups	8.32E8	1	8.32E8	38.4888	7.15E-10
MonthlyIncome	Within Groups	3.17E10	1468	21,613,286.8879		
MonthlyIncome	Total	3.26E10	1469			

YEARS IN CURRENT ROLE – ATTRITION



One-way analysis of variance (ANOVA)

Descriptive Statistics

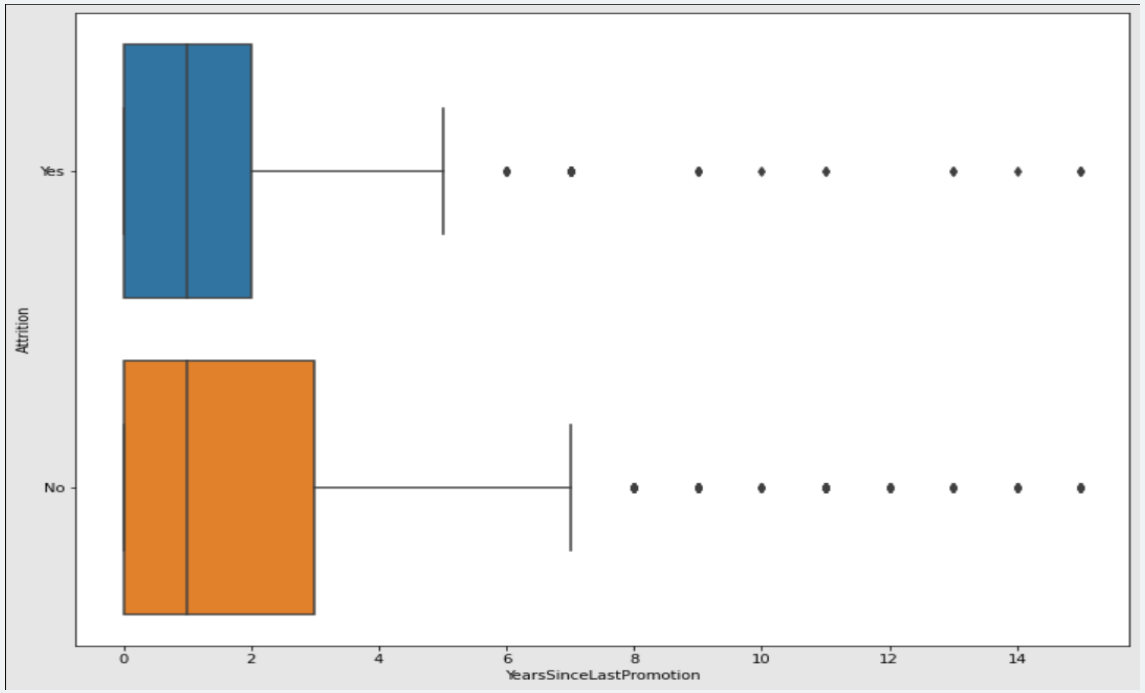
Confidence Interval (CI) Probability: 95.0%

	Group	N	Missing	Missing Group	Mean	Std. Deviation
YearsInCurrentRole	Yes	237	0	0	2.903	3.1748
YearsInCurrentRole	No	1233	0	0	4.4842	3.6494
YearsInCurrentRole	Total	1470	0	0	4.2293	3.6231

ANOVA

	Source	Sum of Squares	df	Mean Square	F	p-value
YearsInCurrentRole	Between Groups	497.0326	1	497.0326	38.8383	6.00E-10
YearsInCurrentRole	Within Groups	18,786.7095	1468	12.7975		
YearsInCurrentRole	Total	19,283.7422	1469			

YEARS SINCE LAST PROMOTION – ATTRITION



One-way analysis of variance (ANOVA)

Descriptive Statistics

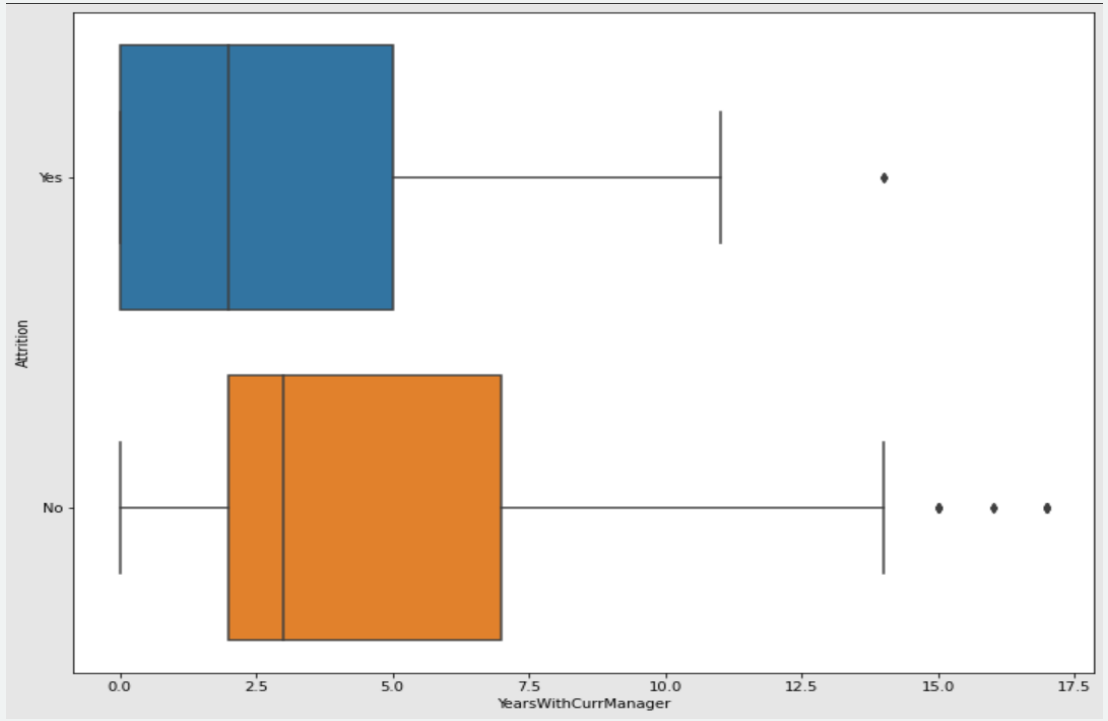
Confidence Interval (CI) Probability: 95.0%

	Group	N	Missing	Missing Group	Mean	Std. Deviation
YearsSinceLastPromotion	Yes	237	0	0	1.9451	3.1531
YearsSinceLastPromotion	No	1233	0	0	2.2344	3.2348
YearsSinceLastPromotion	Total	1470	0	0	2.1878	3.2224

ANOVA

	Source	Sum of Squares	df	Mean Square	F	p-value
YearsSinceLastPromotion	Between Groups	16.6307	1	16.6307	1.6022	0.2058
YearsSinceLastPromotion	Within Groups	15,237.5489	1468	10.3798		
YearsSinceLastPromotion	Total	15,254.1796	1469			

YEARS WITH CURRENT MANAGER – ATTRITION



One-way analysis of variance (ANOVA)

Descriptive Statistics

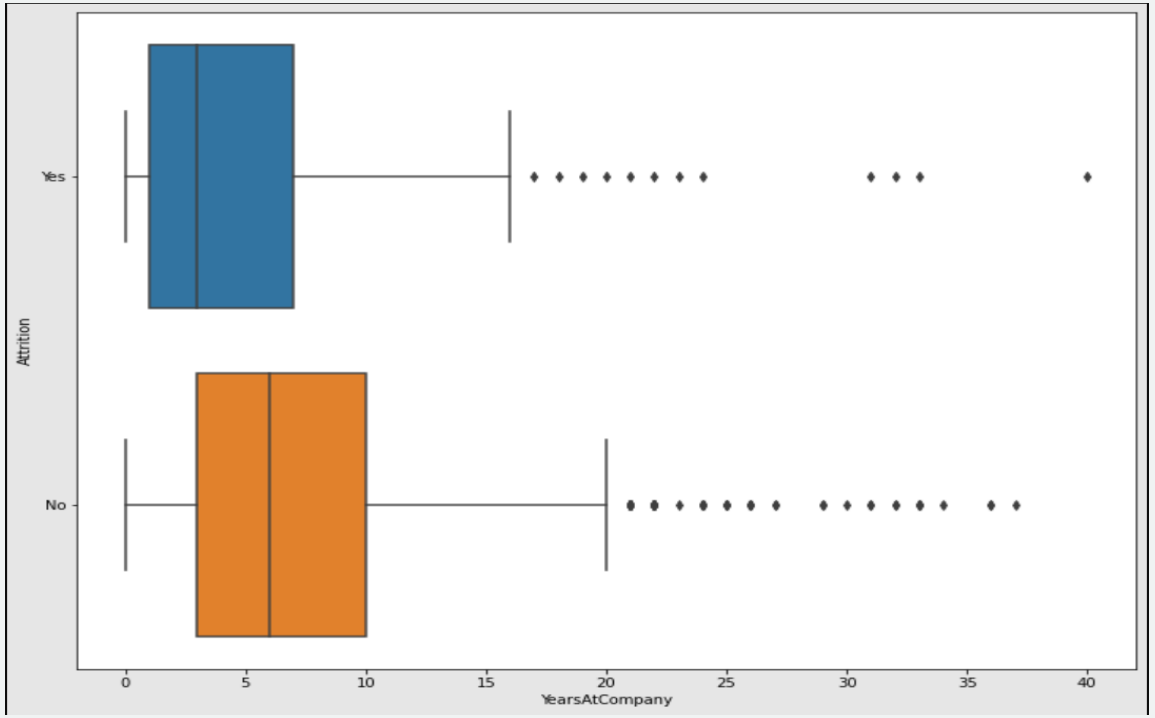
Confidence Interval (CI) Probability: 95.0%

	Group	N	Missing	Missing Group	Mean	Std. Deviation
YearsWithCurrManager	Yes	237	0	0	2,8523	3,1433
YearsWithCurrManager	No	1233	0	0	4,3674	3,5941
YearsWithCurrManager	Total	1470	0	0	4,1231	3,5681

ANOVA

	Source	Sum of Squares	df	Mean Square	F	p-value
YearsWithCurrManager	Between Groups	456,313	1	456,313	36,7123	1,74E-9
YearsWithCurrManager	Within Groups	18.246,4006	1468	12,4294		
YearsWithCurrManager	Total	18.702,7136	1469			

YEARS AT COMPANY – ATTRITION



One-way analysis of variance (ANOVA)

Descriptive Statistics

Confidence Interval (CI) Probability: 95.0%

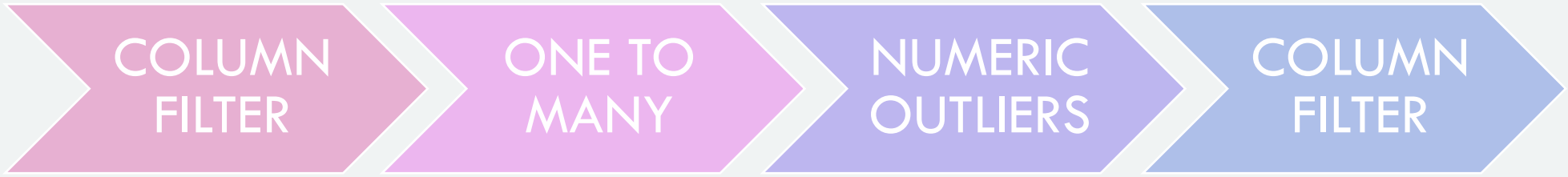
	Group	N	Missing	Missing Group	Mean	Std. Deviation
YearsAtCompany	Yes	237	0	0	5,1308	5,95
YearsAtCompany	No	1233	0	0	7,369	6,0963
YearsAtCompany	Total	1470	0	0	7,0082	6,1265

ANOVA

	Source	Sum of Squares	df	Mean Square	F	p-value
YearsAtCompany	Between Groups	995,8604	1	995,8604	27,0016	2,32E-7
YearsAtCompany	Within Groups	54.142,0417	1468	36,8815		
YearsAtCompany	Total	55.137,902	1469			

DATA PREPARATION

OVERVIEW



INITIAL STEPS: MISSING VALUES

Before looking for the columns that can be filtered out of the dataset, we checked if there are any **missing values**:

Variable	Member count
Monthly Income	1470
Num Companies Worked	1470
Total Working Years	1470
Training Times Last Year	1470
Years At Company	1470
Years In Current Role	1470
Years Since Last Promotion	1470
Years With Current Manager	1470

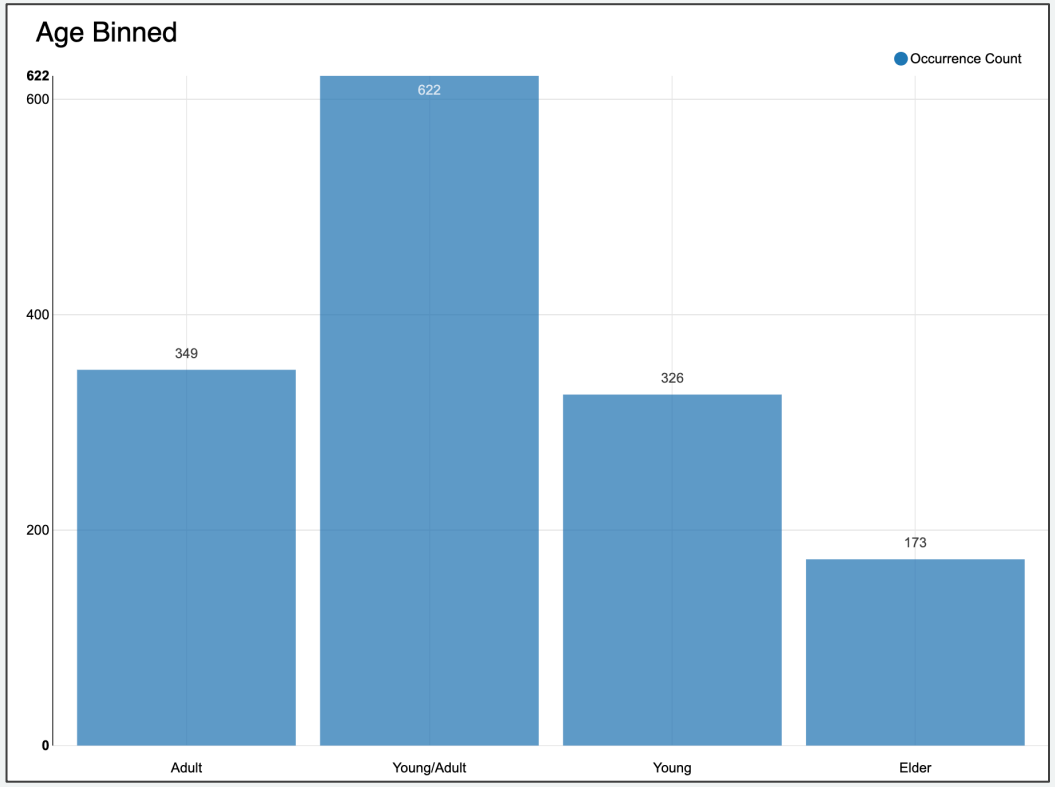
Using the **Statistics** node we conclude that there are no missing values in our dataset.

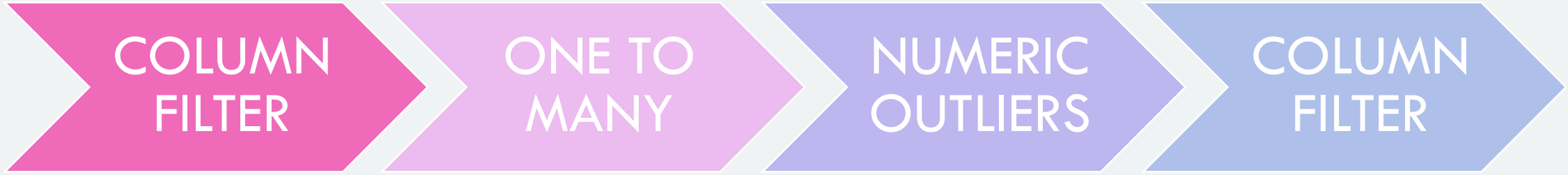
INITIAL STEPS: AGE BINNING

Another thing to do before configuring the Column Filter was **feature engineering:**

Using the **Numeric Binner** node we transformed numerical variable AGE into categorical, using the following schema:

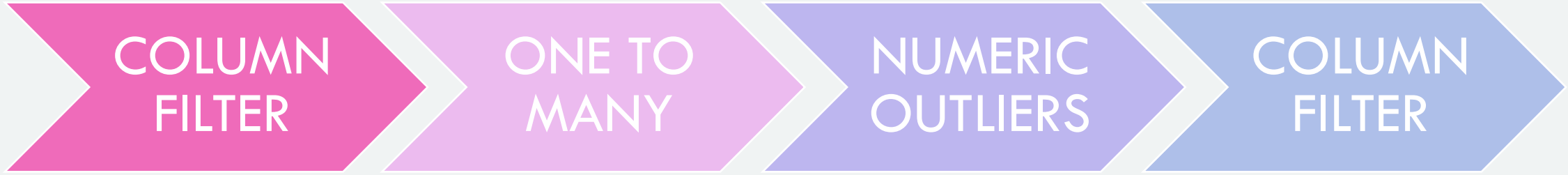
- age < 30: YOUNG
- 30 ≤ age < 40: YOUNG/ADULT
- 40 ≤ age < 50: ADULT
- age ≥ 50: ELDER



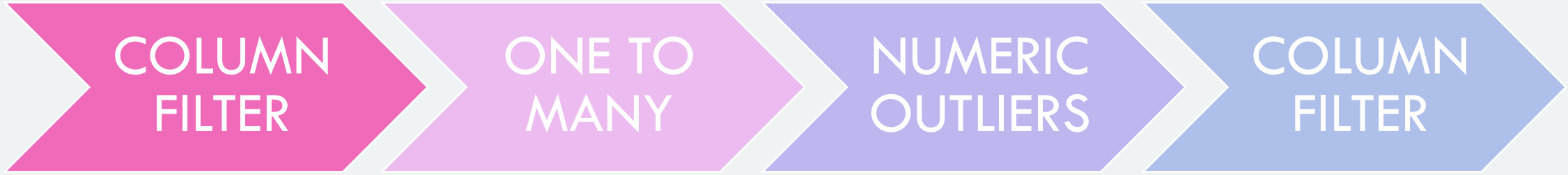


Then, the following features were removed:

- **EMPLOYEE NUMBER**: It is removed because it is simply the Employee ID in the dataset
- **OVER 18**: Since minimum working age starts at 18, OVER18 just represents a column of 1s
- **EMPLOYEE COUNT**: As for OVER18, it is just a column of 1s since all datapoints are in fact employees, and the EMPLOYEE COUNT is a binary variable that assigns either 0 (unemployed) or 1 (employed)
- **STANDARD HOURS**: Redundant since it is a constant value of 80 for all employees



- **JOB LEVEL**: as shown in Bivariate Analysis, we found out that Cramer's V for **JOB LEVEL** and **ATTRITION** is 0.157, which can be considered relatively high, while the Pearson correlation coefficient between **JOB LEVEL** and **MONTHLY INCOME** equals to 0.95. Since one-way ANOVA showed that people who do not quit have a higher monthly income compared to those that do quit the company (6833 versus 4787), we had to discard either JOB LEVEL or MONTHLY INCOME. Deciding which one to keep and which one to discard, we discarded JOB LEVEL since it is not fully explained by dataset producers and it is more straightforward to use MONTHLY INCOME.

