IBM human resources data analysis

Predicting and understanding employee attrition

Bruguet Marie, Bulliard Loris, Pion Victor

A final report presented for the master
Econometrics, Big Data and Statistics
(EBDS)

Aix-Marseille School of Economics
Aix-Marseille Université
2021-2022

1 Introduction

1.1 Context

Employee attrition is a significant issue for companies. Indeed, human resources invest a lot to recruit and train workers who become valuable assets for companies. Especially in our economic context, more than 50% of French firms report having hiring difficulties (Banque de France).
Indeed, workers can be specialized and it becomes difficult to find someone with the same skills, especially in times of global labor shortage. It can be time consuming to find another worker and train him until he reaches the level required to be independent in his tasks. Predicting attrition is also important for the company to be proactive, to better anticipate what type of employee will leave in the following months. For instance, understanding why workers leave and improving their working conditions to encourage them to stay constitute a way for the company to be proactive. We can imagine a trade-off between the cost of improving employee benefits to incentive them to stay and the cost of attrition by letting them leave.

It's harder to measure, but highly motivated employees have a significant impact on business productivity. Our work will focus on predicting if an employee will leave the company and understand why by establishing different profiles. We can make a small digression about ethics. These algorithms can also be used to discriminate against some categories of people. For instance, if a company realizes that until now, a certain type of people resign quickly, it can decide to stop recruiting them.

1.2 Literature review

As the work presented here has several parts : attrition studies, machine learning models, penalization methods and interpretability of the models with the shape values. We conducted our literature review by selecting some papers that helped us to do our analysis.

Attrition prediction and interpretation is a topic that has already been studied in the human ressources domain, moreover the particular IBM dataset is often used because it is rare that a firm allows to use their employee data for a research publication. Thus the literature we read about is mainly based on the same dataset we will use. First, **Aseel Qutub et all. (2021)** trained 5 different samples and sub-sample drawn from the IBM attrition dataset. They used 6 different prediction models, logistic linear regression, decision tree, random forest, adaboost model, gradient boosting model and ensemble methods which are a combination of several learning algorithms. The main result is that the logistic regression model is the one giving the best prediction, relying on several metrics evaluation such as the accuracy or the AUC. The authors put forward that it can be due to the small size of the sample and that other models can perform better with a larger dataset.

Yang and Islam (2021) investigated the reasons why employee choose to resign. They first used Random Forest and K-means clustering to select the most important features that have an obvious impact on employee attrition. In a second part, they use a logistic regression model to compare the difference between the two classes of employee i.e. those highly willing to resign against those less willing to resign. Their main results are that people who traveled frequently, are single, have a low income, have longer commutes and/or work in a human resource department are more likely to leave their job.
Guerranti (2021), as Yang and Islam (2021), put the focus on understanding what causes employees to quit their jobs. On the data description, the use is made of a relative Gini Heterogeneity index to describe the categorical variable and account for the frequency of their classes. Once the

data description and the preprocessing are done, the paper relies on five classification models : a logistic regression, a classification tree, a random forest, a naïve bayes classifier and a neural network. Using metrics to account for the prediction accuracy, the naïve bayes classifier is excluded first because the accuracy is lower than 75%. Then using the ROC curve the Random Forest and Logistic regression are considered the better predictors. Moreover, these two estimation methods allow us to account for the weight of the variables in the prediction. The Random Forest seems to be better in this situation since the Linear Regression strangely excludes the monthly income.

To perform an exploratory analysis of our data, we will use a traditional KMeans clustering, coupled with dimensionality reduction techniques due to our high number of variables compared to our relatively low number of observations. In *"K-means Clustering via Principal Component Analysis"*, **C. Ding and X. He (2004)**. They used PCA on gene expressions of tissue samples and performed KMeans clustering on the 5 first principal components. They obtained a satisfying clustering accuracy of 0.875. In *"Using [...] t-SNE for cluster analysis and spatial zone delineation of groundwater geochemistry data"*, **H. Liu al.(2021)** used the t-distributed Stochastic Neighbor Embedding (t-SNE) method as a tool for cluster analysis to understand spatial patterns of groundwater geochemistry. t-SNE allowed them to choose the number of clusters and to assign clusters to the data. They also used PCA which was outperformed by t-SNE. They determined the perplexity parameter of the algorithm empirically. For our own study, we will use PCA and t-SNE to perform dimensionality reduction and data visualization, and apply the KMeans algorithm on the new data.

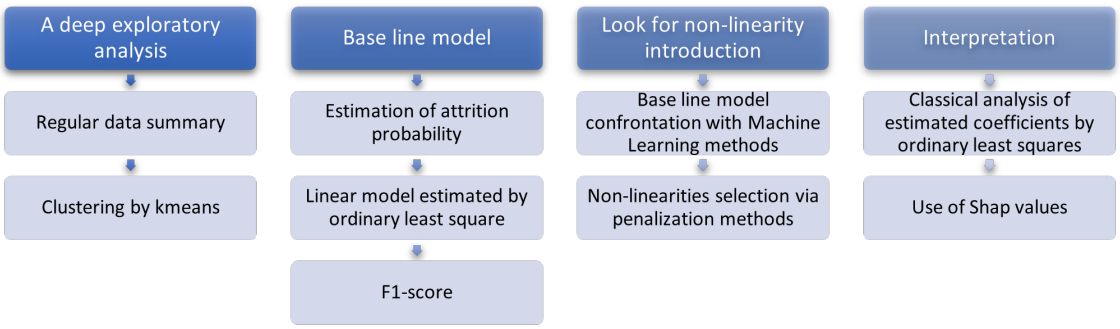
From our literature review, we did not find any paper that addresses attrition using automatic selection model methods. Therefore, we will directly apply the methods found in theoretical papers. **Doornik Hendry (2015)** proposes to account for non linearities in a model by creating new variables from the existing ones. In the case of cross-section data, this will consist of taking the variables to the square, to the cube and to the exponential. We will inspire ourselves from what they have done, by creating those variables and adding them to our database.

To have an interpretation and determine the importance of our variables, we will use the SHAP (SHappley Additive exPlanation) values. In the paper A Unified Approach to Interpreting Model Predictions (**Scott Lundberg Microsoft 2017**), the authors present the seminal work about shap values and how it works in theory. For our application, we will pre-select the variables and estimate a parsimonious model to finally use the shap value to understand why individuals leave the company.

1.3 Method summary

To define the best predictive model that can be interpretable, we proceeded with the following steps :

FIGURE 1 – Methodology path



The first step consists in a deep exploratory analysis of the data ; thus it allows us to have a clear overview of the composition of the dataset and make first assumptions about what our study will show. In particular, the use of dimension reduction techniques coupled with k-means clustering methods is a strong tool describe already well the behavior of individuals regarding attrition. Once this step is done, the goal is to model attrition to predict it and be able to define which features have importance on the attrition decision. To that extent, we first define a baseline linear model estimated by ordinary least squares. The idea is to use this first estimation as a benchmark to assess our following work, the goal being to improve the probability prediction of attrition. To compare our models, we use the F1-score as our dataset is strongly unbalanced.

One main concern is that this first linear modeling does not capture nonlinear behavior. The non-linearities study was divided into two parts : first test for the need of non-linearities introduction, and then introduction of non-linearities. Thus, to test for the need of non-linearities introduction, we confront our baseline model with machine learning algorithms that are not constrained by modeling and take potential non-linearities into account. We first struggled to define accurate machine learning methods because of the dataset size which is rather small, however we finally found out that support vector machines and extreme gradient boost are adapted algorithms to deal with small datasets ; these algorithms succeed to outperform our baseline model. Since we were able to find such machine learning algorithms that lead to better classification (i.e better F1-score), the introduction of non-linearities was needed. To this purpose we introduce non-linearities in mass and then process automatic features selection via penalization methods : ridge, lasso and general to specific procedure via autometrics.

Once we have a well designed non-linear probability model, we estimate it via ordinary least squares and SVM to assess which method reaches the best metric and thus the best prediction. Finally, we study the interpretability of the estimation : What does impact the attrition decision ? To this purpose we introduce the use of Shap values that mainly allows to have a relative importance of each variable.

1.4 Key results

We began our study with an exploratory analysis of our data using clustering methods. With the help of the elbow method, we defined our optimal number of clusters for our data which is 4 and we ran a KMeans algorithm. Due to the high dimensionality of the data (50 dimensions), we obtained mixed results with a silhouette coefficient of 0.01, which means that our clusters overlap. We then ran a Principal Component Analysis after determining that the 10 first principal components explain 50% of the variance. After running KMeans on the new 10 dimensions dataset, we obtained better results, but not yet satisfying. Finally, we used the t-Stochastic Neighbor Embedding tool to reduce our data to 2 dimensions and ran another KMeans algorithm with 4 clusters. We obtained satisfying results from our evaluation scores. Finally we analyzed the employees in each of the 4 clusters. The first cluster is composed of human resources jobs and has the highest number of females across the 4 clusters. The second cluster contains the sales jobs and displays the highest rate of attrition. The third cluster has only technical employees, mostly males that are underpaid compared to the other clusters. It also shows a high attrition rate. Finally the last cluster is mainly composed of managers, the oldest employees, which have the best incomes and the lowest rate of attrition. These results are satisfying because they are coherent and make sense.

In the second part, we tried to predict if an employee will leave the company or not. To do so, we first tried to determine if there are non linear relationships between our variables. We began by establishing a baseline OLS model and compared the results with different machine learning models like XGBoost our Support Vector Machine. The two models obtained better results of F1-score than the OLS model, which means that they have captured non linearities when the OLS model couldn't. We then added the squares and cubes of our variables in the dataset and tried penalization methods such as Ridge, Lasso and the GETS procedure with Autometrics to see which variables were kept by the models. We compared the variables kept in each method and concluded that variables such as age or monthly income are important in attrition, as well as non linear variables such as the square of the number of years with the current manager.

Lastly, we decided to interpret our significant variables with the help of Shap values. We obtained the relative importance of each variable on attrition. The number of years with the current manager and the age of the employee have the greater impact on attrition, with young employees who have kept the same manager for a short amount of time being the ones who are more likely to leave. Finally, we displayed the impact of the main variables on the attrition probability of one employee which is interesting from a business point of view.

2 Materials and methods

2.1 Data description

Our dataset comes from the website Kaggle. IBM released the data set in 2015. This data set is "based off real data with all personal identifiers removed," but was "also tweaked so that it performs better in telling a story about attrition", the dataset presents itself as a record of IBM's human resources employee attrition and performance. Following the information provided by **Frye et all. (2018)**, this data was provided as part of an IBM Watson Analytics promotion to push their new analytics platform. The data set was provided by IBM as a sample use-case scenario in which they identified the primary features correlated to attrition as well as determined the attrition rate for several demographic categories of employees. In our search phase, we have uncovered several articles and papers using this same dataset to predict employee attrition. Based on

this hypothesis of common approval of the data, we have decided to use this dataset for our project.