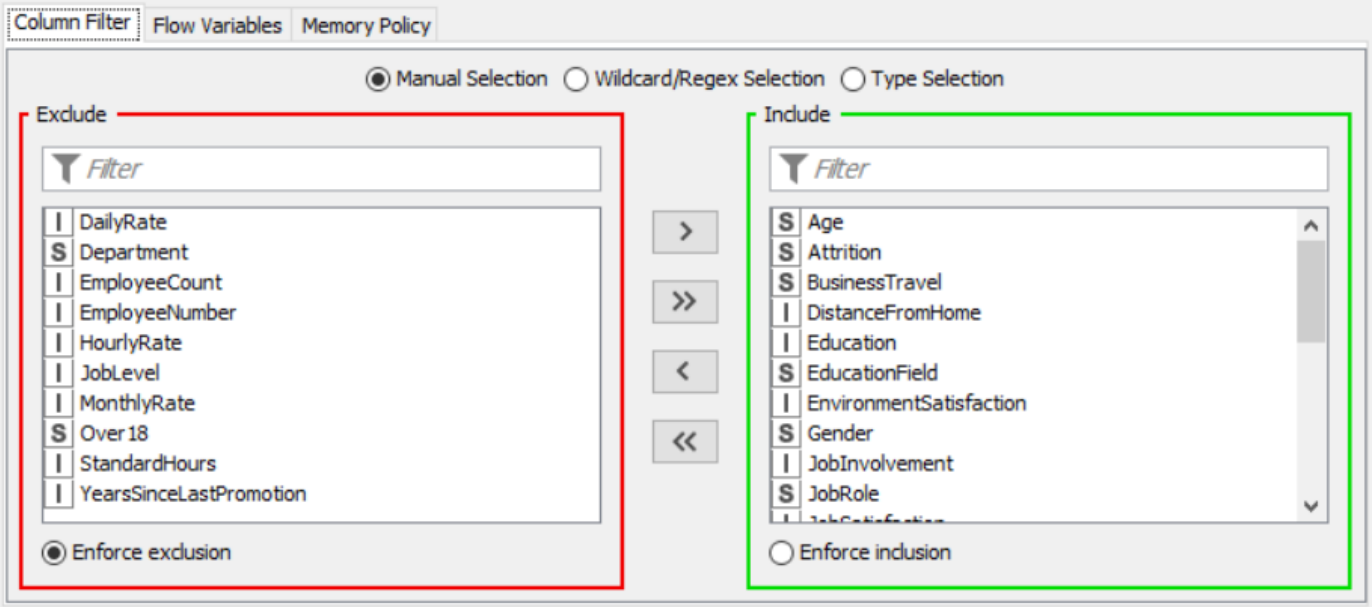


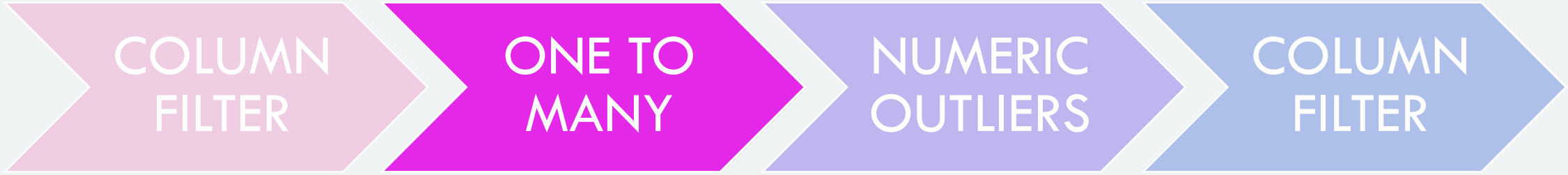
- **DEPARTMENT**: by performing Cross tabulation between **ATTRITION** and **DEPARTMENT** we obtained Cramer's V value of just 0.06, meaning that employee's department is not helpful in explaining whether he or she will leave the company.

- **YEARS SINCE LAST PROMOTION**: One-way ANOVA between **ATTRITION** and **YEAR SINCE LAST PROMOTION** showed us that the difference in means between leaving the company or not is not statistically significant (p-value of 0.21 was obtained). Therefore, we decided to remove it.

COLUMN FILTER → ONE TO MANY → NUMERIC OUTLIERS → COLUMN FILTER

Lastly, **DAILY RATE, HOURLY RATE** and **MONTHLY RATE** were filtered out, as we decided to proceed only with MONTHLY INCOME



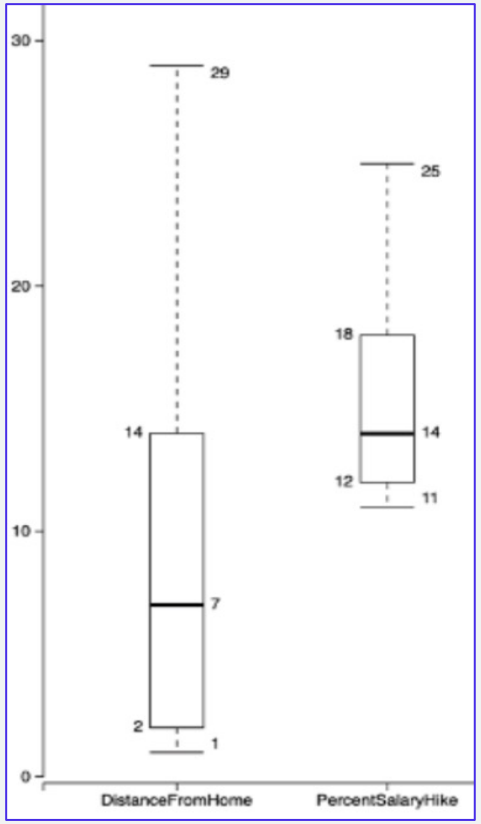


The next step in the data preparation process was implementation of the **One Hot Encoding** technique in order to transform the categorical features into a set of dummy variables. This technique was chosen since it proved to be the most helpful for our prediction.

One to Many node was used, and the features chosen for encoding were **AGE, BUSINESS TRAVEL, EDUCATION FIELD, GENDER, JOB ROLE, MARITAL STATUS** and **OVERTIME.** Twenty nine resulting columns were concatenated with the dataset.

COLUMN FILTER → ONE TO MANY → NUMERIC OUTLIERS → COLUMN FILTER

Using the **Box Plot** node on numerical features we determined that **DISTANCE FROM HOME** and **PERCENT SALARY HIKE** do not have outliers



COLUMN FILTER → ONE TO MANY → NUMERIC OUTLIERS → COLUMN FILTER

On the other hand, for all the other features, the corresponding box plot graphs clearly display the presence of the outliers

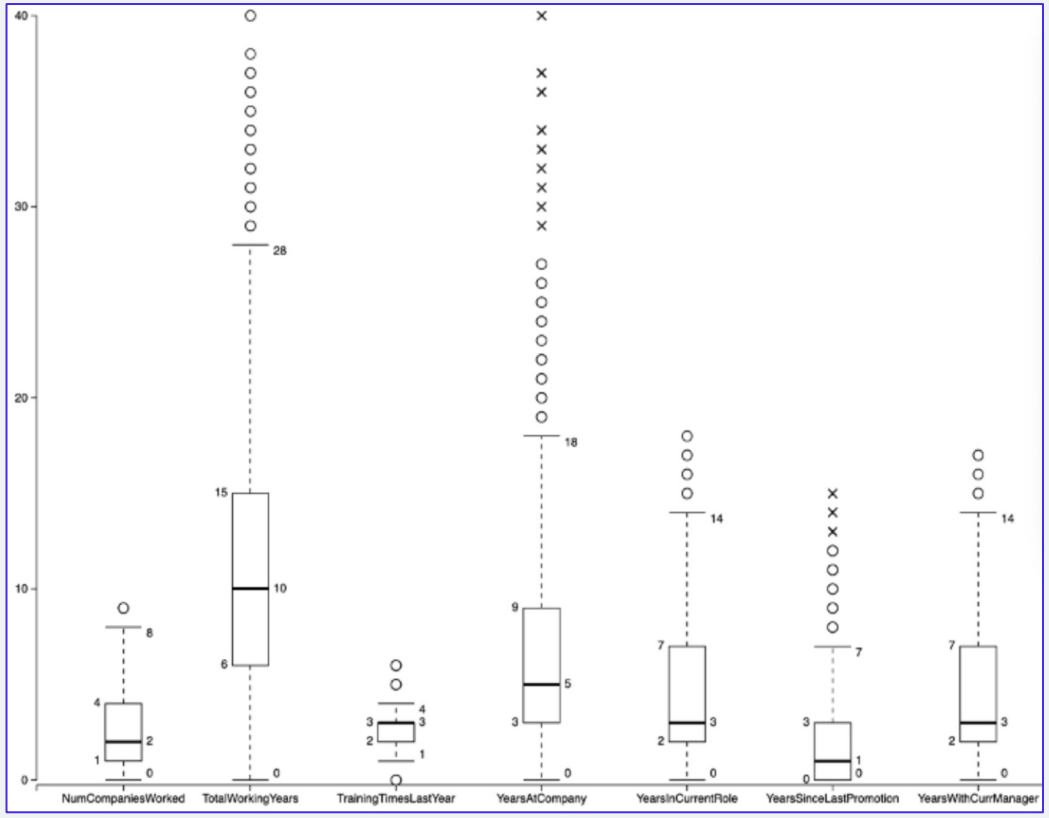
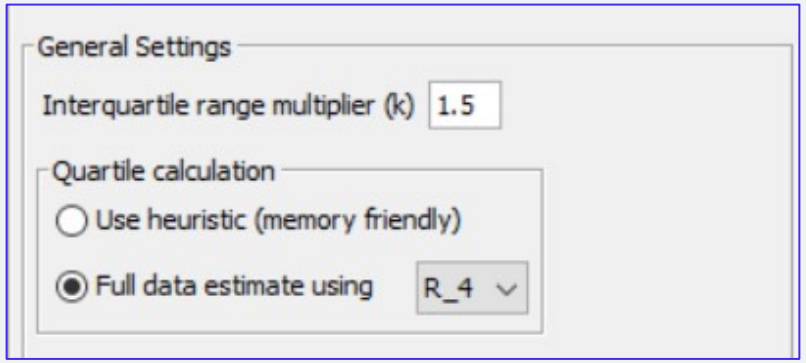


Table "default" – Rows: 8 | Spec – Columns: 5 | Properties | Flow Variables

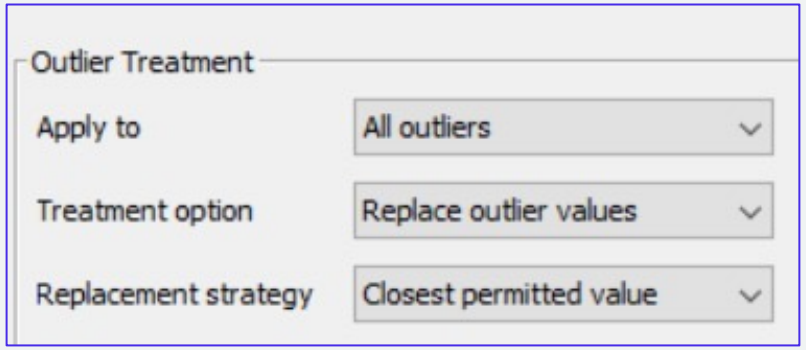
Row ID	Outlier column	Memb...	Outlier count	Lower bound	Upper bound
Row0	MonthlyIncome	1470	114	-5,292	16,580
Row1	NumCompaniesWorked	1470	52	-3.5	8.5
Row2	TotalWorkingYears	1470	63	-7.5	28.5
Row3	TrainingTimesLastYear	1470	238	0.5	4.5
Row4	YearsAtCompany	1470	104	-6	18
Row5	YearsInCurrentRole	1470	21	-5.5	14.5
Row6	YearsSinceLastPromotion	1470	107	-4.5	7.5
Row7	YearsWithCurrManager	1470	14	-5.5	14.5

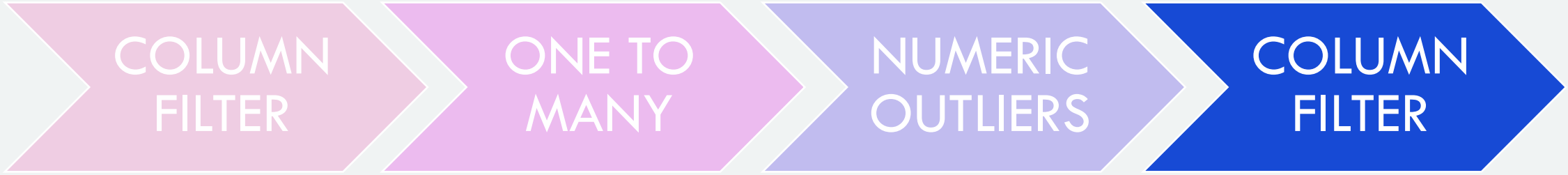


The table above shows us the number of outliers for each of the numerical features

As it can be seen from the configuration of the **Numeric Outliers** node, the interquartile range multiplier was set to 1.5

Since the dataset does not consist of a huge number of observations, instead of removing outliers we replace them with their closest permitted values





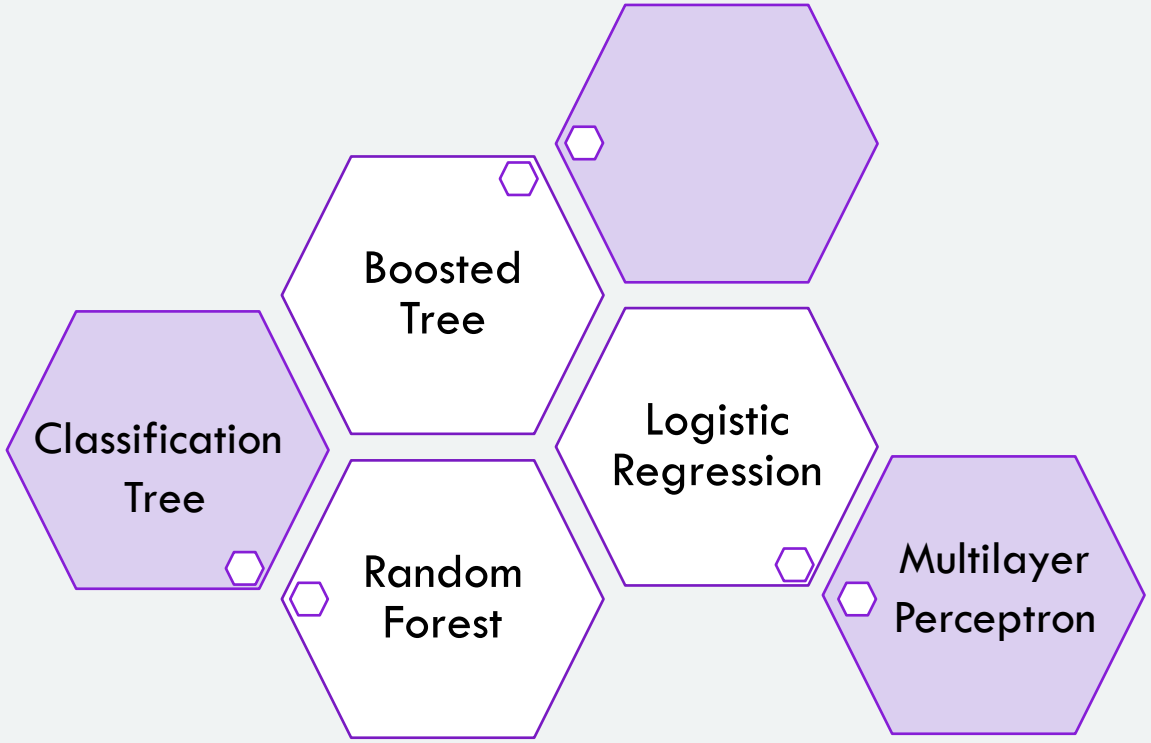
As the last part of the data preparation process, the following columns were removed:

- **AGE, BUSINESS TRAVEL, EDUCATION FIELD, GENDER, JOB ROLE, MARITAL STATUS** and **OVERTIME,** since they were replaced with encoded versions which contain binary values (named with the respective categorical value suffix, e.g. MARRIED MARITAL)

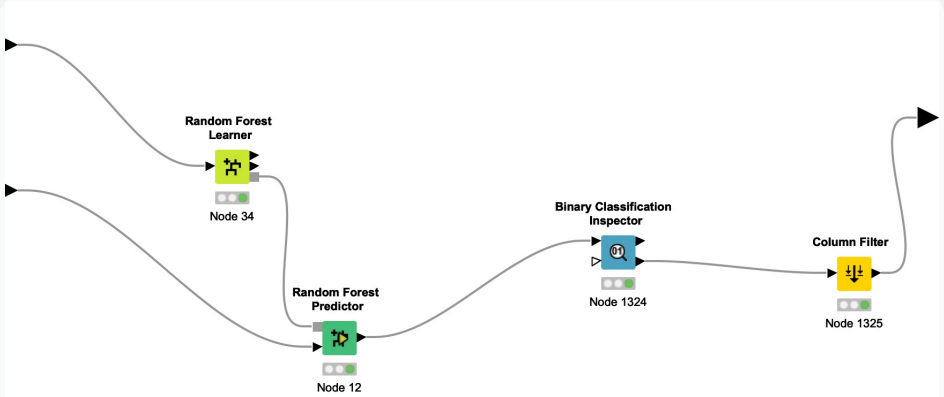
- **GENDER FEMALE** and **OVERTIME NO**, to avoid the problem of multicollinearity (dummy variable trap)

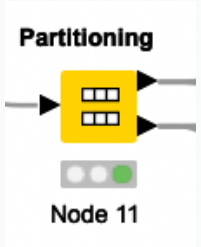
DATA ANALYSIS

Implemented Models



Focus on Models





Random Forest

The first model implemented is the Random Forest, namely an ensamble of several decision trees. A train-test split is performed prior to this metanode using the Partitioning node on the filtered dataset:

- 75% training set
- 25% test set
- Stratified Sampling option

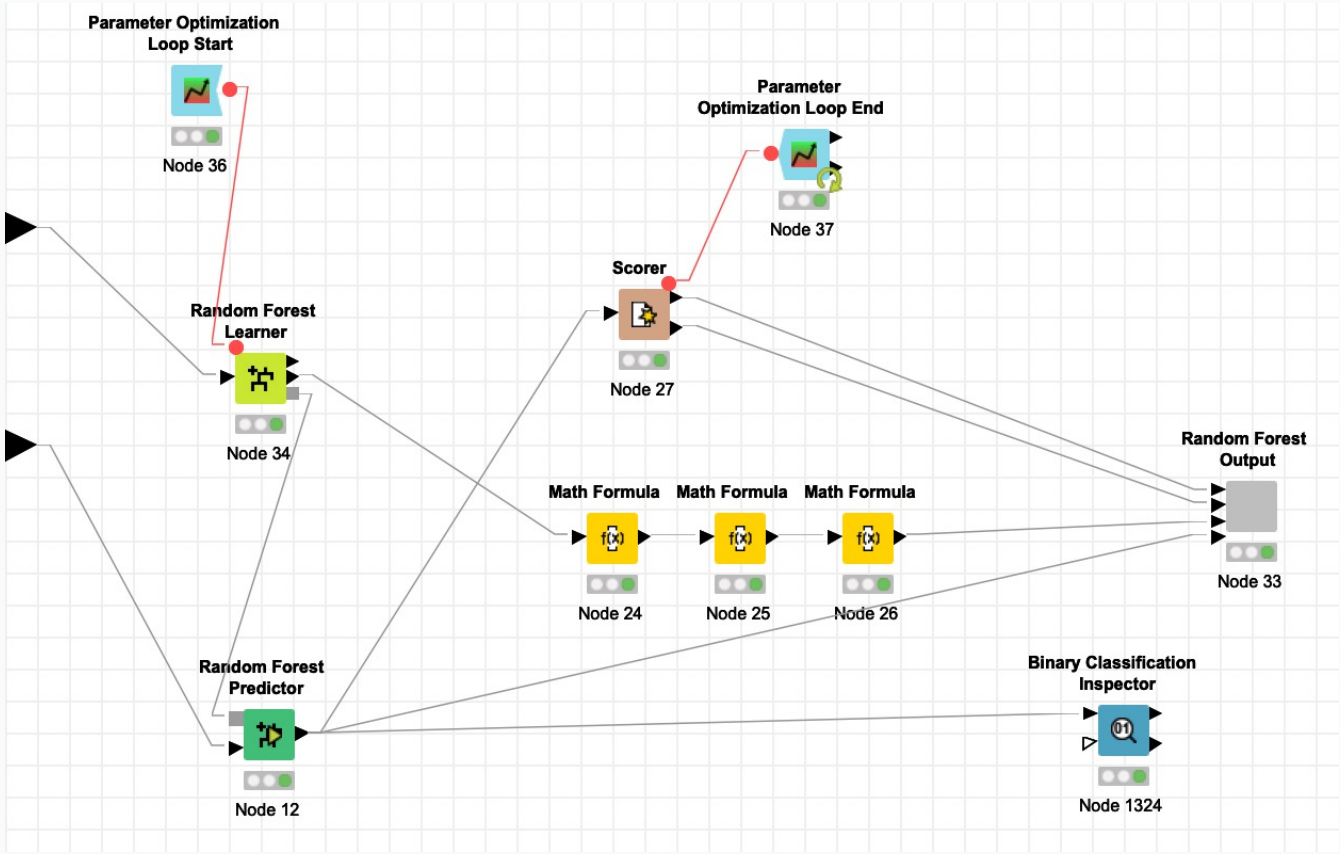
The first step consists of the Random Forest Learner node, where we specified the split criterion, the Information Gain Ratio, but also limited the number of levels (tree depth) to its best value. After that, using a Binary Classification Inspector, we further tried to balance the confusion matrix by maximizing Youden's Index. In the end, with **Column Filt**er we decided to manually include some metrics such as Accuracy, Sensitivity, Specificity and AUC in order to compare the resulting performances later on, together with the ones of the other models.

Random Forest Optimized

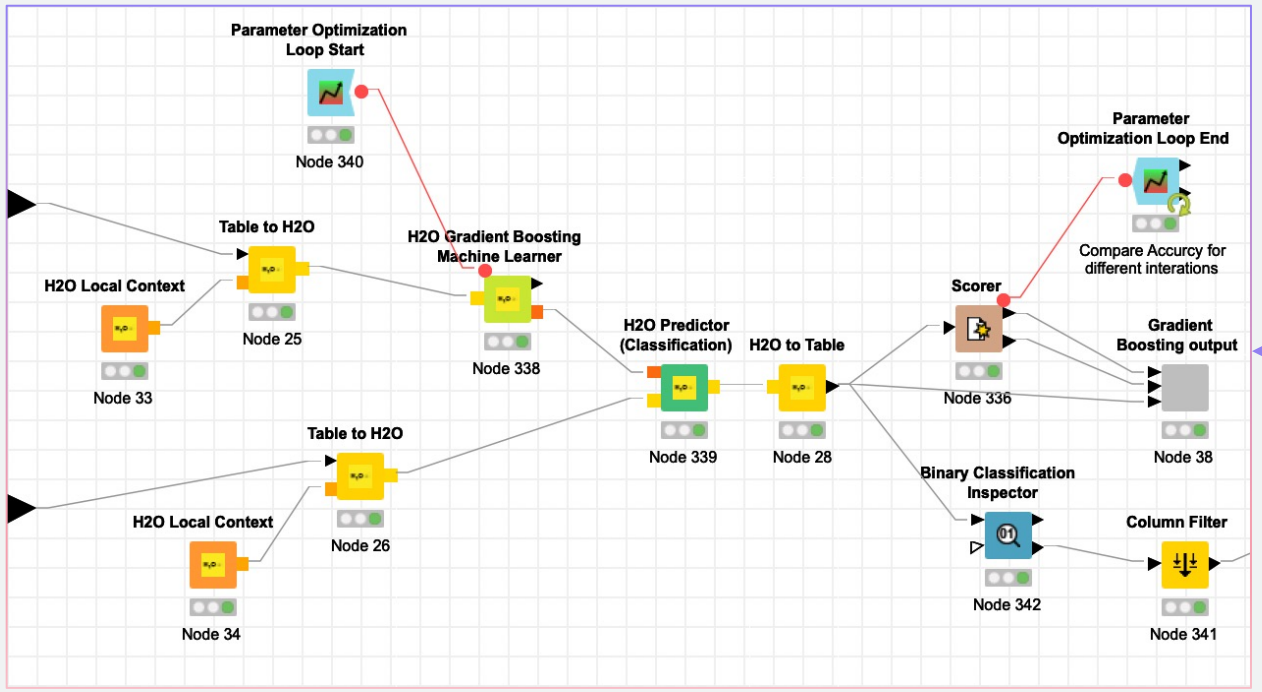
One of the most important steps in ML models definition is to fine tune their hyperparameters.

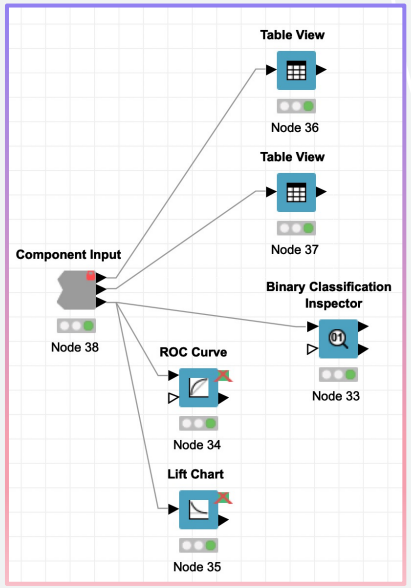
As we all know, we need Cross – Validation loops to train our model with different sets of pre-defined parameters and to test their performances. Then we store the best results to be used as a final hyperparameters for our prediction. We usually adopted a Bayesian or Randomized approach to this task in order to avoid long training times coming from the ”Brute force” method.

In this case we adopted the Random Search to search for node size, number of models and the depth of each tree in the ensamble.



Gradient Boosting





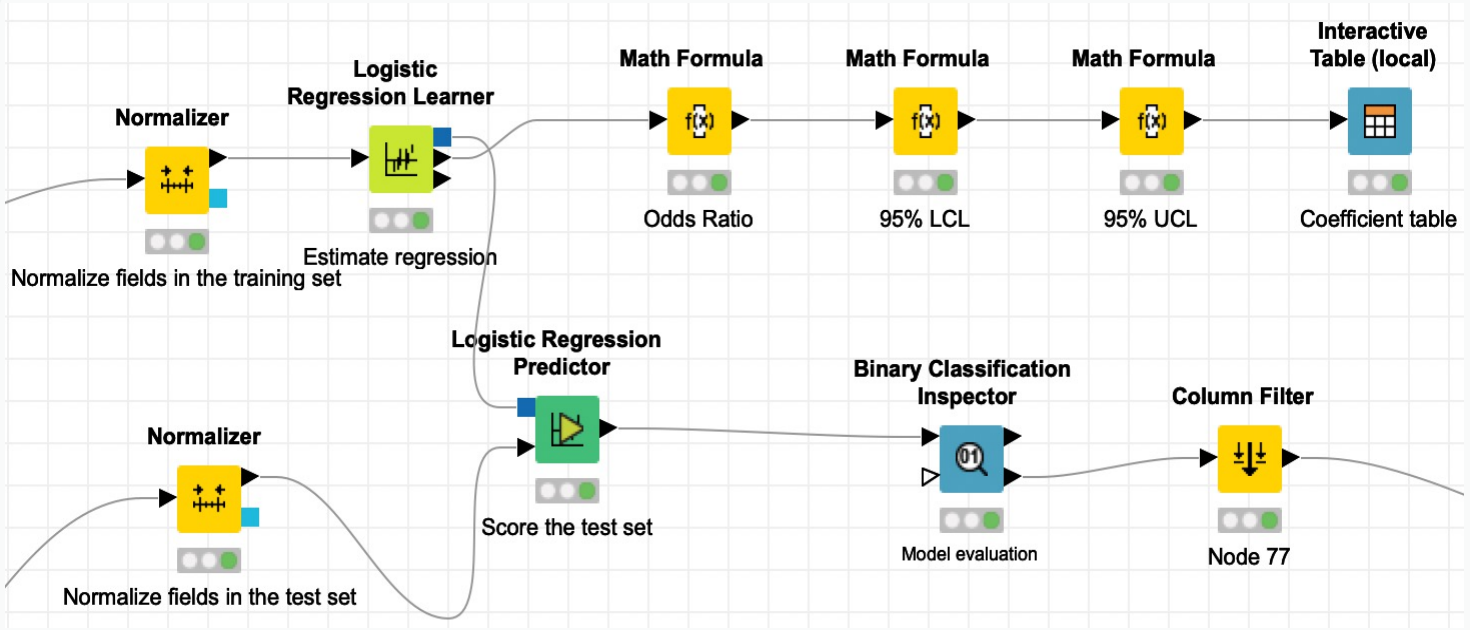
As a second model, we proposed the Gradient Boosting algorithm. The H2O distributed version has been used since it allows for a better parameters optimization. As search strategy we selected the Bayesian Optimization (TPE) which consists of two phases: the first one is a warm-up during which parameters combinations are chosen at random and subsequently evaluated, while the second phase, based on the scores obtained, tries to find promising parameters . Regarding the overall performance of the model, all the useful metrics are displayed through a “Gradient Boosting Output” component, on the right side of the workflow, which gives a complete description of the model, namely the ROC curve, the Lift Chart and some other statistics and values.

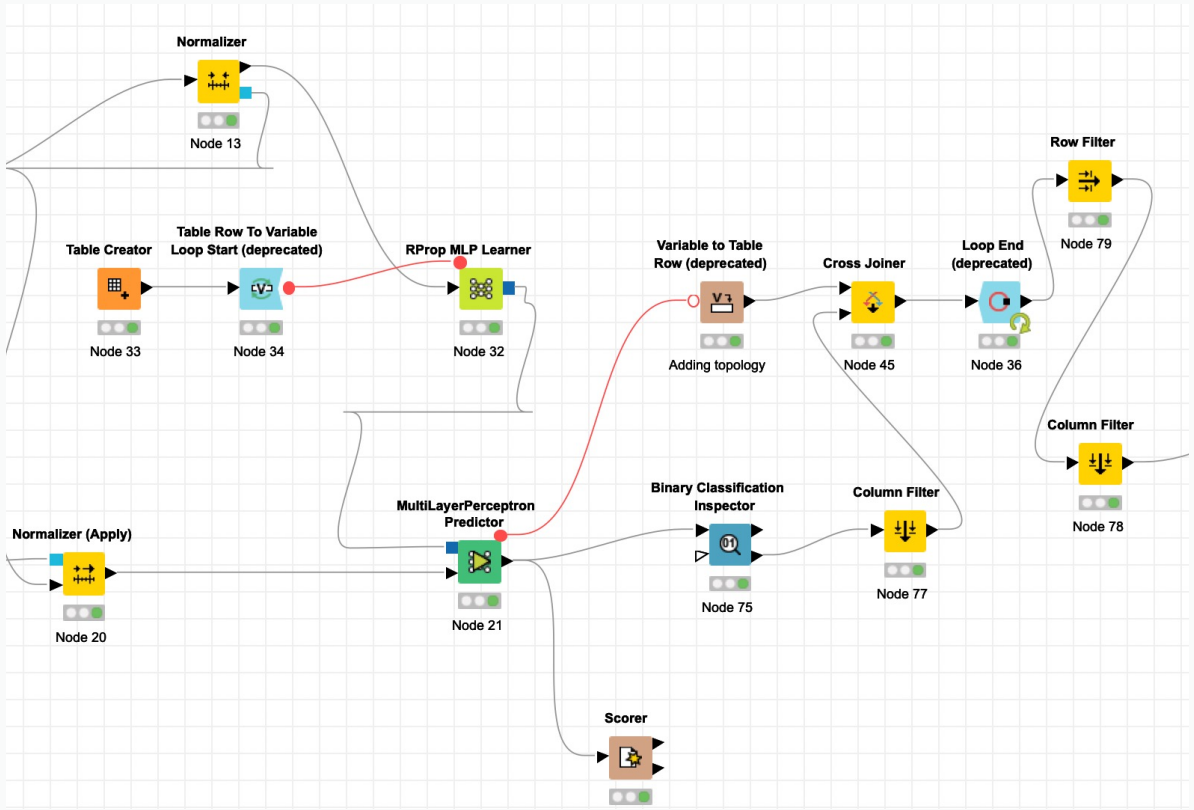
Logistic Regression

Since our target variable is a binary one, we also proposed as a third model a simple Logistic Regression model, with the same partitioning rule as before.

To begin with, we have put two **Normalizer** nodes at the very beginning of the workflow, so that both training and test set fields are normalized according to a Min-Max scaling [0,1].
Then we have chosen the "Iteratively reweighted least squares" as a solver, with a maximum number of epochs of 10,000 and an epsilon equals to 1.0E-5.

Immediately after the learning node, we added three **Math Formula** components to compute respectively the Odds ratio, the 95% C.I. lower and upper bound.
As usual, all the coefficients tables can be retrieved by opening up the Interactive Table node, whereas all the metrics are displayed in the Binary Classification Inspector output.