2.2 Methods

2.2.1 Methodology : general modelization and evaluation metrics

The goal of our study is to be able to predict if an employee will leave or not his job. $y = 1$ means that the attrition occurs and the employee will leave the company. To be able to implement the methods studied in class, we will use a probability model. Thus, the output predicted $\hat{Y}$ is a vector of probability that employees will quit. Each probability of the vector corresponds to an observation, thus if the third probability of the output vector is 0.33 it means that the observation / employee three of the database has a 33% chance to leave his job.

In a more formal way, the model to optimize is the following one :

$$\begin{cases} P(Y = 1|X) = \beta X + e \\ stmin \sum (Y - \hat{Y})^2 + \sum_i (\beta_i)^2 \end{cases} . \tag{1}$$

Thus the expected betas are computed as follow :

$$\hat{\beta}^{ols} = (X'X)^{-1} * X'y$$

We need to choose a metric to evaluate our different models and compare them across the study. We have seen that our dataset is unbalanced, which means that our classifier will not succeed very well at predicting the least distributed class. Using accuracy as a metric will lead to biased results. Indeed, since it is computed as the ratio of correct predictions to the total number of predictions, we will obtain high accuracy scores when in reality the model hasn't predicted well the minority class. We need to find another metric.

The F1-Score is a well-known metric when it comes to unbalanced dataset. It is computed by looking at the precision score and the recall score. Precision is computed by looking at how many positive predicted observations are correctly classified. A model with a high precision score can miss some true positive observations, but among the ones that are classified as positive, there are almost no mistakes. Recall can be seen as the opposite of precision. It is computed by looking at how many positive observations have been predicted on the total of positive observations. A model with high recall will find all positive observations in the data, but may classify negative observations as positive.

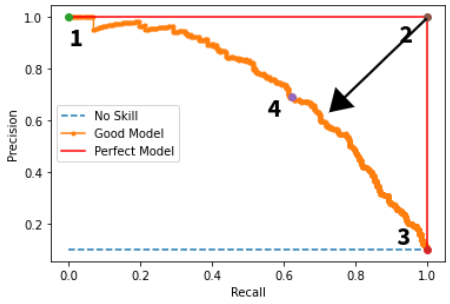
There is a trade-off between precision and recall. A model with higher precision will have lower recall scores and vice-versa. The F1-Score can then be defined as the harmonic mean between precision and recall, giving the same weight to both scores. It is formally defined as follow :

$$F1_{score} = 2 * \frac{(Precision * Recall)}{(Precision + Recall)}$$

Since our variable Attrition has only a few employees who left the company, we will use the F1-Score to compare our models instead of the traditional accuracy.

However, since our output is a vector of probability and not a vector of dummy variables, we need to find a way to use an accurate threshold level to transform the vector probabilities to dummy.

FIGURE 2 – Threshold selection by precision-recall curve



A classic choice is to choose 0.5 has threshold such as :
— if $\hat{y} <= 0.5$ then $\hat{y} = 0$
— if $\hat{y} > 0.5$ then $\hat{y} = 1$
But since our dataset is strongly imbalanced, this leads to poor interpretations of the predicted probabilities.

A first usual way of tuning a threshold is to use the receiving operator curve, but we decided to base our interpretation of prediction on the F1-score, thus the best way to optimize the threshold seems to rely on the optimization of this F1-score using precision-recall curve. This allows us to focus on the performance of our classifier on the positive class, that is the minority one.

A precision-recall curve is plotted by creating probability predictions across a set of thresholds and computing the precision and recall for each threshold. A line plot is created for the thresholds in ascending order with recall on the x-axis and precision on the y-axis. Thus each point on the line assesses the prediction of a certain threshold. Then the goal is to choose the threshold that optimizes the precision-recall trade-off i.e that optimizes the F1-score. Visually, the goal is to find a point the closest possible to the right-top corner i.e that gives the best precision and recall and thus a F1-score that tends to 1.

In other words, for each models seen in the study, we will plot the precision-recall plot to optimize the f1-score threshold in order to be able to compare our model relying on a metric that makes sense and take into account the fact that the dataset is strongly unbalanced.

2.2.2 Clustering methods : a tool for a deep exploratory analysis

KMeans is a clustering algorithm belonging to unsupervised learning methods which aims to create a specified number of groups among the observations. The KMeans algorithm works iteratively by computing distances between observations and cluster centroids. We could resume its process with these steps :

— First we choose a number of clusters k for the algorithm
— Then the algorithm creates n centroids in the space
— Then each observation is assigned to its nearest centroid by computing the euclidean distance between the two points
— Once all the observations are assigned, the algorithm computes new centroids based on the average euclidean distance between all the observations of a given centroid
— Then each observation is assigned again to its nearest centroid among the new ones

— The process continues until the assignment is stable

An efficient clustering is a segmentation where observations within a given cluster are very similar while observations across clusters are very different. There exists different scores which allow us to evaluate an unsupervised clustering. For our study, we will use the Silhouette coefficient, the Calinski-Harabasz index and the Davies-Bouldin index.

The Silhouette coefficient is composed of two scores :
— The mean distance between an observation and all other observations in the same cluster
— The mean distance between an observation and all other observations in the next nearest cluster

The Silhouette coefficient is computed for each observation and the global coefficient is the mean of all coefficients. The score is bounded between -1 (incorrect clustering) and 1 (perfect clustering). If the score is around zero, it means that clusters overlap.

The Calinski-Harabasz index (also known as the variance ratio criterion) is the ratio of the sum of between-clusters dispersion and within-cluster dispersion for all clusters, where dispersion is the sum of distances squared. A higher score indicates better clustering with well separated clusters.

The Davies Bouldin index looks at the average similarity between clusters, where similarity is a measure which compares the distance between clusters with the size of the clusters. The score is bounded between 0 and $+\infty$ with 0 indicating a perfect partition.

The KMeans algorithm is based on the computation of euclidean distances. As the number of dimensions increases, distances become more and more complex to apprehend which can cause difficulties to create clusters of observations. When there are not enough observations, data becomes sparse and distances become similar. This problem is known as the curse of dimensionality.

Our dataset contains only a few observations (1470) and a relatively high number of variables (50). We don't have enough observations to obtain satisfying clustering results in 50 dimensions, so we need to use dimensionality reduction techniques to facilitate our clustering task.

Principal Component Analysis is dimension reduction technique, it allows us to represent our dataset in a reduced dimension while capturing maximum variability of this dataset.
Indeed, some of our k variables may have similar behavior meaning they change together and are collinear. Our goal is to find a few values, the principal components, to represent the data set and get rid of collinearities between variables. In other words, from the k-dimension space, we would like to find a hyperplane of lower dimension to approximate the initial dimension. Principal component approximation is only a linear combination of variables, so we totally ignore potential non linearities in our representation.

The T-Distributed Stochastic Neighbor embedding (or t-SNE) is a visualization tool that can be used for dimensionality reduction, and which can take non linear relationships between variables into account. After computing a similarity measure based on distance between each pair of observations, it converts the measure into a probability. Two close observations will then have a high probability. The method projects the observations in a low dimensional space and compares the similarity probabilities to create clusters.

2.2.3 Machine learning methods : assess the existence of non-linearities

Parametric econometric assumes that the data come from a generating process that takes a well-defined form such as $y = \beta X + \epsilon$. However, we know that it usually has a non-linear relationship between the dependent variable and the explanatory ones. Keeping the econometric modeling defined in a parametric way can lead to missing important information. Machine learning methods do not make any assumption on how the data have been generated, the processing can be described as $y = m(X)$. Thus, machine learning models are non-parametric and do not have any constraints on the modeling form of the data. It is supposed to take every possible behavior between the dependent variables and the explanatory ones into account.

The goal here will be to define a baseline model using OLS to predict the probability of attrition, then use several machine learning methods that deal with our specific data and assess if we need to introduce behaviors in our parametric models so that it can perform better. In other words, if at least one machine learning method reaches a better estimate of the attrition probability, then we need to re-specified the baseline model.

One major point to focus on to define the machine learning models to use is the kind of data we are dealing with. The dataset is first of all unbalanced, as shown previously, but is also quite small. For the unbalanced part, we decided to not rebalance it but use a metric that allows us to take this situation into account. Then, to deal with small dataset, and even smaller training dataset, it is important to use what is generally called as "simple machine learning methods'. Indeed, with few data methods with too much complexity will not perform well because the amount of different situations and data to train on will not be enough. Following recommendation and literature, we decided to use a non-linear **Support vector machine** and an **eXtreme Gradient Boosting** ensemble learning.

The objective of ensemble methods is to combine the predictions of several estimators built with a given learning algorithm in order to improve the generalizability/robustness compared to a single estimator. This method increases the performance and stability of the model, minimizes its variance and achieves a level of accuracy much higher than what would be achieved by using any of these models separately. There are generally two families of ensemble methods :

- In averaging methods, the guiding principle is to construct several estimators independently and then average their predictions. On average, the combined estimator is generally better than any single base estimator because its variance is reduced.
- In contrast, in boosting methods, the basic estimators are constructed sequentially and an attempt is made to reduce the bias of the combined estimator. The motivation is to combine several weak models to produce a powerful set.

Boosting will produce models that are highly dependent on each other, unlike the bagging principle. The first step is to create a first basic model from a chosen algorithm. It is trained on the data. At the beginning, all observations are given equal weights. From the results obtained from this model, if an observation is misclassified, it increases its weight.

Then, a second model is built to try to correct the errors present in the first model. It is trained using the weighted data obtained in the first step. This procedure continues and models are added until the entire training data set is predicted correctly or the maximum number of models are added. The predictions of the last model added will be the overall weighted predictions provided by the previous tree models. (Figure n°2).

Gradient boosting is a special case of boosting where errors are minimized by the gradient descent algorithm. The idea is to correct the predictions for which the residuals are high, and not

to touch those with low residuals. In other words, we only want to correct for the observations that have been predicted incorrectly. And this is precisely what the next decision tree will do : it will compensate for the errors committed previously without deteriorating the predictions that were correct. To do this, the learning base is modified between two trees. Instead of predicting y, the second tree should predict $r = y - \hat{y}$, because by summing the previous prediction plus the second tree, we should get the answer y. On the other hand, if the prediction was already correct, then $r = y - \hat{y} = 0$, and so the second tree will predict a very small value, which will not change the value already predicted. The eXtreme gradient boosting (XGBoost) is a machine learning model based on sequential ensemble learning and decision trees. Let's detail the algorithm behind this model. Consider an initial weak classifier. After optimizing it, the boosting method seeks to build a new weak classifier f_1 from f_0 by introducing a residual term h :

$$f_1(x) = f_0(x) + h(x)$$

$$h(x) = \eta + \frac{\sum_i residuals_i}{f_0 + +(1 - f_0)}$$

So that f_1 is more efficient than f_0 . Repeating the operation a number of times, say p, we construct a final complex classifier that is a linear combination of f_i , where each of the is associated with a weight α_i :

$$F(x) = \sum_{i=1}^{n} \alpha_i f_i(x)$$

In practice, we use the XGBClassifier from the xgboost library. There are a large number of parameters that can be tuned. For our study we followed this procedure :

- Step 1 : Define a rather large learning rate, here we choose $learningrate = 0.5$
- Step 2 : optimize by cross validation the depth of trees with **max_depth**, the cover, that represents the minimum threshold for the number of individuals present in a node, with **min_child_weight** and then two penalization regularization parameters use in the computation of the similarity score for the leaf construction of each tree, named **reg_alpha** and **reg_lambda**.
- Step 3 : Use the results from step 2 in the classifier and now optimize by cross validation **gamma** the parameter that allow to regularize the depths of the trees, the higher the values of this hyperparameter, the more the penalty prevents to build trees too deep if the performance contribution is not so high.
- Step 4 : Since we have a rather small dataset, we test for the increase of the number of trees and the reduction of the learning rate. Once more this is done by cross validation to optimize **n_estimators learning_rate**.

Our final hyper-parameters choice for the implementation of XGBoost are stored in the table n°1.

TABLE 1 – Hyper-parameters for XGBoost

Parameters	Description	Grid	Final choice
max_depth	Depth of trees	[2,3,4,5]	2
min_child_weight	Cover	[0,1,2,3,4]	3
reg_alpha	Similarity score penalization (L1)	[0.1,0.5,0.7]	0.7
reg_lambda	Similarity score penalization (L2)	[0.1,0.5,0.7]	0.7
learning_rate	Step size shrinkage used in update to prevents overfitting.	[0.001, 0.01, 0.1,0.5]	0.5
gamma	Minimum loss reduction required to make a further partition on a leaf node of the tree	[0,0.5,1,1.5,2]	2
n_estimators	Number of trees (boosting steps)	[100, 200, 300, 400, 500]	100

Support vector machine allows to deal with nonlinearities using polynomial functions and interaction terms between predictors.The idea of the support vector machine is to fit a separating hyperplane in a space with a higher dimension than the predictor space. Instead of using the set of predictors, the idea is to use a kernel. The support vector linear classifier can be defined as follows :

$$f(x) = \beta_0 + \sum_i \alpha_i < x, x_i >$$

With $< x, x_i >$ the inner products between all pairs of training observations. With the SVM the idea is to replace this inner product by a generalized form using kernel.

$$f(x) = \beta_0 + \sum_i \alpha_i K(x, x_i)$$

The goal of the support vector machine is to find a new feature space where the data can become linearly separable as with the initial support vector classifier. The so-called kernel has a goal to define this new feature space, the main goal is to find the kernel that better describes our data distribution. For example in figure n°3, the data distribution is clearly not linearly separable in the input space because blue points encircle the red one. Using a non-linear kernel allows defining the new feature space that is shown and this time the data are linearly separated.

Our final hyper-parameters choice for the implementation of SVM are stored in the table n°2.

TABLE 2 – Hyper-parameters for SVM

Parameters	Description	Grid	Final choice
Kernel	Define the new feature space	['rbf','poly','sigmoid']	Sigmoid
Gamma	Sensitivity parameter to differences in features vectors (RBF & Sigmoid)	[1, 0.1, 0.01, 0.001, 0.0001]	0.01
C	Regularization parameter, squared l2 penalty	[0.1, 1, 10, 100, 1000]	0.1

2.2.4 Methods to determine non-linearities

The goal of adding non-linearity is to capture in a more fine way the behavior of the true data generating process. Our methods consist in adding the square and the cube [1] of each variable to our dataset, this considerably increases the number of explanatory variables. Then we use penalization methods to automatically select the variables that do affect the dependent variable that we want to predict : the attrition.
Thus our new features matrix is denoted as : $X_{nl} = X + X^2 + X^3$

Penalization is used to reduce the variance that appears when the number of features is too large. It leads to high variance for two mains reasons :

— A number of observations smaller than the number of features

— High correlation between features

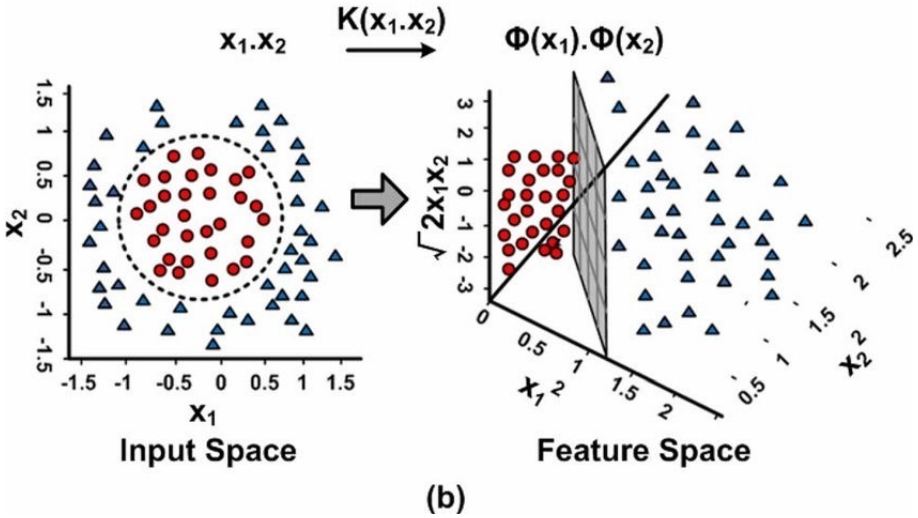
This also leads to overfitting, the model learned to be close to the train data and thus gives poor prediction. Adding a penalty will allow us to find a regression line that can give more accurate predictions.

The goal of penalizing a model is to have a trade off between variance and bias, introducing a little bias by reduction method in the estimate might lead to a substantial decrease in variance and thus a decrease in prediction error term.

As more terms are included in the model, coefficient estimates suffer from higher variance. Introducing a little bias in our estimates might lead to a decrease in the variance and hence decrease the prediction error.
Thus the main problem is the rank deficiency as the number of terms in the model increases, it leads to multicollinearity and increases the variance. The idea is to select the relevant variables among the set of covariates. One solution is to use regularization methods such as Ridge and Lasso. A high variance introduces wrong prediction, this is called overfitting. This there is a too high variance, the idea is to find a new regression line using a penalization term λ by introducing some bias.

1. adding cross-product and exponential has led to a too big database to compute all the functions needed, howether, with more time and work on the subject it could be interesting to add them.

FIGURE 3 – SVM illustration



Remark : Another use of the regularization is that it can be used to estimate models where the number of observations is less than the number of features. Indeed, with OLS it is impossible to invert $X'X$ but with the penalized term, the estimation of beta becomes $\hat{\beta} = (X'X + \lambda I_n)^{-}1X'y$ and it becomes invertible.

We try three different penalization methods : Ridge / Lasso / Autometrics procedure

Ridge and Lasso procedure rely on the same idea : adding a penalization term i.e. some bias to the estimate on the minimization of the sum of squared residuals. This allows to tend some estimation to zero and reduce the variance. The greater the penalization term, the closer to zero tend the estimates but it will never reach zero.

Two importants properties are :
— if $\lambda = 0$ then the penalization term is inefficient and the situation is the same as a classical OLS
— if $\lambda = \infty$ then β tend to 0 for Ridge or is set to zero for Lasso
In a more formal way, the model to optimize is the following one :

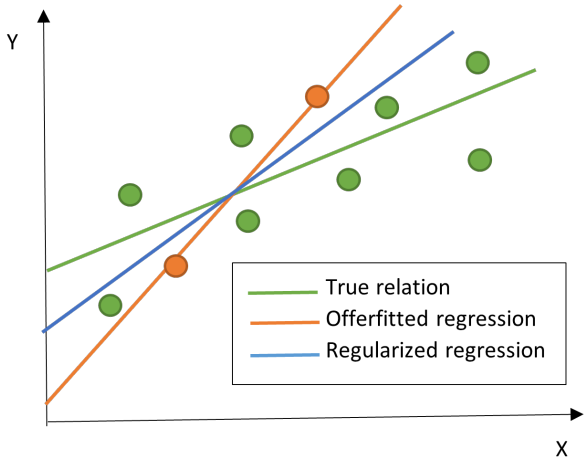
$$\begin{cases} P(Y = 1|X_n l) = X_n l + e \\ stmin \sum (Y-)^2 + \lambda \sum_i (\beta_i)^2 \end{cases} . \tag{2}$$

Thus the expected betas are compute as follow :

$$\hat{\beta}_{\lambda}^{ridge/lasso} = (X_n l' X_n l + \lambda I_n)^{-1} * X'y$$

The goal is to find the optimal λ that will optimize the model and give the best prediction. Once it is done, the selection is made looking at the estimated coefficients. These coefficients cannot be interpreted because they are biased by the penalization, however the closer to zero it gets the less importance it has. Thus, we can select only the variables that have an estimated coefficient greater than a certain level.

FIGURE 4 – Penalization visualisation



The third technique employed to select variables is an implementation of the GEneral To Specific (Gets) procedure called Autometrics. First, We define a starting model with all variables that we have. This is the General Unrestricted Model (GUM).
The goal is to select the best possible model from the variables available to us. Model selection is an iterative search procedure, i.e. a tree search with multiple paths to select a parsimonious model. Each insignificant variable found reduces the number of paths possible. The first path is chosen by deleting the most insignificant variable, the one with the lowest t-values. This path continues by deleting the least significant variable. The model is re-estimated each time. The path terminates when all variables are significant. This is a GEneral To Specific (Gets) procedure. On top of that,the particularity of Autometrics comes from the fact that the GUM model must be congruent and terminal nodes must satisfy diagnostic tests that are only evaluated when a terminal is reached.