Multilayer Perceptron

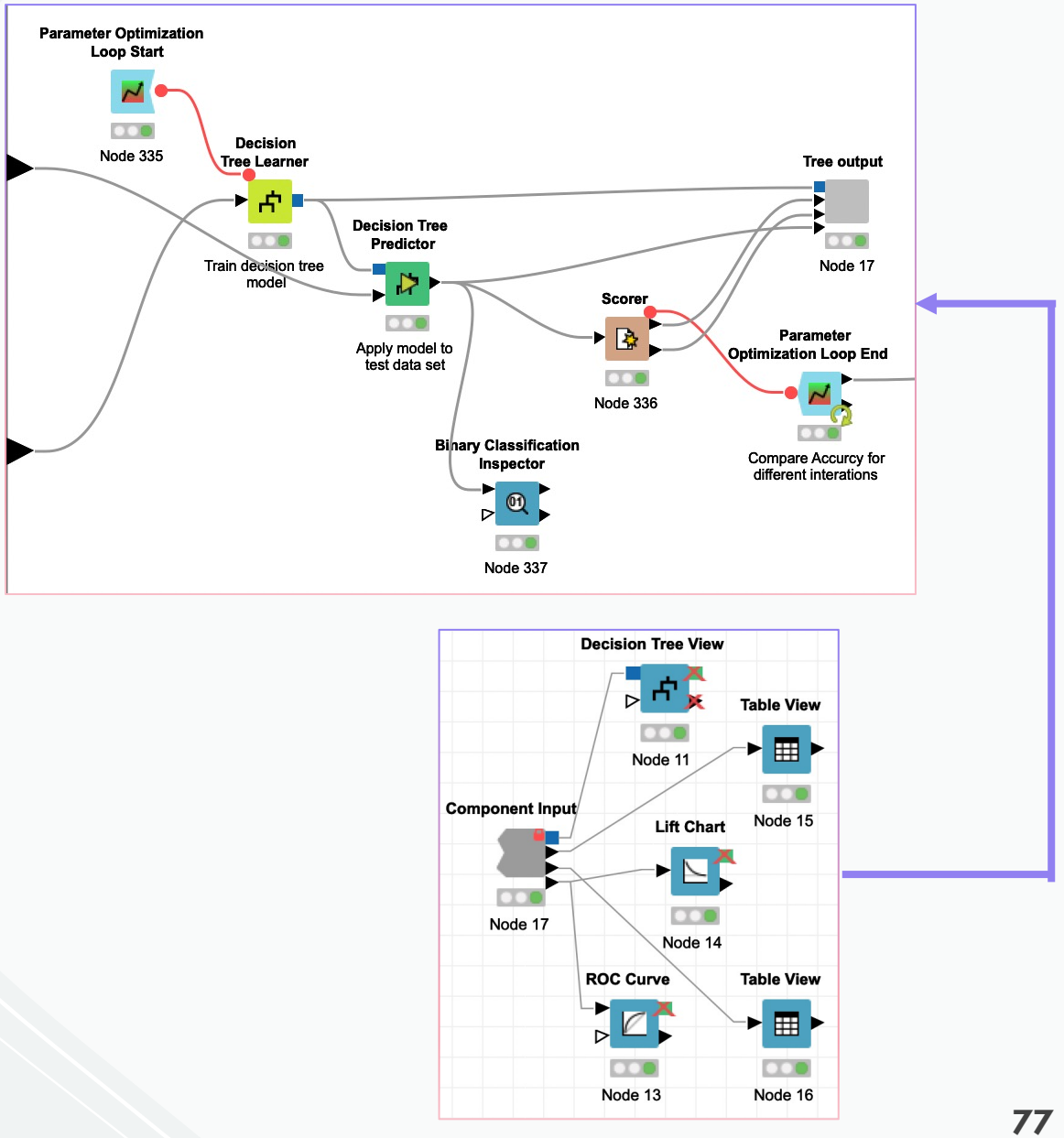
As additional check, we tested a MLP model as well. First, we normalized the regressor with a Min-Max rule [0,1] to have them on the same scale and to have a better model performance. Using an **Rprop MLP Learner** node, we tuned the number of hidden layers in the network and the number of neurons per each layer, specifying a certain number of iterations.

Classification Tree Optimized

Lastly, we added the Classification Tree. As a sign of gratitude for being the ancestor of all the previous models, we included it knowing that it would have poorly performed.

In the **Decision Tree Learner**, we opted for a Gini Impurity measure to evaluate the quality of the splits, with no pruning and we tuned the maximum number of splits and the minimum number of datapoints to be evaluated at each node.

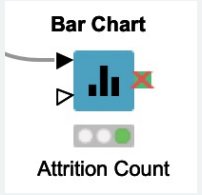
Regarding the Loop nodes, we engaged in a “Brute Force” search strategy.



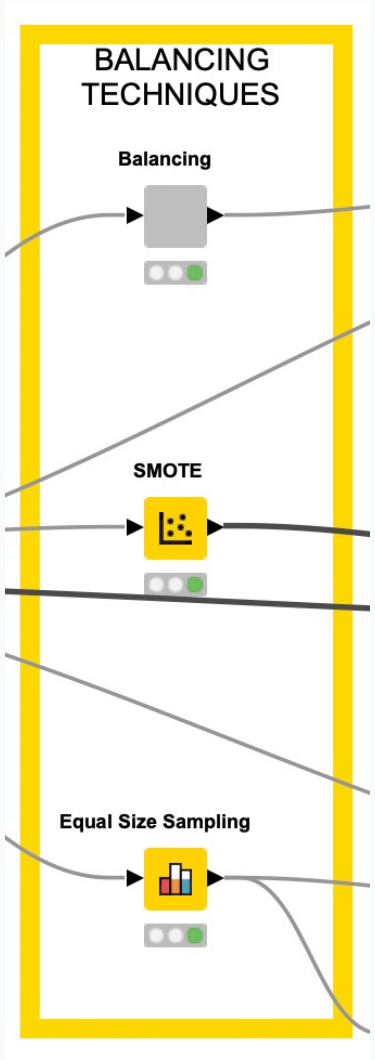
Balancing Techniques

Before implementing any model on our data, we decided to focus on our target variable, namely **Attrition**. By using the **Bar Chart** node, we explored its distribution and we noticed a strong unbalancing between the two classes:

- 237 “yes” occurrences
- 1233 “no” occurrences



Imbalanced classification pose a challenge for predictive algorithms as they are designed for an equal number of examples for each class. Usually this is a problem since the minority class is the one we are more interested in. The model is biased towards the majority class, thus being prone to classification errors for the minority class.



To work around the problem, we altered the *a priori* distribution of the class of the training set. There are many resampling techniques that can be applied; in particular, we have chosen the following three:

- Balancing (SMOTE Oversampling rate = 4, Undersampling rate =0.85). We obtain 1675 observations : 890 “Yes” & 785 “No”.
- SMOTE, to oversample the minority class. We obtain 1848 observations : 924 “Yes” & 924 “No”.
- Equal Size Sampling, to undersample the majority class to have the same number of datapoints as in the minority one. We obtain 356 observations : 178 “Yes” & 178 “No”.

Sampling Pros & Cons

	+	−
Balancing	▪ It equilibrates the two balances minimizing the variability loss which is generated using Smote only ▪ Allows to have many datapoints (differently from undersampling)	▪ Cutoff on the majority class losing information that might be helpful
Undersampling	▪ It balances the two classes with respect to the minority class ▪ Gives importance to the minority class	▪ It cuts many values belonging to the majority class ▪ Might have too few datapoints for training a model
Oversampling	▪ It balances the two classes with respect to the majority class using a Nearest-Neighbour approach to keep the data variability	▪ It might have to create too many values thus incurring in repetition ▪ Models are not properly learning the oversampled class

OUTCOMES

Models Results and Performances

1. BALANCED MODELS
2. OVERSAMPLING MODELS
3. UNDERSAMPLING MODELS

Balanced Models

Random Forest

Hyperparameter Tuning

In this section we are going to take a careful look at the outcomes of the models presented so far. Following the same order, we start with Random Forest. At the beginning of the Loop, we selected Random Search as optimization strategy passed together with the following parameters:

Parameters

Parameter	Start value	Stop value	Step size	Integer?
Minimum node size	1	30	1.0	☑
Limit number of levels	1	20	1.0	☑
Number of models	10	100	1.0	☑

Max. number of iterations: 100

☑ Early stopping

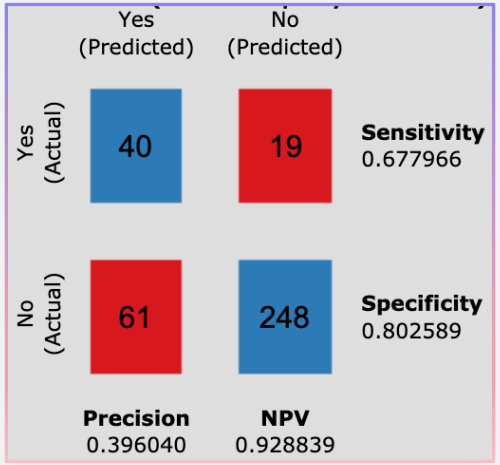
Number of rounds: 5

Tolerance: 0.01

The best parameters for this setting can be retrieved from the Loop End node, together with the Objective value function:

Row ID	Minimum node size	Limit number of levels	Number of models	Objec...
Best parameters	10	19	15	0.875

To get more a sense of real performance, it is helpful to take into account the relative Confusion Matrix:



The level of Accuracy is high, around 83%, and a Precision of 39.6%, which definitely has to be improved. The threshold is set to 0.27

XG Boost Hyperparameter Tuning

Parameter	Start value	Stop value	Step size	Integer?
minobs	5	30	1.0	☑
maxdepth	6	15	1.0	☑
learnrate	0	0.7	0.1	☐

Max. number of iterations	100
Number of warm-up rounds	20
Gamma	0.25
Number of candidates per round	25

The optimal parameters can be read through "Parameter Optimization Loop End", selecting "Best parameteres".

Row ID	I minobs	I maxdepth	D learnrate	D Objec...
Best parameters	14	15	0.023	0.837

Subsequently, these ones above are used to train a "H2O Gradient Boosting Learner" for best results.

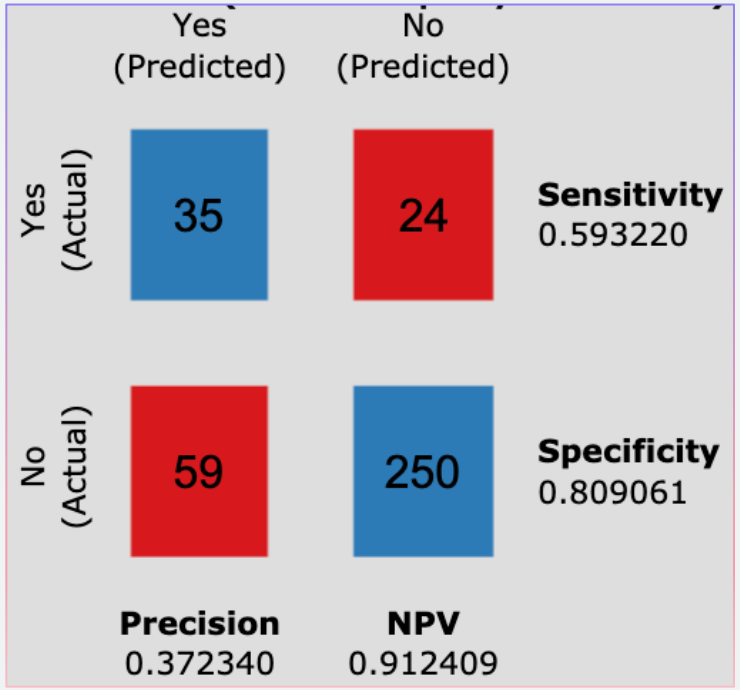
Moving on to XG Boosting, we set:

- Minobs, that specifies the minimum number of observations for a leaf to split;
- Maxdepth, that is the maximum number of levels in a tree;
- Learnrate parameter that suggests the step size to be used to optimize the objective function.

Moreover, the number of iterations is set to 100 with 20 warm-up rounds and 25 candidates per each. Finally, in the Loop End node we stated our objective function to maximize, namely the level of Accuracy.

By investigating the confusion matrix for our target variable Attrition, we discovered that for a threshold of 0.5015, we have sensitivity and specificity, namely 59.3% and 80.9% respectively. Though the precision is extremely low: 37.23%!

In total, 285 observations have been correctly classified, against 83 that, conversely, have been wrong classified, with an Accuracy level of 77.4% and AUC of 70.2%



Multilayer Perceptron Hyperparameter Tuning

	units	layers
Row0	30	5
Row1	30	6
Row2	30	7
Row3	40	5
Row4	40	6
Row5	40	7
Row6	50	5
Row7	50	6
Row8	50	7

Maximum number of iterations: 100

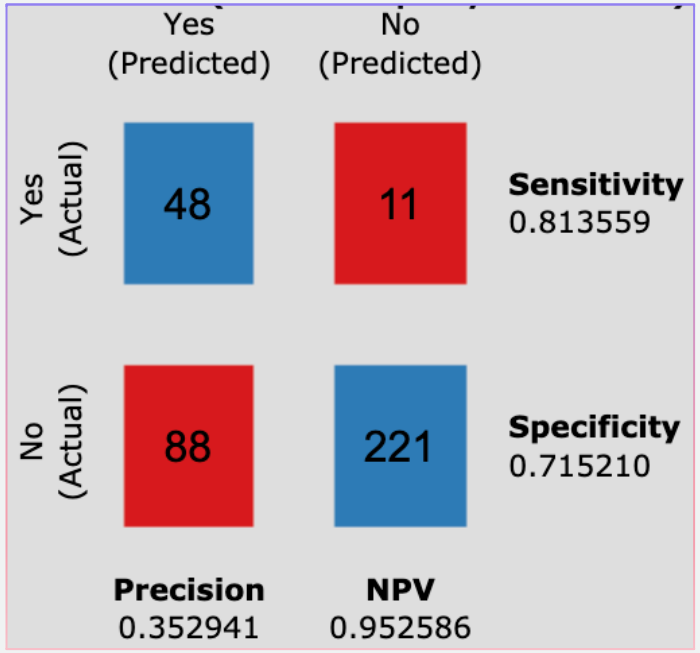
Number of hidden layers: 6

Number of hidden neurons per layer: 30

Row ID	units	layers	Model Name	Accur...	Precisi...	Se...	Specifi...	AUC	Iteration
Row5_P (At...	40	7	P (Attrition=Yes)MLP	0.707	0.333	0.831	0.683	0.795	5
Row1_P (At...	30	6	P (Attrition=Yes)MLP	0.731	0.353	0.814	0.715	0.795	1
Row8_P (At...	50	7	P (Attrition=Yes)MLP	0.728	0.339	0.729	0.728	0.797	8
Row3_P (At...	40	5	P (Attrition=Yes)MLP	0.785	0.404	0.712	0.799	0.803	3
Row4_P (At...	40	6	P (Attrition=Yes)MLP	0.769	0.38	0.695	0.783	0.756	4
Row0_P (At...	30	5	P (Attrition=Yes)MLP	0.799	0.418	0.644	0.828	0.699	0
Row7_P (At...	50	6	P (Attrition=Yes)MLP	0.793	0.407	0.627	0.825	0.778	7
Row6_P (At...	50	5	P (Attrition=Yes)MLP	0.799	0.412	0.593	0.838	0.788	6
Row2_P (At...	30	7	P (Attrition=Yes)MLP	0.829	0.472	0.576	0.877	0.75	2

For the MLP, we searched for the number of hidden layers (5, 6 or 7) and for the number of units per layer (30,40 or 50).

Iterating over these parameters, we obtain 9 different configurations and, evaluating each of them with respect to our final objective, we have chosen to have 6 hidden layers with 30 neurons per layer (approximatively the number of regressors)



Inspecting the Confusion Matrix, we can see very good results for a threshold of 0.031:

- 81.35% Sensitivity
- 71.52% Specificity
- 35.29% Precision

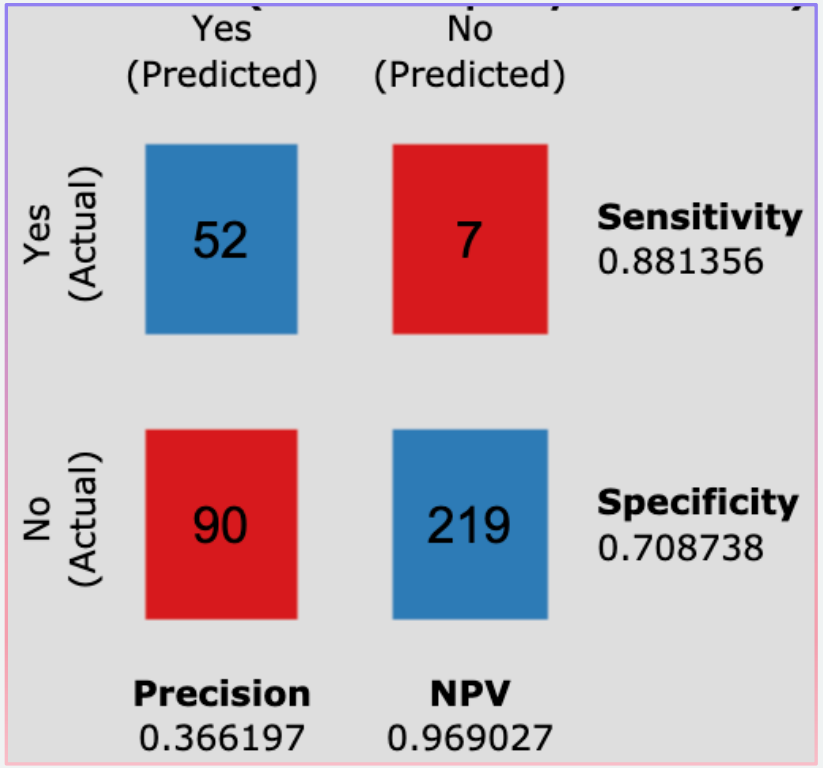
which add up to an almost 79.5% AUC.