Autometrics is an implementation of the Gets procedure. The starting point for Autometrics is the entire space of models generated by all the variables in the initial model. Every node of the tree is a model which can be estimated. The algorithm can be seen as a data mining procedure as the knowledge of the analyst doesn't intervene in the process of selecting the relevant variables.

For p insignificant variables, there are 2^p possible models. Visiting each node and estimating every model is not possible. The multiple-path search is an unstructured way of searching the model space, meaning that in all possible sets of the variables in the GUM : many paths may turn out to be the same, while other paths are left unsearch. The procedure only estimates unique models before estimating them.

The main benefit of the Gets procedure is the control for the influence of other variables while estimating the models.Another advantage is that the procedure results in a parsimonious model which is useful in our case to give an interpretation to the model. However, the procedure has tendencies to retain some irrelevant variables, the more correlated the variables, the higher the tendency.

2.2.5 Interpretation methods

Understanding why a model makes a certain prediction can be as critical as the accuracy of the prediction in many applications, such as employee attrition in our case.
The accuracy of the prediction in many applications, such as employee attrition in our case. Indeed, as economists, we would like to have a good prediction of attrition but also a good understanding of the characteristics of individuals that are important in determining whether or not they leave the company.

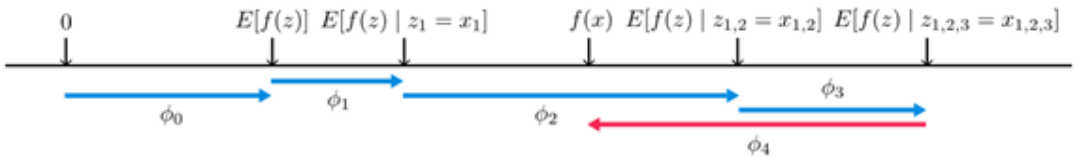
The problem is that often, a better accuracy means a more complex model like machine/deep learning models. There is a trade-off between accuracy and interpretability.
The method that we use here is SHAP (SHappley Additive exPlanation) which is based on Game Theory.. It assigns each feature an importance value for a particular prediction.
There already exist different methods (LIME, DEEPLIFT, Layer-Wise Relevance Propagation etc) but Shap is a unified one and works in many contexts.

The main specificities of the SHAP results are the following ones :

— Game theory guarantees a unique solution
— Shap is better for human understanding than other methods
— Explainability of any model prediction is a model itself, we can call it an explanation model

The figure n°5 shows the sign and importance for each feature in the process of the prediction. Note that the order of the variable is important here. For instance, the added importance of the second variable knowing the first one is represented by phi2 is positive and has a higher impact in the prediction process than the first variable.

FIGURE 5 – Illustration of Shap results



3 A global study of attrition decision in employment

3.1 A deep exploratory analysis

3.1.1 Classical descriptive statistics

The dataset contains 1470 observations for 35 variables. Each observation is a fictional employee of IBM. Here's the glossary of the variables (Glossary)
Among the 35 variables, some of them don't bring any information to our study. Indeed, the variables "EmployeeCount", "Over18" and "StandardHours" contain only one value. Since these variables are constant, they can be removed from the model. Furthermore, the variable "EmployeeNumber" attributes a different number to each employee which doesn't contain any information on the employee. We can also remove this variable.
After removing these 4 variables, we can compute some descriptive statistics about our data.
The figure n°1 displays the distribution of our explained variable : "Attrition". As we can see, our dataset is imbalanced. Indeed, 83.88% of the employees haven't left the company, which can cause biased accuracy results. Since the employees who leave the company are under-represented in our data, we will need to be cautious in the analysis of the results of our models. We can recode the variable to be numerical instead of a string with 0 and 1

On figure n°5 it can also be read that 2 out of 3 employees at IBM work in the "Research Development" department according to the distribution of the variable Department. The "Human Resources" department contains less than 5% of the employees while the others work in the "Sales" department.
Moreover, the distribution of the variable BusinessTravel tells us that the vast majority of IBM employees travel rarely. Less than 20% of them travel frequently while 10% have never traveled.

With figure n°6, we can see from which education fields IBM employees come from. It seems that the vast majority of them have studied in "Life Sciences" or "Medical" fields. On the contrary, the least studied field is "Human Resources". These results are coherent with the distribution of the variable Department that we saw earlier.
Then it also displays the distribution of the variable JobRole. As we can see, most employees work as "Sales Executive", "Research Scientist" or "Laboratory Technician", which is consistent with our previous results. The job role least distributed is "Human Resources".

On figure n°7, the pie graphs show the distribution of the variable Gender. It seems that 60% of the employees are males, while 40% are females.The distribution of the variable MaritalStatus allows us to know that 46% of the employees are married. 32% of them are single while the least distributed category is "Divorced" with 22%. And finally, it displays the distribution of the variable Overtime. More than 70% of the employees work overtime in the dataset.

To describe the *mean* employee, we use the descriptive statistics (table n°17 in annexes). Thus, on average, an employee is 37 years old at IBM. The youngest employee is 18 years old while the ol-

FIGURE 6 – Some important variable distribution

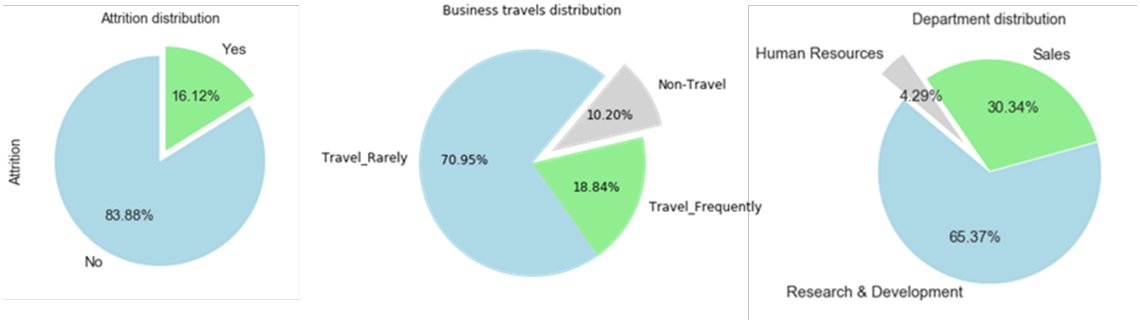
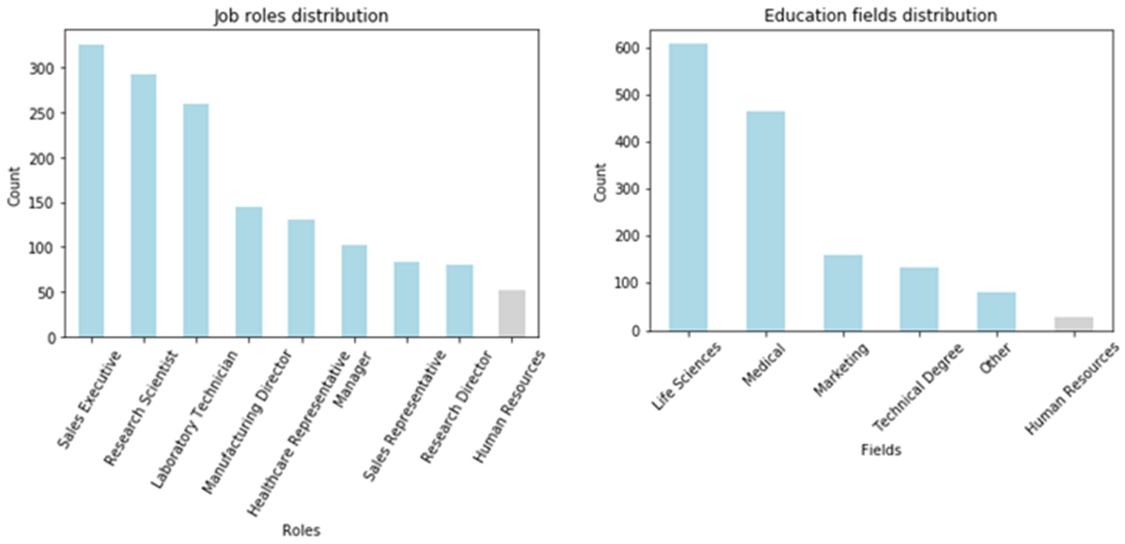


FIGURE 7 – Some important variable distribution



dest is 60 years old. 75% of the employees have a level of education superior or equal to college. The average monthly income is 6503 dollars in the dataset while 25% of the employees have a monthly income superior or equal to 8379 dollars. 50% of the employees have worked in 2 companies or more before entering IBM. 25% of the employees have worked less than 6 years in their life, with the maximum number of years spent working being 40. On average, the employees have spent 7 years working for IBM and stayed 4 years in their current job at the company. 75% of the staff haven't had a promotion in 3 years and 25% of the employees have kept the same manager for 7 years or more.

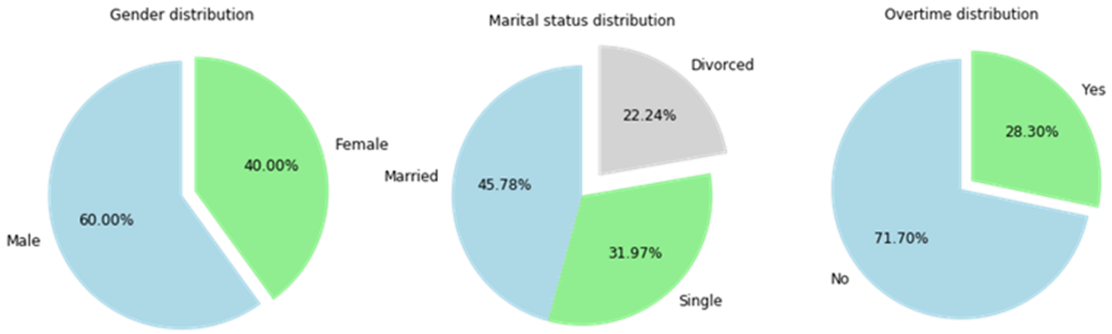
TABLE 3 – Variance Inflation Factor

Feature	**VIF**
monthlyincome	18.1224
joblevel	14.1404
totalworkingyears	5.0210
yearsatcompany	4.7648
yearswithcurrmanager	2.8586
yearscurrentrole	2.8066
percentsalaryhike	2.5651
performancerating	2.5509
age	2.0934
stockoptionlevel	1.9159

To prevent having multicollinearity in our model, which can lead to misleading results, we can compute the Variance Inflation Factor (VIF) of our variables. This metric measures how much the variance of a variable increases because of collinearity. In principle, we can use a rule of thumb which tells us that each variable with a VIF superior or equal to 2 is considered as a factor of multicollinearity in the model. As we can see it in table n°3, there are several variables that cause multicollinearity, with MonthlyIncome and JobLevel being the two most colinears. Fortunately, the three methods that we'll use in this report (PCA, Ridge, Lasso) can be used with multicollinear variables.

Before testing models on our dataset, we need to do some preprocessing steps. First, we cleaned

FIGURE 8 – Some important variable distribution



special characters from strings and put the characters in lowercase. Then, we encoded the string variables of the database with one-hot-encoding. The variables concerned are : Attrition (y), Overtime, Gender, Businesstravel, Department, EducationField, JobRole and MaritalStatus. This brings us to a total of 49 variables. Finally, since most of our features are spread on different scales, we'll use a Min-Max-Scaler to normalize the values to the same range.

3.1.2 Common patters define by clustering methods

Now we will use unsupervised learning methods to uncover subgroups in our dataset. Indeed with clustering methods, we can define categories of employees which have several characteristics in common. For example, we could imagine a category of employees which are older on average and have a greater salary than others. Maybe the individuals in this group are also less likely to quit the company because of their status.

For this exploratory analysis, we will use a KMeans algorithm coupled with dimensionality reduction techniques to organize our data in a defined number of clusters. To do so, we will proceed in several steps :

- Determine the optimal number of clusters to create segment of employees with the elbow method
- Apply a benchmark KMeans model to see how the algorithm performs with our dataset
- Transform our data with dimensionality reduction techniques (PCA and t-SNE)
- Apply KMeans on the new low dimensional spaces
- Compare the three methods with our cluster evaluation scores
- Choose the best model and describe the employees inside each clusters

The elbow method allows us to get the optimal numbers of clustering for our data. It consists in running the KMeans algorithm for a range of cluster values and computing the sum of squared distances from each point to its centroid at each iteration of KMeans. We can then display the sum of squared distances for each number of clusters and select the lowest sum of squared distances for the lowest number of clusters (the elbow of the curve). A lower sum of squared distances corresponds to a better similarity of observations in each cluster.

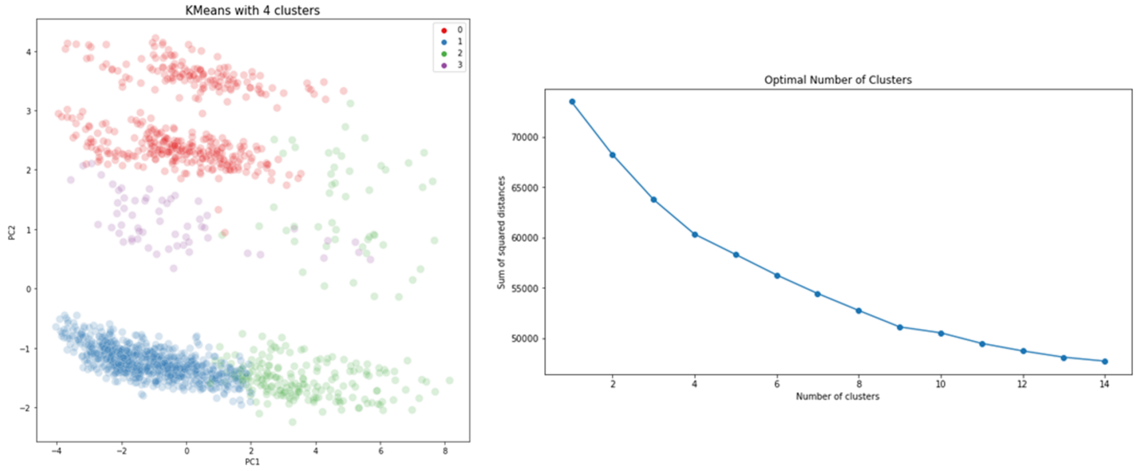
By running the elbow method on our data, it seems that 4 would be an optimal number of clusters for creating subgroups of employees.
After running KMeans with 4 clusters, we get the clustering scores registered in table n°4. While we can't compare these scores yet, the Silhouette coefficient indicates overlapping clusters. This is not a surprising result given the high dimensionality of our data.

Table 4 – Clustering scores

Silhouette coefficient	Calinski-Harabasz index	Davies-Bouldin index
0.1	95.5	2.3

We can use the Principal Component Analysis method to reduce our data to 2 dimensions to plot the result of our clustering. As we can see on figure n°9, it seems that the first two clusters are well segmented while the two others are more sparse.

FIGURE 9 – Optimal number of clusters - original dimension



Now we will use PCA to reduce our 50 variables to only a small amount of principal components. We can determine the optimal number of components by plotting the amount of variance explained by each component. Looking at figure n°9, we can tell that the 20 first principal components explain approximately 70% of the variance. We can then apply PCA with 20 principal components and use these data to perform another KMeans clustering.
Here again, it seems that 4 is the number of clusters which minimize the sum of squared distances and the number of clusters. Overhaul, it seems that the PCA dimensionality reduction has slightly increased our clustering performance, table n°5. However, our results are not satisfying yet so we will try another dimensionality reduction technique.

TABLE 5 – Clustering scores with PCA

Silhouette coefficient	Calinski-Harabasz index	Davies-Bouldin index
0.14	163.1	2.0

FIGURE 10 – Optimal number of clusters with PCA

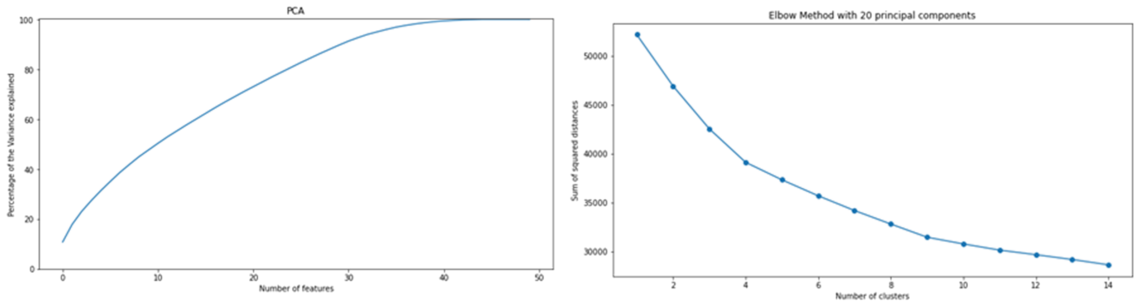
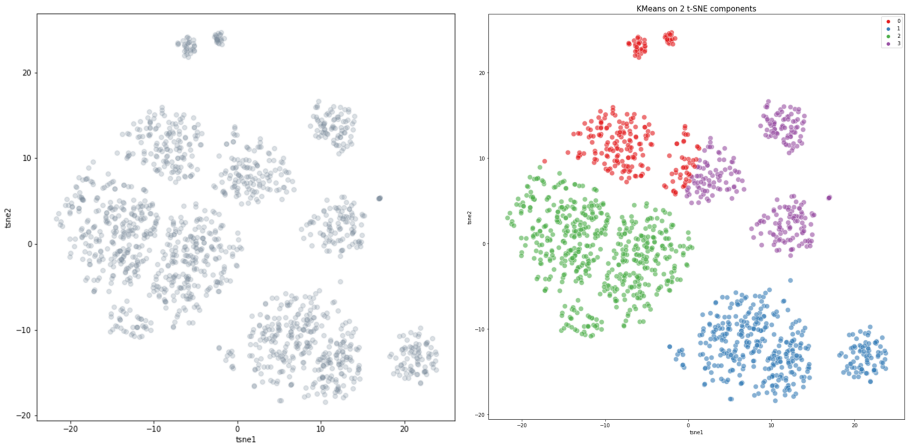


FIGURE 11 – Optimal number of clusters with t-SNE



Now we use a t-SNE algorithm to reduce the dimension of our data to 2 components. After tuning the perplexity parameter (the number of nearest neighbors taken into account in the algorithm).

With this method, it's much easier to determine clusters with a human eye so we should logically obtain better results after running KMeans. For the sake of the study, we will again choose 4 clusters for this method, even if the elbow method tends to 3. After running KMeans on the two t-SNE components with 4 clusters, we obtain the results present in table n° 6 and that can be visualised in figure n° 11. This time, the KMeans algorithm gives an almost perfect clustering with no overlapping except for the first and last clusters.

TABLE 6 – Clustering scores with t-SNE

Silhouette coefficient	Calinski-Harabasz index	Davies-Bouldin index
0.48	1876.34	0.76

We can conclude by saying that t-SNE has allowed a significant improvement in our clustering results with the KMeans algorithm. The two methods combined have obtained the highest Silhouette coefficient and Calinski-Harabasz index, as well as the lowest Davies-Bouldin index.

TABLE 7 – Overview of clustering scores

	Silhouette coefficient	Calinski-Harabasz index	Davies-Bouldin index
KMeans	0.01	106.7	2.5
PCA KMeans	0.21	286.6	1.5
t-SNE KMeans	**0.48**	**1876.34**	**0.76**

Table 8 – Observations distribution accross cluster

	Cluster 0	Cluster 1	Cluster 2	Cluster 3	All data
Mean attrition	10.2%	**22.0%**	19.78%	**4.9%**	16.12%
Mean age	37.78	35.56	**34.17**	**43.95**	36.92
Proportion of males	**56.73%**	58.43%	**63.88%**	57.36%	60.0%
Mean monthly income	6572.46	6052.01	3238.65	**13921.82**	6502.93
Mean job level	2.21	2.07	**1.22**	**3.66**	2.06
Mean career length	11.72	9.8	**7.69**	**20.63**	11.28
Mean years at company	6.67	6.57	**5.07**	**12.03**	7.01
Mean years in current role	4.23	4.28	**3.24**	**6.21**	4.23
Mean years since last promotion	1.73	2.19	**1.47**	**4.10**	2.19
Mean years with current manager	4.24	4.07	**3.25**	**5.91**	4.12
% of HR jobs	**21.22%**	**0.0%**	**0.0%**	4.15%	4.28%
% of R&D jobs	78.77%	**0.0%**	**100%**	81.89%	65.37%
% of sales jobs	**0.0%**	**100%**	**0.0%**	13.96%	30.34%
% of HR studies	**8.57%**	**0.0%**	**0.0%**	2.26%	1.84%
% of life & sciences studies	**47.75%**	**32.76%**	45.73%	38.87%	41.22%
% of marketing studies	**0.0%**	**35.45%**	0.0%	5.28%	10.82%
% of medical studies	26.93%	**20.29%**	36.66%	**42.64%**	31.56%
% of healthcare representatives	19.59%	**0.0%**	0.0%	**31.32%**	8.91%
% of HR jobs	**21.22%**	**0.0%**	**0.0%**	**0.0%**	3.54%
% of laboratory technicians	**0.0%**	**0.0%**	**47.0%**	**0.0%**	17.62%
% of managers	**0.0%**	**0.0%**	**0.0%**	**38.49%**	6.94%
% of manufacturing directors	**59.18%**	**0.0%**	**0.0%**	**0.0%**	9.86%
% of research directors	**0.0%**	**0.0%**	**0.0%**	**30.19%**	5.44%
% of research scientists	**0.0%**	**0.0%**	**53.0%**	**0.0%**	19.86%
% of sales executives	**0.0%**	**79.71%**	**0.0%**	**0.0%**	22.18%
% of sales representatives	**0.0%**	**20.29%**	**0.0%**	**0.0%**	5.64%
% of married employees	43.26%	45.23%	**43.19**	**54.34**	45.78%
% of single employees	29.79%	35.21%	**35.57%**	**21.51%**	31.97%

Now that we have created satisfying clusters, we can make an exploratory analysis of the employees inside these 4 clusters. First, we could look at the distribution of employees across the 4 clusters, table n° 8. While the majority of employees are in the third cluster (Cluster 2), all clusters contain at least 15% of employees so the distribution is suitable.