Logistic Regression Hyperparameter Tuning

By its own nature, Logistic Regression does not need fine tuning and we directly applied the model.

The results obtained are shown in the Confusion Matrix.
Not only it is overperforming compared to other models in both Specificity and Sensitivity, but it also achieves an AUC of 85.5%.

The threshold is set to 0.39.



Classification Tree Hyperparameter Tuning

For the Decision tree, we tuned the minimum number of records per node and the maximum number of binary splits to execute, obtaining 9 and 2 as best parameters of a Brute Force Search.

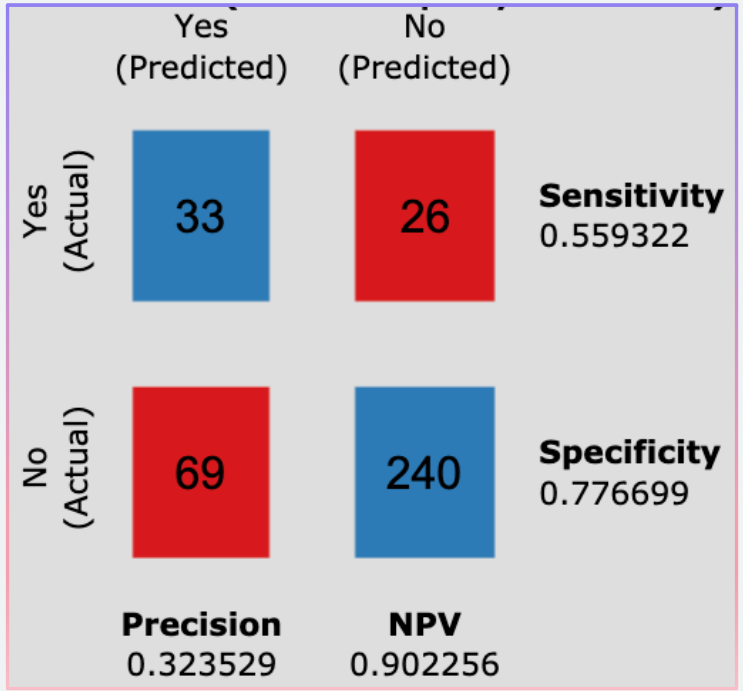
Parameters

Parameter	Start value	Stop value	Step size	Integer?
minNumberRecordsPerNode	5	40	1.0	☑
maxNumNominalValues	2	10	1.0	☑

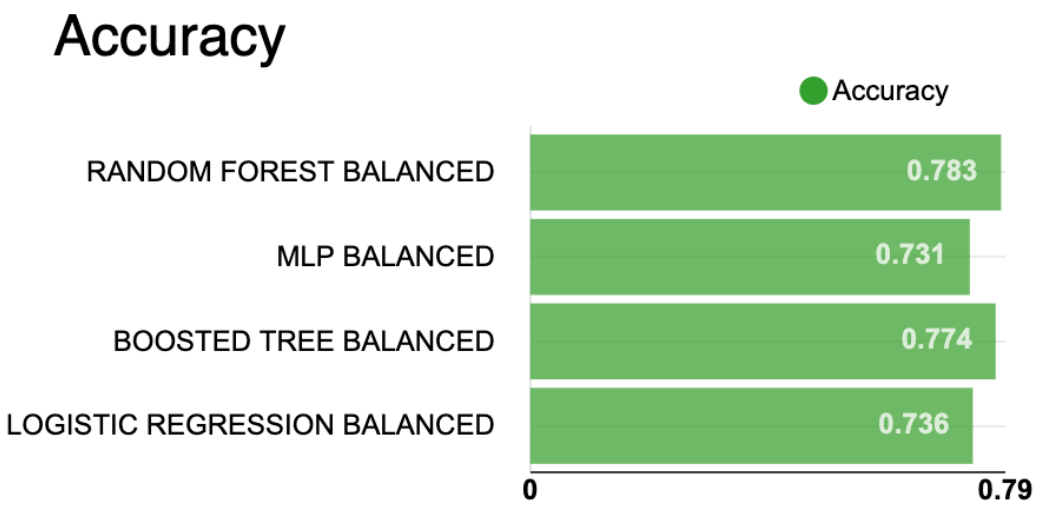
Row ID	minNumberRecordsPerNode	maxNumNominalValues	Objec...
Best parameters	9	2	0.807

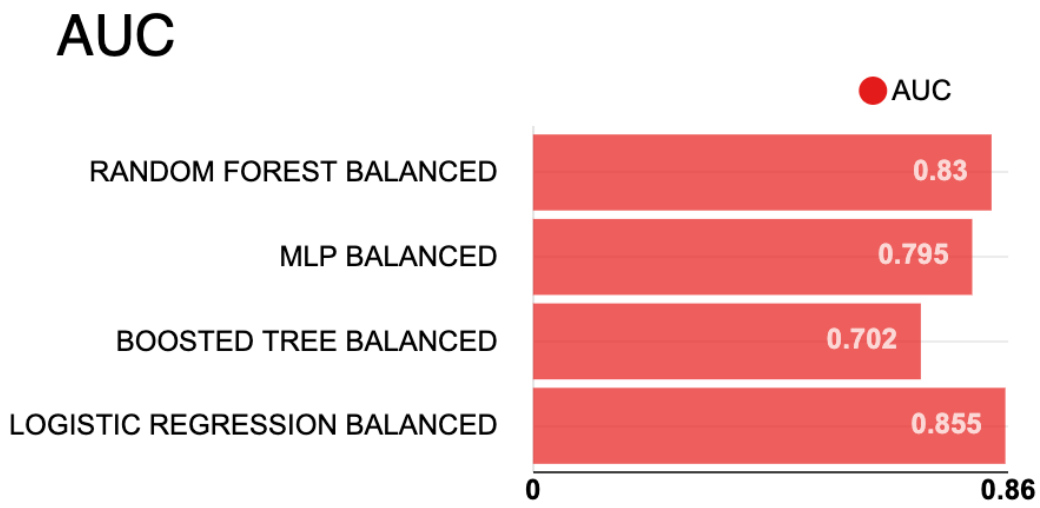
Looking at the Confusion Matrix we immediately grasp that it is not the best model for this task: 77.7% specificity but only 55.9% Sensitivity and 32.35% Precision which are the matrices we are mostly interested in.

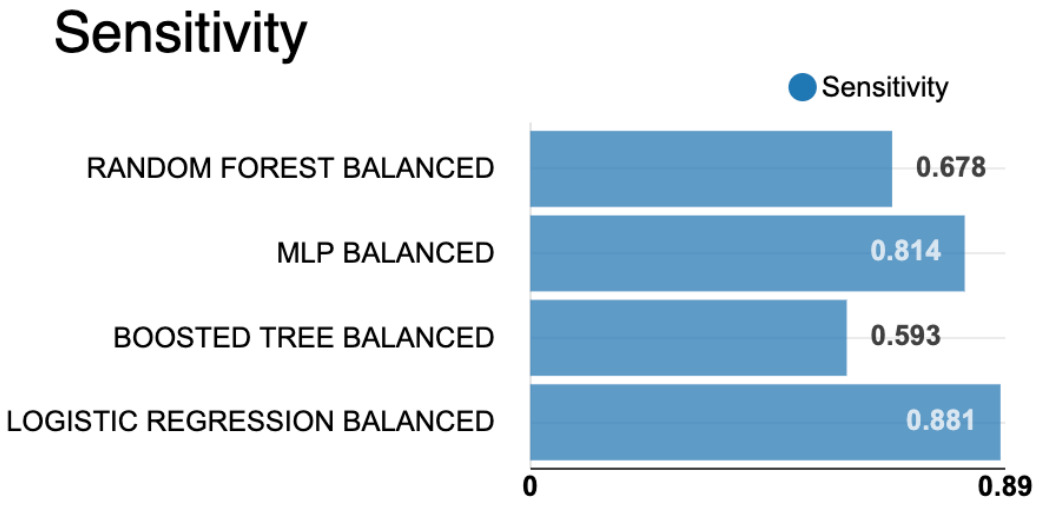
The threshold is set to 0.3 to maximise Max Youden's Index.

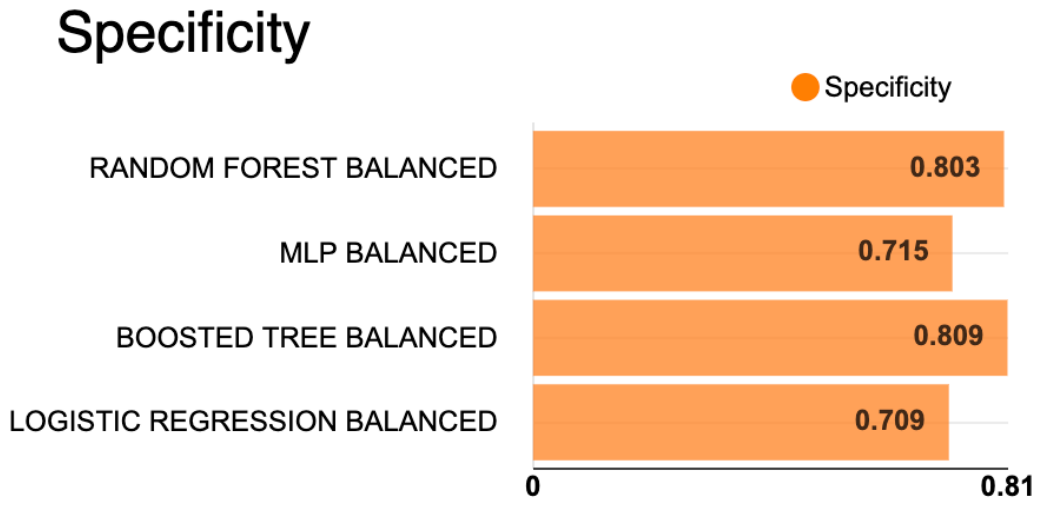


Model Comparison









Oversampled Models

DISCLAIMER: in the following two sections, we will show the same slides's structure shown for the Balanced models changing the outcomes

Random Forest

Hyperparameter Tuning

In this section we are going to take a careful look at the outcomes of the models presented so far. Following the same order, we start with Random Forest. At the beginning of the Loop, we selected Random Search as optimization strategy passed together with the following parameters:

Parameters

Parameter	Start value	Stop value	Step size	Integer?
Minimum node size	2	30	1.0	☑
Limit number of levels	1	10	1.0	☑
Number of models	10	100	1.0	☑

Max. number of iterations: 100

☑ Early stopping

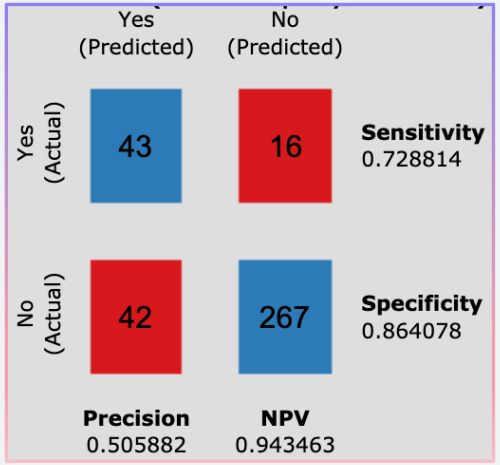
Number of rounds: 5

Tolerance: 0.01

The best parameters for this setting can be retrieved from the Loop End node, together with the Objective value function:

Row ID	Minimum node size	Limit number of levels	Number of models	Objec...
Best parameters	20	8	78	0.878

To get more a sense of real performance, it is helpful to take into account the relative Confusion Matrix:



The level of Accuracy is high, around 84%, and a Precision of 50.6%, which definitely has to be improved. The threshold is set to 0.256.

XG Boost Hyperparameter Tuning

Parameter	Start value	Stop value	Step size	Integer?
minobs	5	30	1.0	☑
maxdepth	6	15	1.0	☑
learnrate	0	0.7	0.1	☐

Max. number of iterations	100
Number of warm-up rounds	20
Gamma	0.25
Number of candidates per round	25

The optimal parameters can be read through "Parameter Optimization Loop End", selecting "Best parameteres".

Row ID	I minobs	I maxdepth	D learnrate	D Objec...
Best parameters	12	9	0.39	0.837

Subsequently, these ones above are used to train a "H2O Gradient Boosting Learner" for best results.

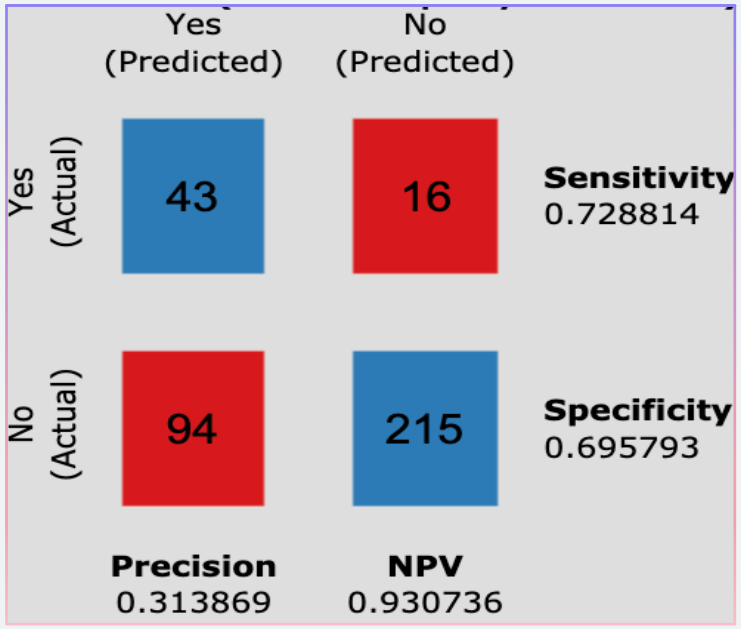
Moving on to XG Boosting, we set:

- Minobs, that specifies the minimum number of observations for a leaf to split;
- Maxdepth, that is the maximum number of levels in a tree;
- Learnrate parameter that suggests the step size to be used to optimize the objective function.

Moreover, the number of iterations is set to 100 with 20 warm-up rounds and 25 candidates per each. Finally, in the Loop End node we stated our objective function to maximize, namely the level of Accuracy.

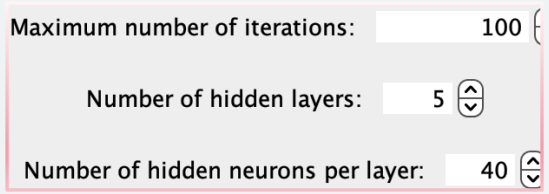
By investigating the confusion matrix for our target variable Attrition, we discovered that for a threshold of 0.211, we have a pretty balanced sensitivity and specificity, namely 72.9% and 69.6% respectively. Though the precision is extremely low: 31.4% !

In total, 258 observations have been correctly classified, against 110 that, conversely, have been wrong classified, with an Accuracy level of 69.92% and AUC of 75.59%



Multilayer Perceptron Hyperparameter Tuning

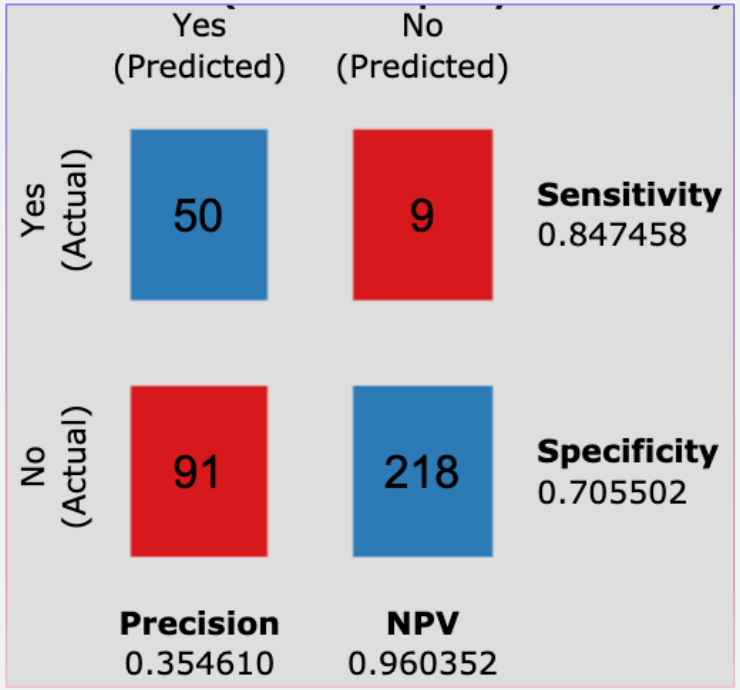
	units	layers
Row0	30	5
Row1	30	6
Row2	30	7
Row3	40	5
Row4	40	6
Row5	40	7
Row6	50	5
Row7	50	6
Row8	50	7



Row ID	units	layers	Model Name	Accur...	Precisi...	Se...	Specifi...	AUC	Iteration
Row3_P (At...	40	5	P (Attrition=Yes)MLP	0.728	0.355	0.847	0.706	0.836	3
Row8_P (At...	50	7	P (Attrition=Yes)MLP	0.755	0.376	0.797	0.748	0.758	8
Row0_P (At...	30	5	P (Attrition=Yes)MLP	0.761	0.378	0.763	0.761	0.822	0
Row2_P (At...	30	7	P (Attrition=Yes)MLP	0.753	0.357	0.678	0.767	0.78	2
Row4_P (At...	40	6	P (Attrition=Yes)MLP	0.799	0.421	0.678	0.822	0.82	4
Row6_P (At...	50	5	P (Attrition=Yes)MLP	0.774	0.38	0.644	0.799	0.741	6
Row1_P (At...	30	6	P (Attrition=Yes)MLP	0.815	0.446	0.627	0.851	0.758	1
Row5_P (At...	40	7	P (Attrition=Yes)MLP	0.867	0.581	0.61	0.916	0.784	5
Row7_P (At...	50	6	P (Attrition=Yes)MLP	0.826	0.465	0.559	0.877	0.726	7

For the MLP, we searched for the number of hidden layers (5, 6 or 7) and for the number of units per layer (30,40 or 50).