Table 9 – Observations distribution across clusters

Cluster 0	Cluster 1	Cluster 2	Cluster 3
16.67%	27.82%	37.48%	18.03%

We have computed basic descriptive statistics about our 4 clusters, table n°8. It's time to analyze the results and describe the typical employee of each cluster.

— **Cluster 0** : This cluster contains **less males than any other cluster** (56.73% of males). Employees have a similar age, monthly income, job level, mean career length, mean years at company, mean years in current role and mean years with current manager than the average employee across the 4 clusters. Employees in this cluster stand out for the **proportion of human resources jobs**, which is the highest among the 4 clusters (21.22% of HR jobs).

This result is consistent with the **proportion of human resources studies**, which is also the highest among the 4 clusters (8.57% of HR studies). The cluster also contains the **highest number of manufacturing directors**, since they represent 59.18% of the employee's jobs.

→ **Cluster 0 is the cluster of human resources employees**, with more females than usual. Employees in this cluster have a lot of similarities with the average employee across the 4 clusters such as their income, job level, career length etc... are pretty standard.

— **Cluster 1** : This cluster displays the **highest rate of attrition** in any cluster (22.00% of employees have left the company). Employees have a similar age, male proportion, monthly income, job level, mean career length, mean years at company, mean years since last promotion, mean years in current role and mean years with current manager than the average employee across the 4 clusters. Employees in this cluster stand out for the **proportion of sales jobs** since **79.71% of them are sales executives** and **20.29% of them are sales representatives** for a total of **100% of sales jobs.**

→ **Cluster 1 is the cluster of sales employees**, which is also the cluster with the highest rate of attrition. However, they also share a lot of common characteristics with the average employee across the 4 clusters (income, job level etc...).

— **Cluster 2** : The third cluster contains **the youngest employees** with 34.17 years old on average and the **highest proportion of males** (63.88%). It's also the cluster with the **least average monthly income** (3238.6) and the **least average job level.** Employees in this cluster have a **short career** (7.69 years on average), have **stayed less in the company** (5.07 years on average) and have **stayed less in their current role** (3.24 years on average). It's the cluster with the **least amount of years on average since the last promotion** (1.47 years) but that may be due to the fact that employees are the most recent ones in the company. Same thing for the number of years with their current manager (3.25 years on average). **100% of the employees are in a RD job. 47% of them are laboratory technicians** while the other **53% are research scientists.** Finally, it's the cluster with the **lowest proportion of married employees** and the **highest proportion of single employees** (43.19% are married while 35.57% of them are single).

→ **Cluster 2 is the cluster of technical employees**, which is mostly composed of males. Employees in this cluster are young and have a low income. They don't stay long in the company and it's the second cluster when it comes to attrition (19.78% of them left the company).

— **Cluster 3** : The final cluster contains the **oldest employees** (43.95 years old on average). It's also the cluster with the **lowest rate of attrition** since only 4.9% of the employees left the company. The employees earn the **highest income** of the 4 clusters with 13 921.82 dollars on average. They have the **highest job level**, and this can be explained by the fact that **38.49% of them are managers.** Employees have the longest career, the longest time in the company, in their current role, since their last promotion and with their current manager. 42.64% of them come from medical studies. Finally, it's the cluster with the **highest proportion of married employees** and the **lowest proportion of single employees** (54.34% of them are married while 21.51% of them are single).

→ **Cluster 3 is the cluster of senior executives**, which contains the oldest employees who have access to the highest salary and the best roles, mostly managers. This explains why it's also the cluster with the least rate of attrition.

To conclude this exploratory analysis, we could say that our clustering results are very satisfying. We have found coherent clusters which reflect typical life situations in companies and make a lot of sense. The use of dimensionality reduction methods was very helpful to obtain a great segmentation of the employees, with t-SNE giving the best results.

3.2 A need for non-linearities introduction : a baseline model confront to machine learning

As described in the methodological part, we first estimate the probability of attrition with a linear model estimate by ordinary least squares. The goal is to find out if a machine learning model that takes non-linear behavior into account can better predict this probability of attrition. We measure our results based on F1 scores, because our data are unbalanced.

Both the support vector machine and the XGBoost ensemble perform better than the linear model estimated by OLS. We see it because of the metrics that are better but it also can be visualized. First with the recall-precision plot where both SVM and XGBoost have some points closer to the east-north side of the plots than the OLS model, i.e. that they better classify.

Then, looking at the residuals distribution plots, it is clear that the two methods lead to a better distribution of residuals with a mean more center around zero and a less large distribution. It is also important to point out at that there is a heavy tail at the right part of the distribution. This is due to the unbalanced situation of our data. Our models are almost not trained to recognize observations that take the value 1 and thus the models seek to predict observations 1.
Thus, when computing the residuals as $residuals = y - \hat{y}$ with $y \epsilon [0, 1]$ dummy observations and $\hat{y} \epsilon [0 : 1]$ that represents a probability to be 1, when the prediction of probability is very small but the true observation is indeed a 1, then it leads to a residuals close to 1 and this is what draw this right tail. We do not have any tail on the left part because the models predict well the case where the observation is 0.

Finally, looking in detail at the metrics, the SVM performs better with a F1 score of 0.5373, against only 0.4931 for the OLS estimate.

Table 10 – Machine learning metrics

Method	F1	Accuracy
Support Vector Machine	0.5373	0.8945
EXtrem Gradient Boosting	0.5135	0.8775
Ordinary Least Squares	0.4931	0.8741

3.3 Introduction of non-linearities via penalization methods

Now that it is clear that there are some non-linearities in the attrition behavior, the goal is to model them to be able to introduce it to a model estimated by OLS. We introduce cube and square of all initial features to our dataset.
Let us now analyze the selection of variables made by the three different penalization methods that have been implemented.

FIGURE 12 – Precision-recall and errors distribution plots

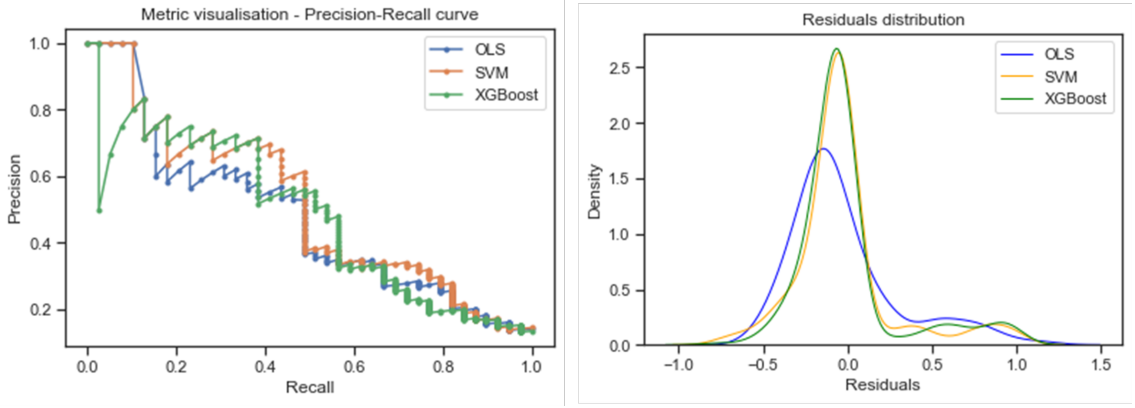


TABLE 11 – Automatic features selection

Methods	Number of initial features	Number of features selected
Ridge	93	48
Lasso	93	35
GET	89	22

The penalization terms are respectively 0.5 and 0.001 for Ridge and Lasso methods. For Lasso, it means that a considerable amount of features are set to 0 and that the selection is efficient. This is reflected in the number of features selected, Ridge selected 48 features out of 93 while Lasso selected 35 out of the 93 initial features. To define the variables selected, we use the threshold of 0.001. For these two methods, the ten first features are illustrated in figure n° 12 and their 20 first features and coefficients can be read in table n° 13 and 14.
Looking quickly at the ten first features of both methods, it seems that the number of years with the current manager, the age, the monthly income and training times last years are the features with the most importance in attrition decisions. More important, square and cube features have been selected and have a strong importance. For example, the yearly variables squared mean that the relation is U shape with attrition, in the first years, one more year has an increasing impact on the attrition decision, but then after a threshold, one more year has a decreasing impact on the attrition decision.

FIGURE 13 – Features importance : Ridge and Lasso

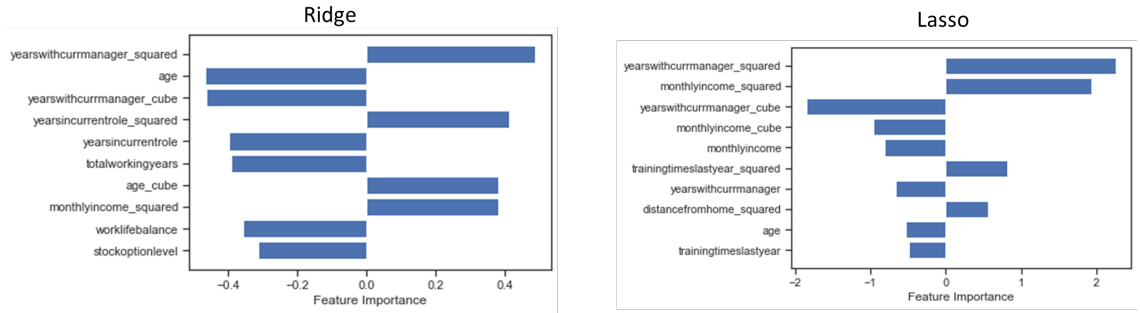


TABLE 12 – Ridge variables selections (20 firsts)

Variables	Coefficients
yearswithcurrmanager_squared	0.4853
age	-0.4653
yearswithcurrmanager_cube	-0.4638
yearsincurrentrole_squared	0.4127
yearsincurrentrole	-0.3981
totalworkingyears	-0.3906
age_cube	0.3819
monthlyincome_squared	0.3818
worklifebalance	-0.3563
stockoptionlevel	-0.3138
environmentsatisfaction	-0.3075
jobinvolvement	-0.3047
worklifebalance_cube	0.3042
trainingtimeslastyear_squared	0.2997
trainingtimeslastyear	-0.2647
distancefromhome_squared	0.2626
yearsatcompany_cube	0.2516
joblevel_squared	0.2401
joblevel	-0.2395
distancefromhome_cube	-0.2369
overtime	0.2330

TABLE 13 – Lasso variables selections (20 firsts)

Variables	Coefficients
yearswithcurrmanager_squared	2.2467
monthlyincome_squared	1.9258
yearswithcurrmanager_cube	-1.8466
monthlyincome_cube	-0.9611
monthlyincome	-0.8170
trainingtimeslastyear_squared	0.8079
yearswithcurrmanager	-0.6615
distancefromhome_squared	0.5610
age	-0.5331
trainingtimeslastyear	-0.4876
age_cube	0.4631
yearsatcompany_cube	0.4490
distancefromhome_cube	-0.4441
trainingtimeslastyear_cube	-0.4203
worklifebalance	-0.4104
yearsincurrentrole_squared	0.4056
yearsincurrentrole	-0.3792
stockoptionlevel	-0.3665
jobinvolvement	-0.3660
totalworkingyears	-0.3415
stockoptionlevel_squared	0.3360

For the Autometrics application, we used the Oxmetrics software to perform the general-to-specific procedure. In the software, one needs to choose a significance level for selection. We choose 0.001 level for p-values to have few variables retained among our dozens of variables as we want a parsimonious model. We also choose to have robust standard deviations to deal with potential heteroskedasticity.

We begin with 89 variables which includes the square and the cube of our variables. Oxmetrics gives interesting statistics about how the Autometrics procedure behaved. Our initial search space is 2^{89} which is too long to estimate. After selecting the unique models, the final search space counts 2^{40} models. Finally, only 1135 models are estimated and 10 terminal models are found, i.e 10 terminal nodes. The procedure ends up with 22 variables that are significant at the 0.001 level, which is the procedure with the lowest number of variables retained among the three we used, table n° 15.

Interestingly, the autometrics algorithm selects the square and the cube of some variables. This corroborates our precedent findings about the presence of some non-linearities behavior in our dataset.

Table 14 – Autometrics procedure variables selections

Variables	Coefficient
age	-0.0308
environmentsatisfaction	-0.0376
joblevel	-0.1780
overtime	0.2021
stockoptionlevel	-0.1626
businesstravel_travel_frequently	0.1045
department_research_development	-0.0879
age_squared	0.0003
joblevel_squared	0.0261
distancefromhome	0.0040
monthlyrate	5.62679e-07
performancerating	0.8464
yearswithcurrmanager	-0.0619
jobinvolvement_cube	-0.0023
jobsatisfaction_squared	-0.0071
numcompaniesworked_squared	0.0017
performancerating_squared	-0.1212
stockoptionlevel_squared	0.0478
worklifebalance_squared	-0.0572
worklifebalance_cube	0.0123
yearswithcurrmanager_squared	0.0102
yearswithcurrmanager_cube	-0.0004

3.4 Study of the best classifier

We finally choose to keep the features selection of Autometrics, we then predict the attrition probability using the non-linear modeling estimate by OLS and using SVM. We compare these two predictions to the baseline linear model estimated by OLS.

Table 15 – Best classifier metrics

Method	F1	Accuracy
SVM	0.5569	0.8809
OLS (non-linear)	0.5176	0.8605
OLS (linear)	0.4931	0.8741

First, the introduction of non-linearities allows to have a better estimation of attrition by ordinary least squares. The F1 metrics was at 0.49 with the baseline modeling and increased to 0.51 with the non-linearities.
However, the support vector machine with a sigmoid kernel allows for a much better prediction. The F1 metrics increase to 0.55. The introduction of non-linearities allows to increase the performance of this machine learning classifier. It means that it was able to capture new behaviors via the introduction of non-linearities.

If we look closer at this best classifier, the confusion matrix shows that observation of employees that stay are well classified at 92.24%, while employees that quit (attrition) is well predicted at only 56.40%, this is still due to the fact that our dataset is unbalanced and the classifier have few observations to train for the true attrition situation.

Table 16 – Confusion matrix of the SVM classifier

	Stay	Quit
Stay	92.94 % (237)	7.06 % (18)
Quit	43.60 % (17)	56.40 % (22)

4 A need for interpretation

After selecting the variables through the Autometrics procedure. We want to have a good understanding of the variables that are most important in determining attrition. To do so, one can use the SHAP package which comes with visualization tools to interpret models.
The most common way to understand a linear model is to look at the learned coefficients for each variable. These coefficients tell us how much the output of the model, the probability of an employee leaving the company, changes when we change each of the input features.
These coefficients can directly tell us what happens if we change the value of a variable, however we can't say anything about the relative importance of each variable. This is because variables have different scales so the magnitude of the coefficient does not reflect the importance of the variable. The shap values help us to appreciate the importance of each variable. The following graph shows variables by order of importance. The variable with the highest importance is "years with current manager". The width is associated with the density of observations for each variable. For categorical variables, "job level" for instance, we have 5 points with different numbers of observations corresponding to the 5 values of the variable. There is also a color gradient from blue to red where red corresponds to individuals who are more likely to leave the company while

blue is the opposite. For instance, following the variable “age”, young people are more likely to quit.

In the second type of graph figure n°15, called waterfall because it is read from top to bottom, the process of the prediction for one specific individual is explained. Here f(x)=-0.074 is the output of the linear probability model. The list of variables is ordered in a decreasing way from the most important variable to the least like the precedent graph. On the left, we have the exact value of the variable for the individual. For instance, here the individual is aged 31 and has spent the last 2 years with its current manager. This variable reduces the probability of attrition by 33%.

FIGURE 14 – Shap values plots

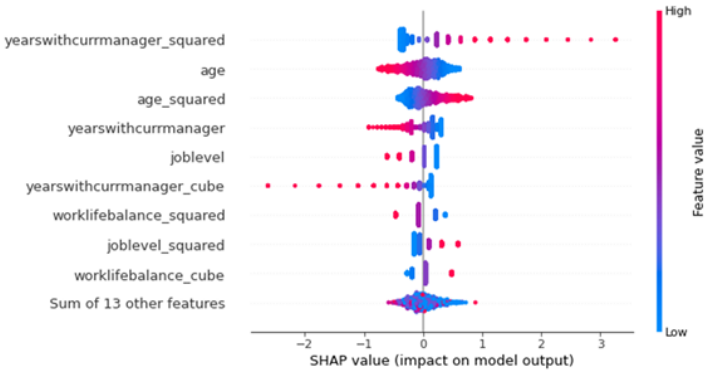
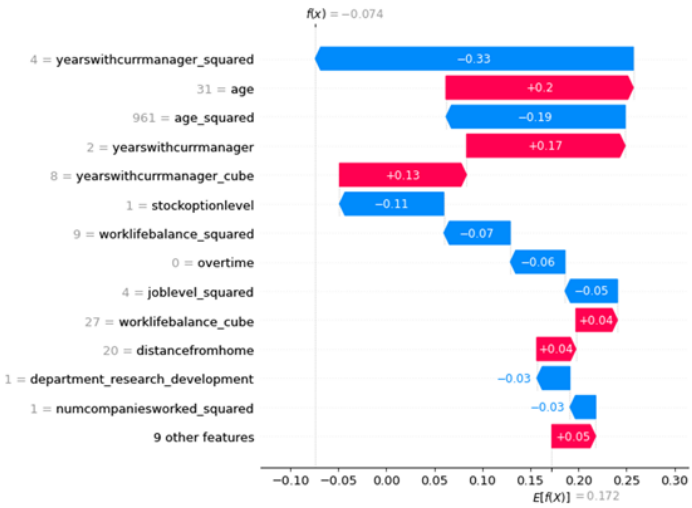


FIGURE 15 – Shap values plots



5 Conclusion

To sum-up, this study aimed to understand the factors which influence attrition. In a first part, we conduct an exploratory analysis by creating clusters of employees. We obtained a satisfying segmentation of the individuals in 4 distinct and coherent clusters. Then in a second part, we focused on predicting attrition and mostly, on the features which explains the phenomenon best. We compared a baseline OLS model with machine learning methods to detect potential non linearities and discover the best model to predict attrition. Then we incorporate non linearities and automatically select the best variables with penalization methods such as Ridge, Lasso or the GETS procedure. The best model for predicting attrition with our data is a Support Vector Machine model. Finally, we analyzed the impact of the best features on attrition via the Shap values to understand how they influence the decision of employees to leave the company.

In future works, we could integrate more non linearities, such as cross products or exponential transformations. Since we have not rebalanced our dataset, we could also use resampling methods to obtain better results such as oversampling. We suffered from a lack of observations relative to the number of variables and one solution could be to add data from other sources to complete our dataset, as well as bringing new information. However, we must keep in mind that predicting attrition with Machine Learning models which are considered as black-boxes for their lack of interpretability, can be subject to raise ethical questions. Since 2008 in France, using personal data such as the age of the employee in the process of employment is illegal due to discrimination concerns, so constructing models to know in advance if an employee will leave the company can't be done. But this kind of study can help companies to keep their employees and avoid churn by focusing on the factors of attrition.