Iterating over these parameters, we obtain 9 different configurations and, evaluating each of them with respect to our final objective, we have chosen to have 5 hidden layers with 40 neurons per layer (approximatively the number of regressors)



Inspecting the Confusion Matrix, we can see very good results for a threshold of 0.0125 :

- 84.75% Sensitivity
- 70.55% Specificity
- 35.46% Precision

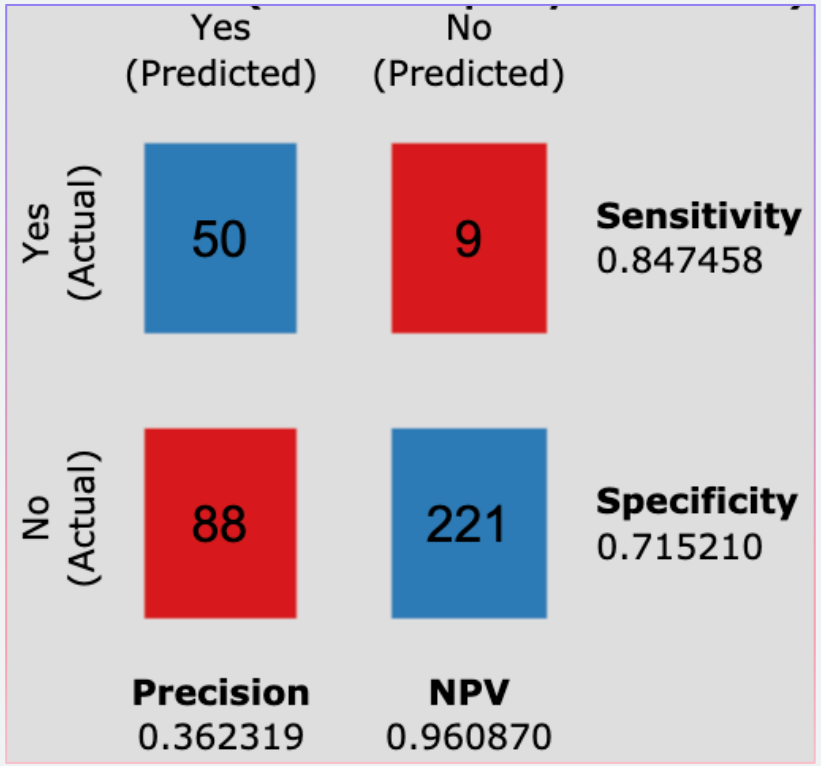
which add up to an almost 83.6% AUC.

Logistic Regression Hyperparameter Tuning

By its own nature, Logistic Regression does not need fine tuning and we directly applied the model.

The results obtained are shown in the Confusion Matrix. Not only it is overperforming compared to other models in both Specificity and Sensitivity, but it also achieves an AUC of 85.85%.

The threshold is set to 0.40.



Classification Tree Hyperparameter Tuning

For the Decision tree, we tuned the minimum number of records per node and the maximum number of binary splits to execute, obtaining 10 and 2 as best parameters of a Brute Force Search.

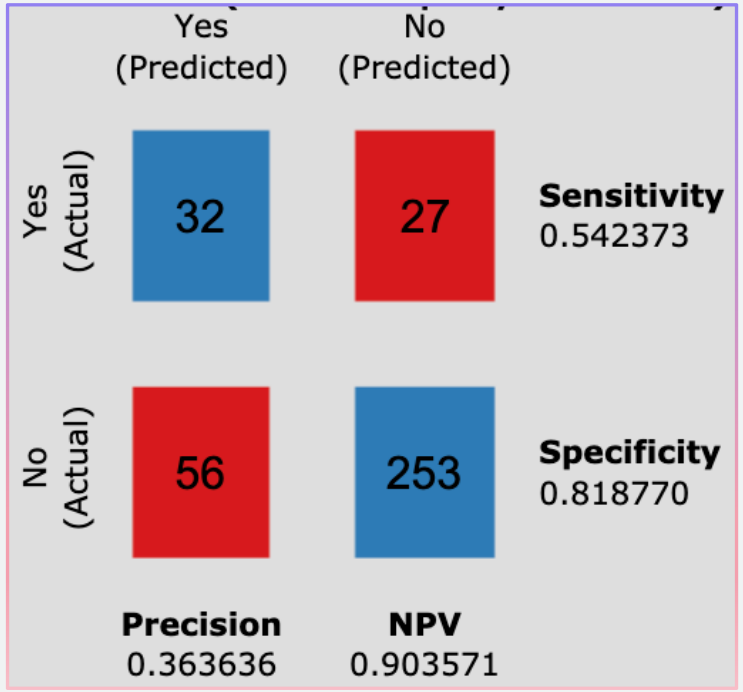
Parameters

Parameter	Start value	Stop value	Step size	Integer?
minNumberRecordsPerNode	5	40	1.0	☑
maxNumNominalValues	2	10	1.0	☑

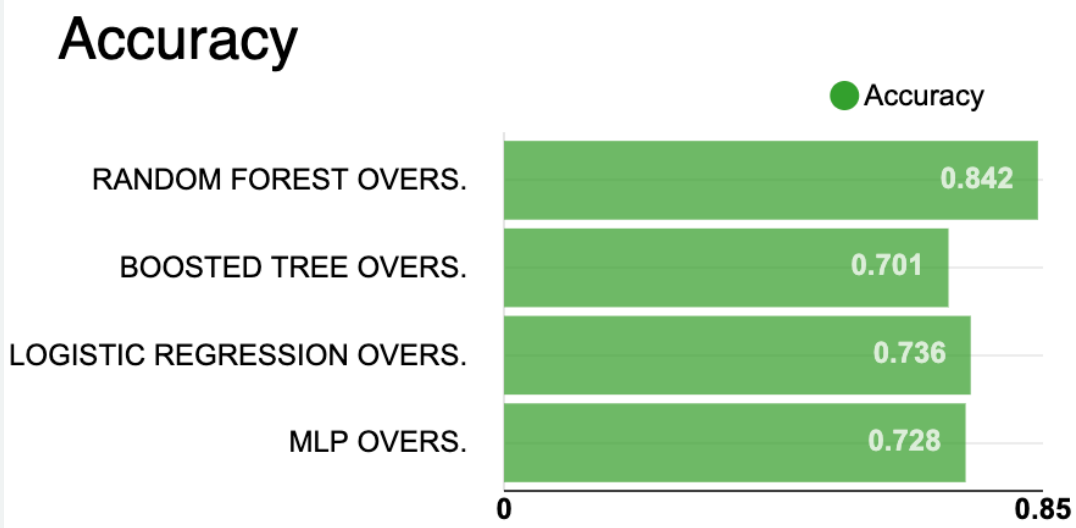
Row ID	minNumberRecordsPerNode	maxNumNominalValues	Objec...
Best parameters	10	2	0.823

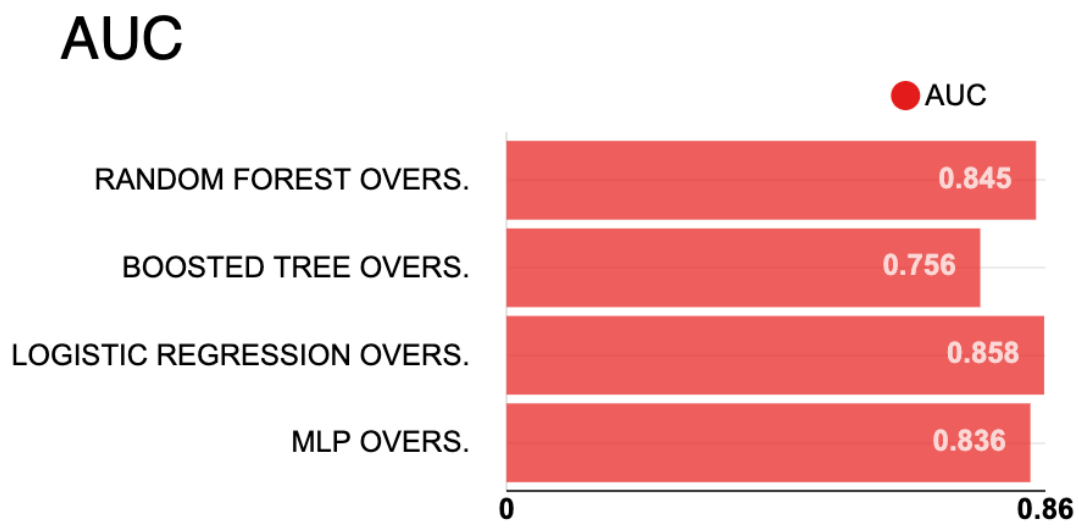
Looking at the Confusion Matrix we immediately grasp that it is not the best model for this task: 81.9% specificity but only 54.4% Sensitivity and 36.36% Precision which are the matrices we are mostly interested in.

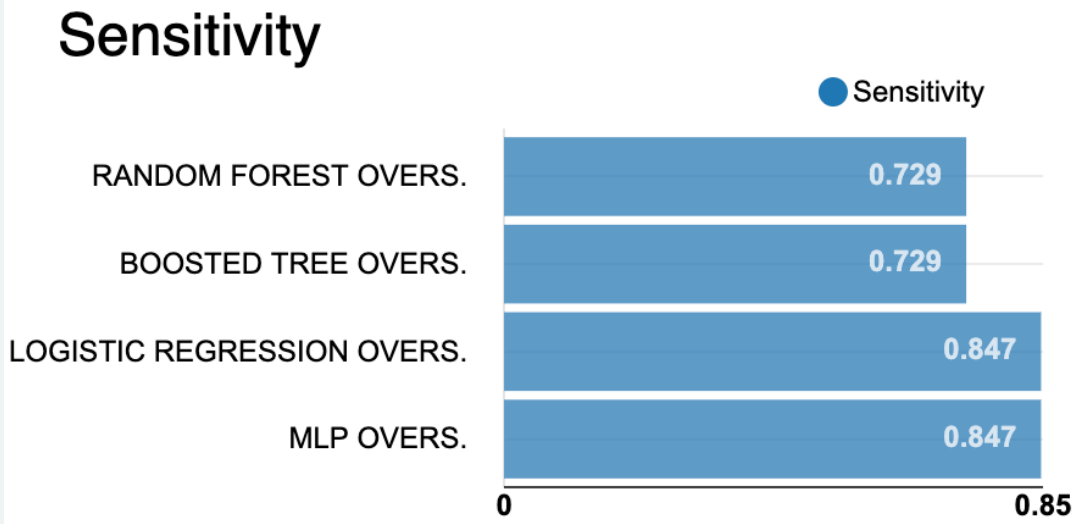
The threshold is set to 0.351 to maximise Max Youden's Index.

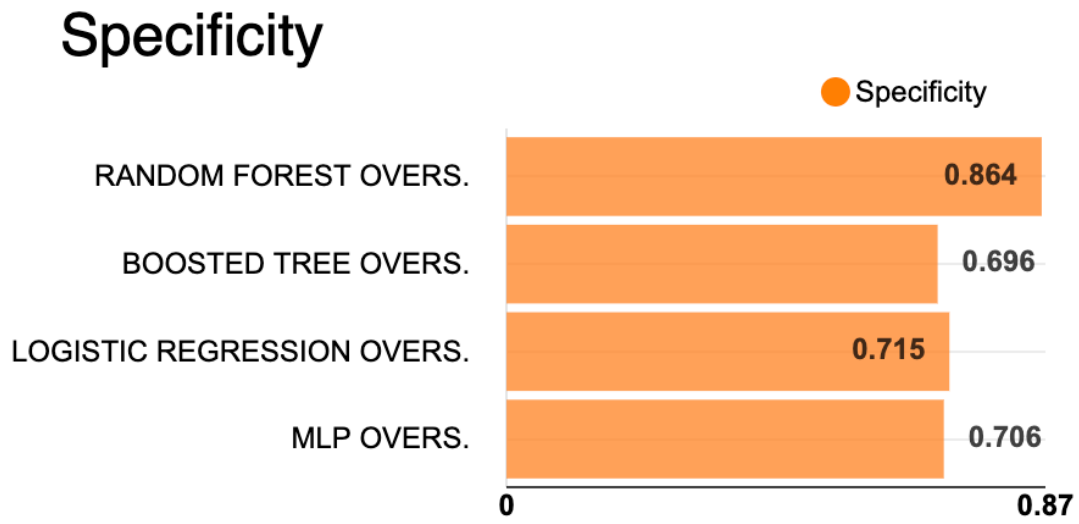


Model Comparison









Undersampled Models

Random Forest

Hyperparameter Tuning

In this section we are going to take a careful look at the outcomes of the models presented so far. Following the same order, we start with Random Forest. At the beginning of the Loop, we selected Random Search as optimization strategy passed together with the following parameters:

Parameters

Parameter	Start value	Stop value	Step size	Integer?
Minimum node ...	1	20	1.0	☑
Limit number of...	1	20	1.0	☑
Number of mod...	10	100	1.0	☑

Max. number of iterations: 100

☑ Early stopping

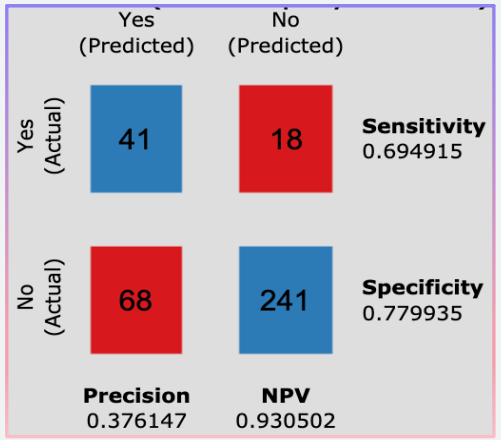
Number of rounds: 5

Tolerance: 0.01

The best parameters for this setting can be retrieved from the Loop End node, together with the Objective value function:

Row ID	Minimum node size	Limit number of levels	Number of models	Objec...
Best parameters	7	16	97	0.638

To get more a sense of real performance, it is helpful to take into account the relative Confusion Matrix:



The level of Accuracy is high, around 76%, and a Precision of 37.6%, which definitely has to be improved. The threshold is set to 0.546.

XG Boost Hyperparameter Tuning

Parameters

Parameter	Start value	Stop value	Step size	Integer?
minobs	1	50	1.0	☑
maxdepth	1	50	1.0	☑
learnrate	0	0,7	0.1	☐

Max. number of iterations	100
Number of warm-up rounds	20
Gamma	0.25
Number of candidates per round	25

The optimal parameters can be read through "Parameter Optimization Loop End", selecting "Best parameteres".

Row ID	I minobs	I maxdepth	D learnr...	D Objec...
Best param...	20	20	0.504	0.707

Subsequently, these ones above are used to train a "H2O Gradient Boosting Learner" for best results.

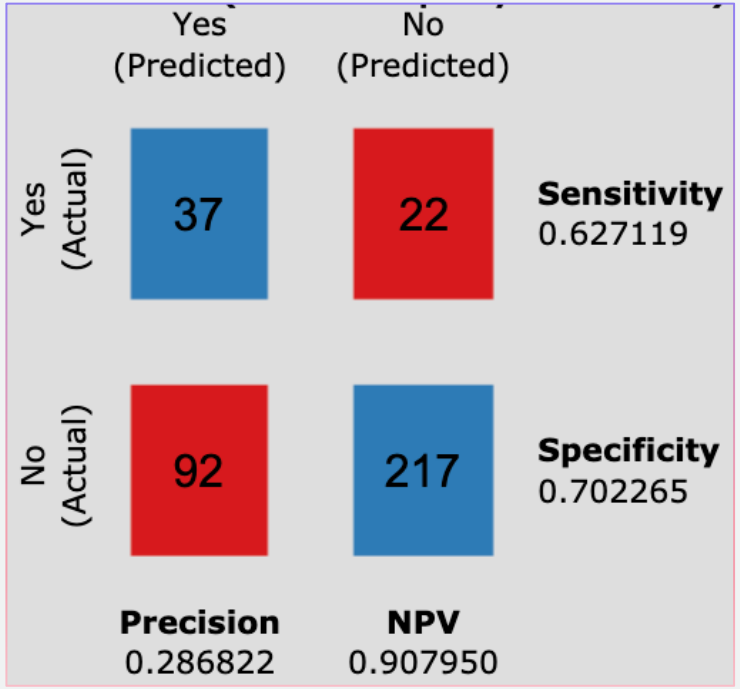
Moving on to XG Boosting, we set:

- Minobs, that specifies the minimum number of observations for a leaf to split;
- Maxdepth, that is the maximum number of levels in a tree;
- Learnrate parameter that suggests the step size to be used to optimize the objective function.

Moreover, the number of iterations is set to 100 with 20 warm-up rounds and 25 candidates per each.
Finally, in the Loop End node we stated our objective function to maximize, namely the level of Accuracy.

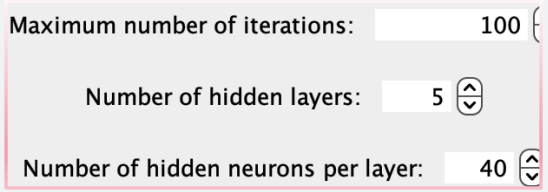
By investigating the confusion matrix for our target variable Attrition, we discovered that for a threshold of 0.62, we have a pretty balanced sensitivity and specificity, namely 62.7% and 70.2% respectively. Though the precision is extremely low: 28.69% !

In total, 254 observations have been correctly classified, against 114 that, conversely, have been wrong classified, with an Accuracy level of 69% and AUC of 73.5%



Multilayer Perceptron Hyperparameter Tuning

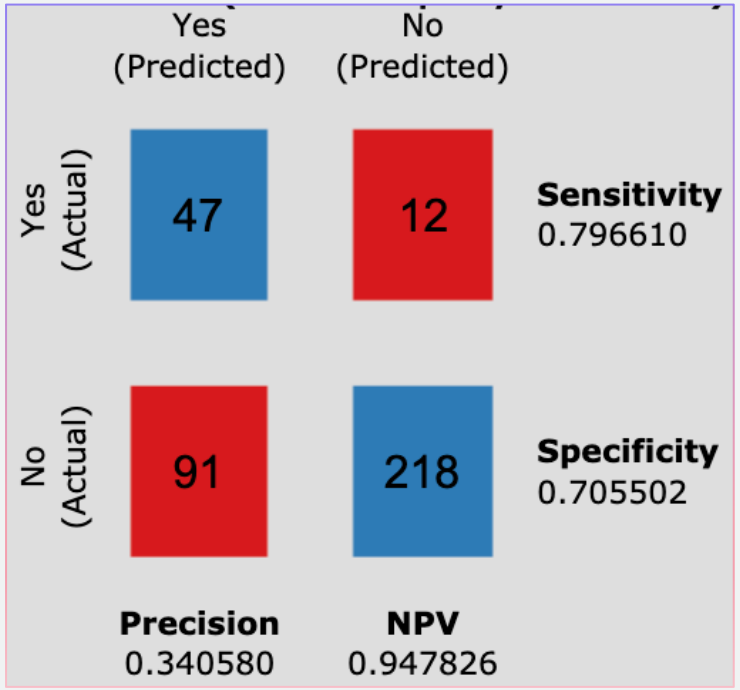
	units	layers
Row0	30	5
Row1	30	6
Row2	30	7
Row3	40	5
Row4	40	6
Row5	40	7
Row6	50	5
Row7	50	6
Row8	50	7



Row ID	units	layers	Model Name	Accur...	Precisi...	Se...	Specifi...	AUC	Iteration
Row3_P (At...	40	5	P (Attrition=Yes)MLP	0.72	0.341	0.797	0.706	0.797	3
Row1_P (At...	30	6	P (Attrition=Yes)MLP	0.761	0.38	0.78	0.757	0.744	1
Row7_P (At...	50	6	P (Attrition=Yes)MLP	0.655	0.287	0.78	0.631	0.773	7
Row5_P (At...	40	7	P (Attrition=Yes)MLP	0.812	0.449	0.746	0.825	0.781	5
Row6_P (At...	50	5	P (Attrition=Yes)MLP	0.704	0.316	0.729	0.699	0.762	6
Row2_P (At...	30	7	P (Attrition=Yes)MLP	0.807	0.438	0.712	0.825	0.741	2
Row8_P (At...	50	7	P (Attrition=Yes)MLP	0.799	0.423	0.695	0.819	0.721	8
Row0_P (At...	30	5	P (Attrition=Yes)MLP	0.804	0.427	0.644	0.835	0.753	0
Row4_P (At...	40	6	P (Attrition=Yes)MLP	0.761	0.351	0.576	0.796	0.678	4

For the MLP, we searched for the number of hidden layers (5, 6 or 7) and for the number of units per layer (30,40 or 50).

Iterating over these parameters, we obtain 9 different configurations and, evaluating each of them with respect to our final objective, we have chosen to have 5 hidden layers with 40 neurons per layer (approximatively the number of regressors)



Inspecting the Confusion Matrix, we can see very good results for a threshold of 0.132:

- 79.66% Sensitivity
- 70.55% Specificity
- 34.06% Precision

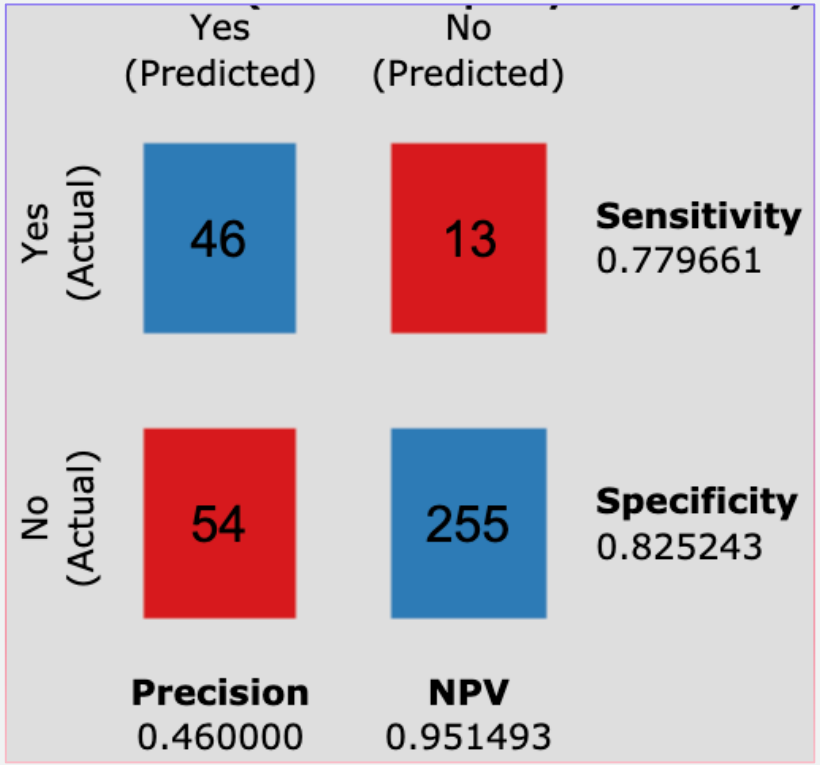
which add up to an almost 80% AUC.

Logistic Regression Hyperparameter Tuning

By its own nature, Logistic Regression does not need fine tuning and we directly applied the model.

The results obtained are shown in the Confusion Matrix. Not only it is overperforming compared to other models in both Specificity and Sensitivity, but it also achieves an AUC of 86.5%.

The threshold is set to 0.557.



Classification Tree Hyperparameter Tuning

For the Decision tree, we tuned the minimum number of records per node and the maximum number of binary splits to execute, obtaining 6 and 2 as best parameters of a Brute Force Search.

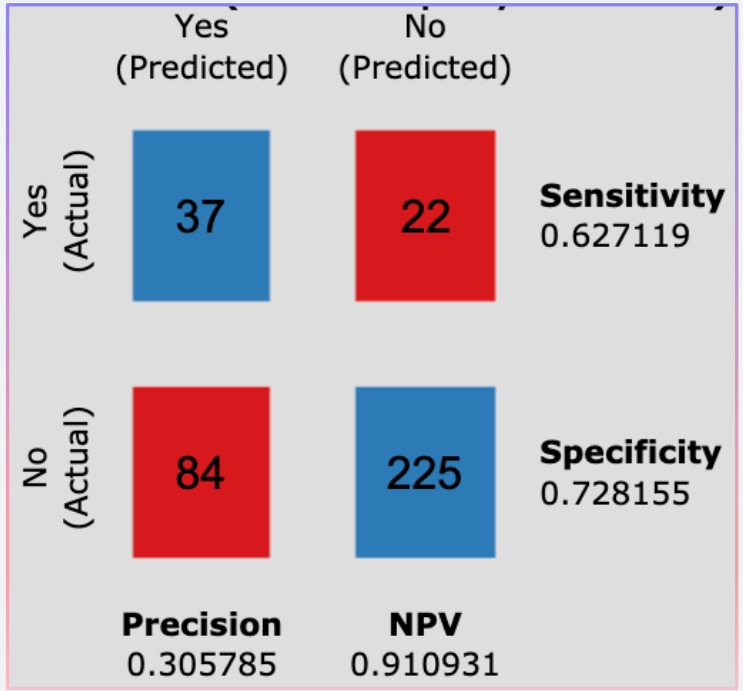
Parameters

Parameter	Start value	Stop value	Step size	Integer?
minNumberRecordsPerNode	5	40	1.0	☑
maxNumNominalValues	2	10	1.0	☑

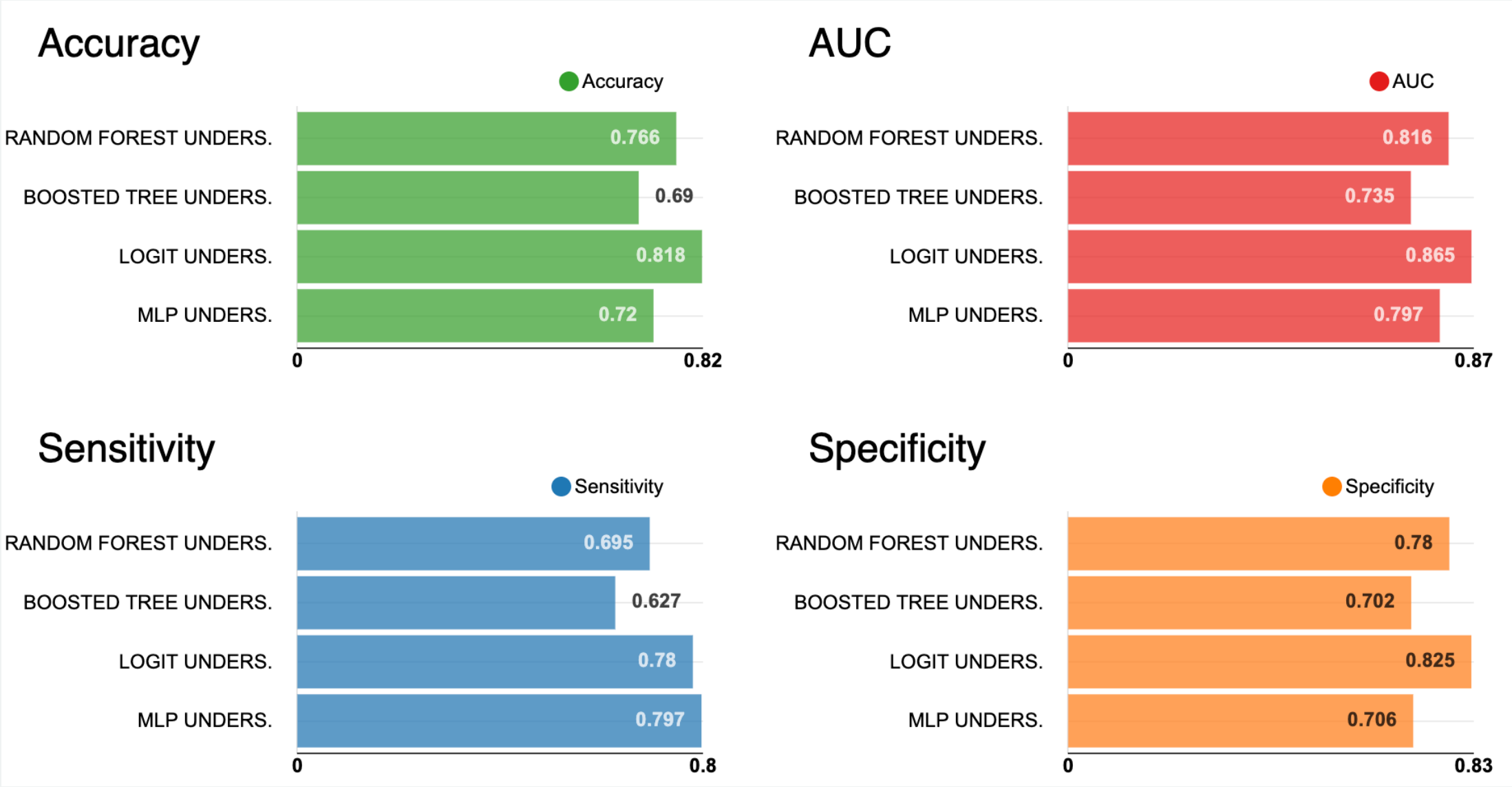
Row ID	minNumberRecordsPerNode	maxNumNominalValues	Objec...
Best parameters	6	2	0.717

Looking at the Confusion Matrix we immediately grasp that it is not the best model for this task: 62.7% specificity and only 72.8% Sensitivity and 30.6% Precision which are the matrices we are mostly interested in.

The threshold is set to 0.50.



Model Comparison



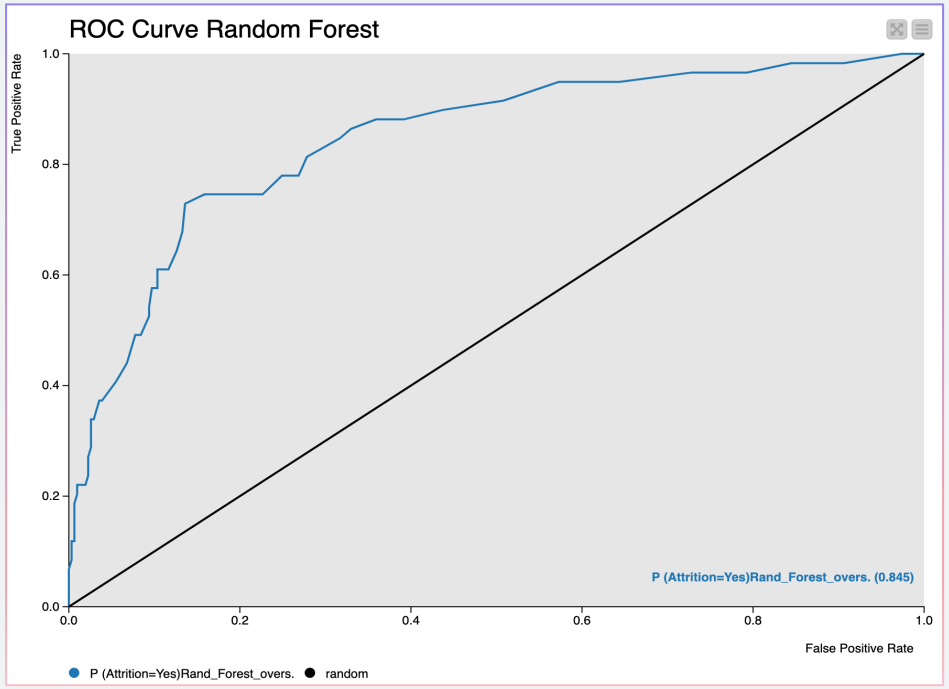
Final Outcomes

The last step, before moving on to managerial implications, is to evaluate all the models, per each balancing technique, according to the metrics we have retrieved.

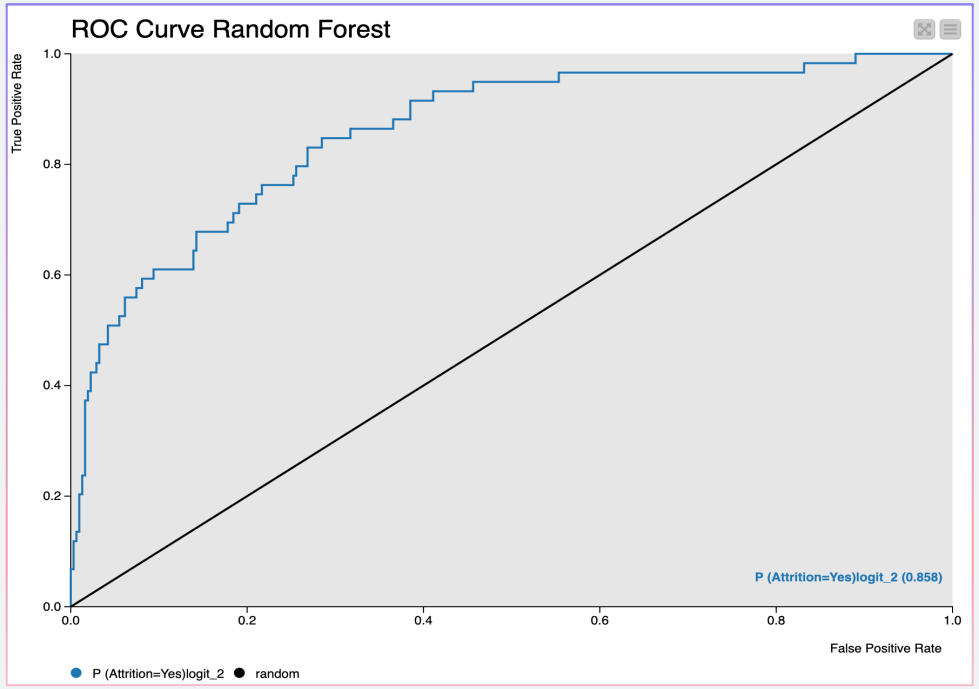
By looking at model comparison slides, we can conclude that the oversampling technique leads to better results compared to the others, in particular Random Forest ensures the highest level of Accuracy, around 84%, Specificity, 86% and Precision of 51% as well. **For these reasons, we decided to pick this one as our final model.**

The second place is instead awarded to the Logit model, which instead has higher AUC, 86% and Sensitivity, around 85%.

ROC Curve - Comparison



Random Forest



Logistic Regression

Considerations on Random Forest Outcomes

Accuracy, Precision and mostly Sensitivity are crucial measures for our prediction.

- With a Sensitivity of about 73%, our model is wrongly classifying a third of the true "quitters" thus not being able to solve all the company's problems in anticipating who is likely to leave. This has a cost for the company which incurs losses in terms of higher employee turnover, spends time and money for hiring , training and integrating new personnel and have them updated on internal processes; or has costs from people abandoning teams working on projects or products.

- Precision is important because it evaluates the percentage of people which we predict will leave and they will actually leave. A low result on this metrics leads to high expenses for the company. In fact, to counteract the willingness to leave, a company concedes bonuses and incentives, increases salaries or employee's position or invests on its tangible assets, namely office, hardware or furniture, for convincing people to stay when they never intended to leave. However, the best we can achieve is 51% given the strong unbalancing of our test set. Still, this is an acceptable measure given the data at hand and we aimed at maximising it to limit company's expenses

- Accuracy is crucial since evaluates the percentage of correctly classified predictions. Having 84% on this metrics allows us to have a good overall estimate of the employees' position towards the company. Moreover, it represents a solid basis for Decision Making in terms of performance improvement and cost limitation.

Results Interpretation: features importance

RowID	#splits (level 0)	#candidates (level 0)	IMPORTANCE
Yes_OverTime	12	12	0.546875
Manufacturing Director_JobRole	10	11	0.4069767441860465
Single_MaritalStatus	12	19	0.33783783783783783
Adult_Age	5	10	0.3132530120481928
JobSatisfaction	4	6	0.3114754098360656
StockOptionLevel	3	12	0.25925925925925924
MonthlyIncome	0	12	0.25806451612903225
Non-Travel_BusinessTravel	5	11	0.24675324675324675
Young_Age	1	5	0.22857142857142856
Healthcare Representative_JobRole	2	9	0.22340425531914893

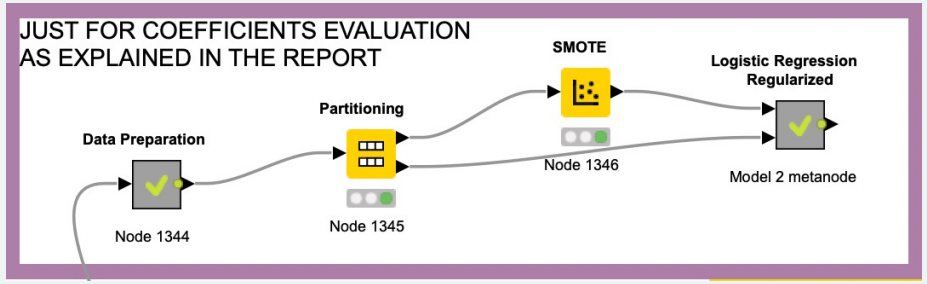
In the table here on the left, we displayed the top-ten features based on the importance measure. YES OVERTIME results to be the most relevant, with around 0.547. The importance metric is calculated as the ratio between the number of times the feature is taken as split in a layer and the sum of the trees in which it was candidate to be split. The best one has a 1 importance on layer 0 which reinforces its crucial position for our prediction.

Importance = sum (#splits) / sum (#candidates)

Then, we have chosen level 0 as in Random Forest models, the importance follows a reversed order, therefore the key features are the first ones we encounter going down in the tree.

However, even though we get insights on the relevance of the variables in predicting Attrition, we are missing their effect on the target variable(shown below).

Results Interpretation: features effect I



We should mention that in our analysis we have to acknowledge and accept the dummy trap. In fact, since we carry out a single data preparation for many models, in order to make the tree-ensambles and MLP perform in the best way, we could not remove valuable features that, unfortunately, cause problems to the coefficients of the Logistic Regression.

However, to obtain a complete and coherent understanding of the regressors' effect on the target variable, and to demonstrate that we understand our issue, we decided to run an additional model, with a different data preparation. Here we get rid of the dummy trap by filtering out a redundant column for each category, keeping N-1 binary variables out of N categories.

As a result, we obtained a classification model which performs similarly to the one with dummy trap but we think this is due to the limited size of the dataset. In addition, its coefficients are more interpretable and more trustworthy as there are many statistically significant variables and more insights about the "Random Forest's relevant features" can be drawn.

Results Interpretation: features effect II

Row ID	Logit	Variable	Coeff.	Std. Err.	z-score	P>...	odds_...	low_95%	upp_9...
Row7	Yes	NumCompaniesW...	2.07	0.249	8.311	0	7.925	1.582	2.558
Row39	Yes	Yes_OverTime	1.863	0.149	12.477	0	6.44	1.57	2.155
Row37	Yes	Single_MaritalStatus	1.775	0.268	6.631	0	5.897	1.25	2.299
Row5	Yes	JobSatisfaction	-1.269	0.192	-6.609	0	0.281	-1.645	-0.893
Row3	Yes	EnvironmentSatisf...	-1.159	0.195	-5.952	0	0.314	-1.541	-0.778
Row10	Yes	RelationshipSatisf...	-1.097	0.19	-5.76	0	0.334	-1.47	-0.724
Row15	Yes	YearsAtCompany	3.407	0.691	4.929	0	30.188	2.052	4.763
Row16	Yes	YearsInCurrentRole	-2.627	0.535	-4.915	0	0.072	-3.675	-1.58
Row22	Yes	Non-Travel_Busin...	-1.314	0.283	-4.652	0	0.269	-1.868	-0.761
Row1	Yes	DistanceFromHome	1.019	0.248	4.109	0	2.771	0.533	1.506
Row12	Yes	TotalWorkingYears	-2.394	0.607	-3.944	0	0.091	-3.583	-1.204
Row4	Yes	JobInvolvement	-1.125	0.292	-3.859	0	0.325	-1.696	-0.554
Row21	Yes	Travel_Frequently...	0.674	0.178	3.779	0	1.962	0.324	1.023
Row35	Yes	Sales Representat...	1.945	0.537	3.618	0	6.992	0.891	2.998
Row40	Yes	Constant	2.255	0.628	3.592	0	9.532	1.024	3.485
Row36	Yes	Research Director...	-3.204	0.907	-3.531	0	0.041	-4.983	-1.426
Row38	Yes	Married_MaritalSt...	0.729	0.209	3.495	0	2.074	0.32	1.138
Row25	Yes	Medical_Educatio...	-2.09	0.625	-3.344	0.001	0.124	-3.314	-0.865
Row23	Yes	Life Sciences_Edu...	-2.109	0.631	-3.343	0.001	0.121	-3.345	-0.872
Row14	Yes	WorkLifeBalance	-0.883	0.279	-3.16	0.002	0.414	-1.43	-0.335
Row28	Yes	Male_Gender	0.45	0.146	3.077	0.002	1.568	0.163	0.736
Row24	Yes	Other_EducationFi...	-2.021	0.674	-3.001	0.003	0.133	-3.341	-0.701
Row18	Yes	Young_Age	0.511	0.184	2.774	0.006	1.667	0.15	0.872
Row13	Yes	TrainingTimesLas...	-0.629	0.232	-2.711	0.007	0.533	-1.084	-0.174
Row19	Yes	Adult_Age	-0.566	0.212	-2.673	0.008	0.568	-0.981	-0.151
Row17	Yes	YearsWithCurrMa...	-1.434	0.55	-2.609	0.009	0.238	-2.511	-0.357
Row31	Yes	Laboratory Techni...	1.187	0.492	2.415	0.016	3.278	0.224	2.151
Row34	Yes	Manager_JobRole	-1.674	0.726	-2.304	0.021	0.188	-3.098	-0.25
Row6	Yes	MonthlyIncome	-1.621	0.714	2.272	0.023	5.06	0.223	3.02
Row26	Yes	Marketing_Educat...	-1.216	0.669	-1.819	0.069	0.296	-2.527	0.094

This second table contains all the coefficients of the "new" Logit model described above, that we use to interpret the effect's magnitude and direction of each feature on our target variable.

We displayed the most statistically significant variables and their coefficient w.r.t. YES ATTRITION; though we need to specify that the coefficients represent the change in log-odds, not a linear change on the target.

YES OVERTIME, the most relevant variable in the Random Forest, has a strong positive effect on the log-odds to leave the company.

Conversely, ENVIRONMENT SATISFACTION decreases the quits.

Using the same reasoning, we can conclude that all the negative coefficients are those on which the company should invest to improve willingness to stay, whereas the positively related ones should be those for which the company carefully accounts: the number of companies an employee worked in (high turnover), home distance from the office or the years in the company.

MANAGERIAL IMPLICATIONS

Conclusions

Why is employee attrition significant?

As previously discussed, reducing employee attrition is of a huge importance for every successful company. Reduced attrition not only enables the company to have more employees with the knowledge of its internal processes, but it also guarantees a good level of stability and satisfaction inside the company.

All these then lead to increased efficiency, making both the company and its employees better off. In addition to that, the company avoids incurring extra costs that come with selection and onboarding processes of new employees.

What causes employee attrition/loyalty?

From results interpretation of the Logistic Regression we infer that there are several features that have a strong positive impact on the likelihood of an employee to leave the company. Namely, those are YES OVERTIME, NUMBER OF COMPANIES WORKED, SINGLE MARITAL STATUS, DISTANCE FROM HOME, TRAVEL FREQUENTLY, YOUNG AGE, YEARS AT COMPANY and MALE GENDER.

On the contrary, there is also a set of features that has a strong negative impact on the likelihood of an employee to leave the company. Some of them are JOB SATISFACTION, ENVIROMENT SATISFACTION and RELATIONSHIP SATISFACTION, JOB INVOLVEMENT, NON TRAVEL, TOTAL WORKING YEARS, YEARS WITH CURRENT MANAGER, ADULT AGE and WORK-LIFE BALANCE.

How to tackle factors that lead to attrition?

YES OVERTIME

A suggestion for the company would be to give more value to hours that are being worked overtime, either through monetary remuneration or through additional days of holidays available to employees.

NUMBER OF COMPANIES WORKED

Results are showing that employees who changed their job more often in the past have a higher chance of continuing to do so, so an idea for the company is to avoid giving them the most important roles, since many other employees, as well as processes would then depend on them, and thus stability is required.

How to tackle factors that lead to attrition?

DISTANCE FROM HOME

Another factor that highly affects employees' attrition is distance between their homes and offices.

Luckily for the company, there are many different ways to tackle this – from encouraging remote working to offering private transportation to employees or providing them with reimbursement of travel expenses, according to their distance (for example, vouchers for gasoline); in addition to that, flexible working hours could be helpful for employees who have to go through a lot of traffic on a daily basis to get to their offices.

How to tackle factors that lead to attrition?

YEARS AT COMPANY

Naturally, after some time spent at the company, employees are starting to looking for opportunities to climb the career ladder. To make them look for changes inside the company, and not elsewhere, managers should show their employees that their history at the company is appreciated; again, there are many ways of doing so – gradual increase of wage, promotion to higher positions, assignment of more suitable tasks, satisfaction of specific needs that employees have

DYNAMIC PROFILE(YOUNG AGE, SINGLE MARITAL STATUS, TRAVEL FREQUENTLY)

If a person is young, single and he is made to travel frequently, he is more likely to leave the company since he is less risk averse and willing to accept changes. He has no family and he is young, which make him less fearful the instability. The company should carefully interact with employees that show this profile by giving them more responsibilities, involving them more in decision making and, maybe, reduce the amount of business trips.

How to enhance employees' loyalty?

JOB SATISFACTION, ENVIRONMENT SATISFACTION & RELATIONSHIP SATISFACTION

As we could assume, some of the most important factors that negatively impact attrition, and thus lead to enhanced loyalty, are employees' satisfaction with their jobs, and working environment in general, as well as relationships inside that environment, so the conclusion is that those three should be maintained and kept high by the company.

YEARS WITH CURRENT MANAGER

The results are showing that one of the aspects in which employees are not looking for a change is who is their manager; thus, a suggestion for the company would be to either try to reduce number of changes in managerial positions, or to try to maintain teams formed around managers once they are ready to be promoted.

How to enhance employees' loyalty?

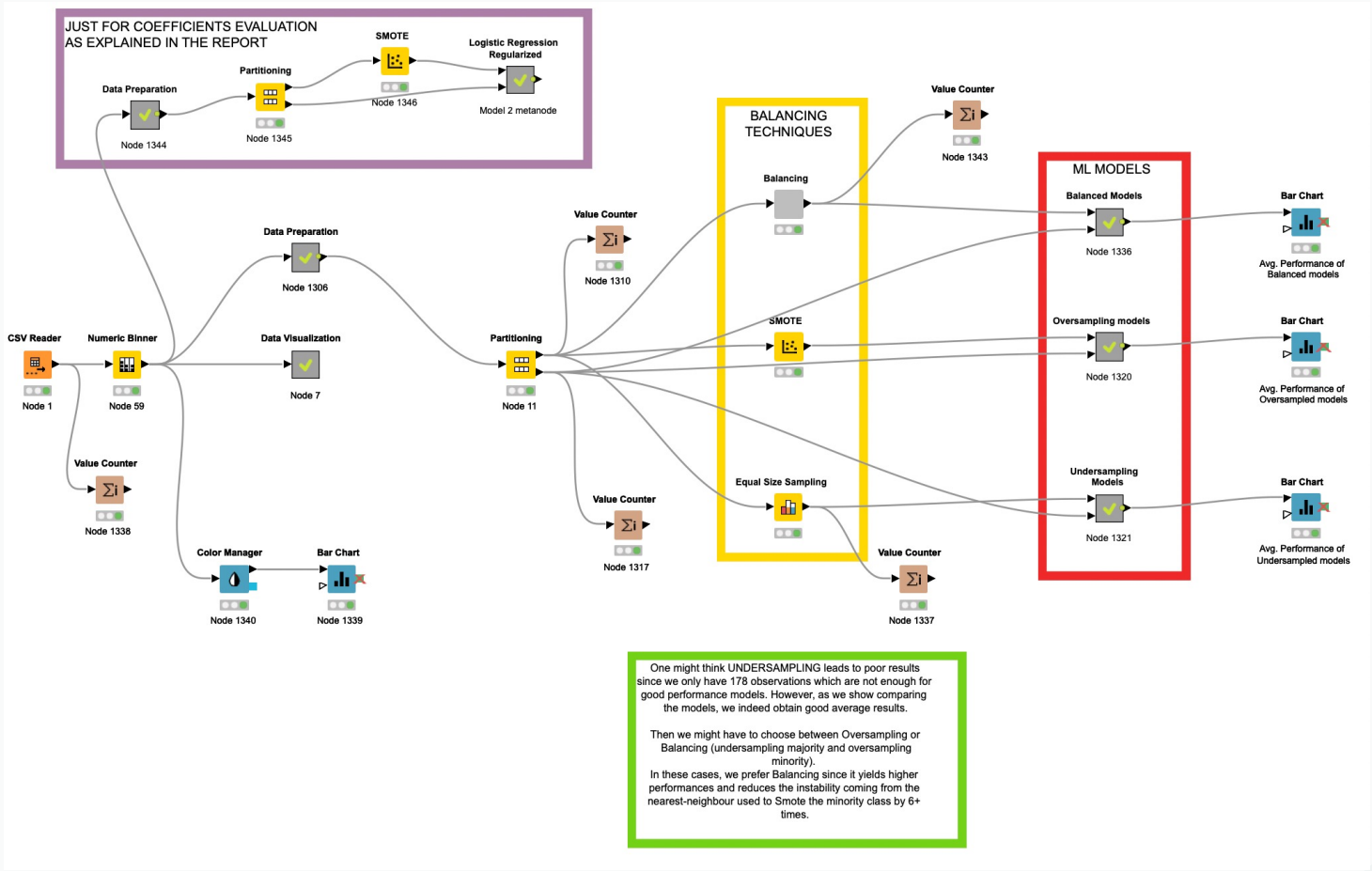
JOB INVOLVEMENT

The most efficient way for the company to see how its employees perceive their involvement in everyday tasks (especially the important ones) is to schedule some HR interviews and/or surveys with them, being conducted on regular basis (e.g. weekly, monthly, quarterly).

WORK-LIFE BALANCE

One last thing that the company needs to consider is the work-life balance of employees, and how to maintain it. However, since employees' profiles differ from industry to industry, suggestions on how to improve their work-life balance differ as well. However, this is a theme that has been growing in importance in the last decade and needs to be carefully addressed.

THE END


JUST FOR COEFFICIENTS EVALUATION
AS EXPLAINED IN THE REPORT
Data Preparation
Node 1344
Partitioning
Node 1345
SMOTE
Node 1346
Logistic Regression Regularized
Model 2 metanode
CSV Reader
Node 1
Numeric Binner
Node 59
Data Preparation
Node 1306
Data Visualization
Node 7
Value Counter
Node 1338
Color Manager
Node 1340
Bar Chart
Node 1339
Partitioning
Node 11
Value Counter
Node 1310
Value Counter
Node 1317
BALANCING TECHNIQUES
Balancing
SMOTE
Equal Size Sampling
Value Counter
Node 1343
Value Counter
Node 1337
ML MODELS
Balanced Models
Node 1336
Oversampling models
Node 1320
Undersampling Models
Node 1321
Bar Chart
Avg. Performance of Balanced models
Bar Chart
Avg. Performance of Oversampled models
Bar Chart
Avg. Performance of Undersampled models
One might think UNDERSAMPLING leads to poor results since we only have 178 observations which are not enough for good performance models. However, as we show comparing the models, we indeed obtain good average results.
Then we might have to choose between Oversampling or Balancing (undersampling majority and oversampling minority).
In these cases, we prefer Balancing since it yields higher performances and reduces the instability coming from the nearest-neighbour used to Smote the minority class by 6+ times.