6 Bibliography

Aseel Qutub (2021), Prediction of Employee Attrition Using Machine Learning and Ensemble Methods.
Brouwnlee, A Gentle Introduction to Threshold-Moving for Imbalanced Classification
C. Ding and X.He (2004), "K-means Clustering via Principal Component Analysis
Doornik Hendry (2015), Statistical model selection with "Big Data".
Fallucchi, Coladangelo, Giuliano and De Luca (2020), Predicting Employee Attrition Using Machine Learning Techniques.
Frye, Boomhower, Smith, Vitovsky and Fabricant (2018), Employee Attrition : What Makes an Employee Quit ?
Guerranti (2021), Employee attrition : what causes employees to quit ?
S. Lundberg (2017) A Unified Approach to Interpreting Model Predictions
H. Liu al. (2021), Using [...] t-SNE for cluster analysis and spatial zone delineation of groundwater geochemistry data
Yang and Islam (2021), IBM Employee Attrition Analysis.
Zou, Xie, Lin, Wu and Ju (2015), Finding the Best Classification Threshold in Imbalanced Classification

Link to the GitHub repository of this project

Table des matières

1 Introduction **1**
- 1.1 Context . . . 1
- 1.2 Literature review . . . 1
- 1.3 Method summary . . . 3
- 1.4 Key results . . . 4

2 Materials and methods **4**
- 2.1 Data description . . . 4
- 2.2 Methods . . . 5
  - 2.2.1 Methodology : general modelization and evaluation metrics . . . 5
  - 2.2.2 Clustering methods : a tool for a deep exploratory analysis . . . 6
  - 2.2.3 Machine learning methods : assess the existence of non-linearities . . . 8
  - 2.2.4 Methods to determine non-linearities . . . 11
  - 2.2.5 Interpretation methods . . . 13

3 A global study of attrition decision in employment **14**
- 3.1 A deep exploratory analysis . . . 14
  - 3.1.1 Classical descriptive statistics . . . 14
  - 3.1.2 Common patters define by clustering methods . . . 17
- 3.2 A need for non-linearities introduction : a baseline model confront to machine learning 22
- 3.3 Introduction of non-linearities via penalization methods . . . 22
- 3.4 Study of the best classifier . . . 26

4 A need for interpretation **26**

5 Conclusion **28**

6 Bibliography **28**

7 Annexes **30**

7 Annexes

TABLE 17 – Descriptive statistics

var	y	age	dailyrate	distancefromhome	education	envrionment staisfaction
mean	0.161	36.924	802.486	9.192	2.913	2.722
std	0.368	9.135	403.51	8.107	1.024	1.093
min	0	18	102	1	1	1
25%	0	30	465	2	2	2
50%	0	36	802	7	3	3
75%	0	43	1157	14	4	4
max	1	60	1499	29	5	4
var	**hourlyrate**	**jobinvolvement**	**joblevel**	**jobsatisfaction**	**monthlyincome**	**monthlyrate**
mean	65.891	2.73	2.064	2.73	6502.931	14313.103
std	20.329	0.711	1.107	1.103	4707.957	7117.786
min	30	1	1	1	1009	2094
25%	48	2	1	2	2911	8047
50%	66	3	2	3	4919	14235.5
75%	83.75	3	3	4	8379	20461.5
max	100	4	5	4	19999	26999
var	**numcopaniesworked**	**percent salaryhike**	**performancerating**	**relationshipsatisfaction**	**stockoptionlevel**	**totalworkingyears**
mean	2.693	15.209	3.154	2.712	0.794	11.279
std	2.498	3.66	0.361	1.081	0.852	7.781
min	0	11	3	1	0	0
25%	1	12	3	2	0	6
50%	2	14	3	3	1	10
75%	4	18	3	4	1	15
max	9	25	4	4	3	40
var	**trainingtimeslastyear**	**worklifebalance**	**yearsatcompany**	**yearsincurrentrole**	**yearssincelastpromotion**	**yearswithcurrmanager**
mean	2.799	2.761	7.008	4.23	2.188	4.123
std	1.289	0.706	6.126	3.623	3.222	3.568
min	0	1	0	0	0	0
25%	2	2	3	2	0	2
50%	3	3	5	3	1	3
75%	3	3	9	7	3	7
max	6	4	40	18	15	17

Table 18 – Glosarry

Variable	Variable type	Description
Attrition (y)	Binary string (Yes/No)	Has the employee left the company ?
Age	Discrete numerical	Age of the employee
Business Travel	String	Does the employee travel rarely, frequently or never ?
Daily Rate	Continuous numerical	The daily cost of the employee for the company.
Department	String	The department of IBM where the employee works.
Distance From Home	Discrete numerical	The distance between the employee's home and IBM.
Education	Discrete numerical	Measure of the education level of the employee from 1 to 5.
Education Field	String	Field of education where the employee comes from.
Employee Count	Constant numerical	This variable takes only the value "1".
Employee Number	Discrete numerical	Number of the employee in the company, one unique number per employee.
Environment Satisfaction	Discrete numerical	Measure of the environment satisfaction of the employee from 1 to 4.
Gender	Binary string (Male/Female)	Gender of the employee.
Hourly Rate	Discrete numerical	The hourly cost of the employee for the company.
Job Involvement	Discrete numerical	Measure of the involvement of the employee in its job from 1 to 4.
Job Level	Discrete numerical	Measure of the level of the employee's job from 1 to 5.
Job Role	String	Role of the employee's job.
Job Satisfaction	Discrete numerical	Measure of the level of the employee's satisfaction in its job from 1 to 4.
Marital Status	String	Marital status of the employee : single, married or divorced.
Monthly Income	Continous numerical	Monthly income of the employee.
Monthly Rate	Continous numerical	The monthly cost of the employee for the company.
NumCompanies Worked	Discrete numerical	Number of companies where the employee has worked before.
Over 18	Constant string	Is the employee over 18 ? All employees are adults, this variable takes only the value "Y" for "Yes".
Over Time	Binary string (Yes/No)	Does the employee work overtime ?
Percent SalaryHike	Discrete numerical	Percentage increase in salary since last year.
Performance Rating	Discrete numerical	Rating of the employee's performance ("3" or "4").
Relationship Satisfaction	Discrete numerical	Measure of the employee's relationship satisfaction from 1 to 4.
Standard Hours	Constant numerical	This variable takes only the value "80".
StockOption Level	Discrete numerical	Measure of the employee's stock option level from 0 to 3.
TotalWorking Years	Discrete numerical	Number of years the employee has worked in his life.
TrainingTimes LastYear	Discrete numerical	Number of times the employee did training last year.
WorkLife Balance	Discrete numerical	Measure of the employee's work and personal life balance from 1 to 4.
YearsAt Company	Discrete numerical	Number of years the employee has worked in the company.
YearsIn CurrentRole	Discrete numerical	Number of years since the employee had his current job.
YearsSince LastPromotion	Discrete numerical	Number of years since the employee's last promotion
YearsWith CurrManager	Discrete numerical	Number of years since the employee works for his current manager.

FIGURE 16 – Boosting illustration

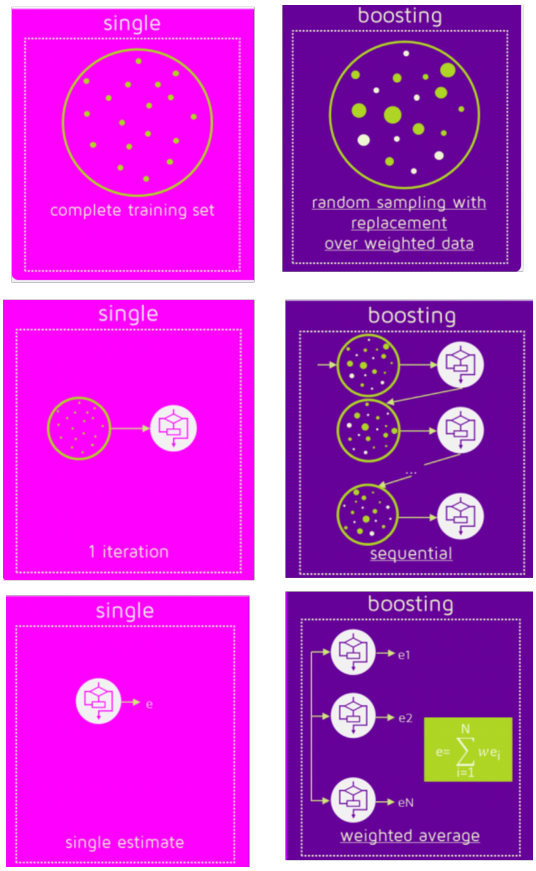


TABLE 19 – Ridge variables selections (full)

Variables	Coefficients
yearswithcurrmanager_squared	0.4853
age	-0.4653
yearswithcurrmanager_cube	-0.4638
yearsincurrentrole_squared	0.4127
yearsincurrentrole	-0.3981
totalworkingyears	-0.3906
age_cube	0.3819
monthlyincome_squared	0.3818
worklifebalance	-0.3563
stockoptionlevel	-0.3138
environmentsatisfaction	-0.3075
jobinvolvement	-0.3047
worklifebalance_cube	0.3042
trainingtimeslastyear_squared	0.2997
trainingtimeslastyear	-0.2647
distancefromhome_squared	0.2626
yearsatcompany_cube	0.2516
joblevel_squared	0.2401
joblevel	-0.2395
distancefromhome_cube	-0.2369
overtime	0.2330
jobsatisfaction	-0.2073
yearssincelastpromotion	0.1859
percentsalaryhike_cube	0.1848
education_cube	-0.1799
percentsalaryhike	-0.1774
yearsatcompany_squared	0.1770
stockoptionlevel_squared	0.1764
jobrole_research_director	-0.1733
monthlyincome	-0.1728
jobrole_sales_representative	0.1715
numcompaniesworked	0.1645
jobrole_human_resources	0.1592
education	0.1546
jobsatisfaction_squared	0.1516
yearswithcurrmanager	-0.1508
yearssincelastpromotion_cube	-0.1463
environmentsatisfaction_cube	0.1344
numcompaniesworked_squared	0.1307
relationshipsatisfaction	-0.1264
jobrole_manager	-0.1201
totalworkingyears_squared	0.1188
stockoptionlevel_cube	0.1116
jobinvolvement_squared	0.1071
dailyrate_squared	-0.1063
numcompaniesworked_cube	-0.1055
totalworkingyears_cube	0.1033
trainingtimeslastyear_cube	-0.0971

TABLE 20 – Lasso variables selections (full)

Variables	Coefficients
yearswithcurrmanager_squared	2.2467
monthlyincome_squared	1.9258
yearswithcurrmanager_cube	-1.8466
monthlyincome_cube	-0.9611
monthlyincome	-0.8170
trainingtimeslastyear_squared	0.8079
yearswithcurrmanager	-0.6615
distancefromhome_squared	0.5610
age	-0.5331
trainingtimeslastyear	-0.4876
age_cube	0.4631
yearsatcompany_cube	0.4490
distancefromhome_cube	-0.4441
trainingtimeslastyear_cube	-0.4203
worklifebalance	-0.4104
yearsincurrentrole_squared	0.4056
yearsincurrentrole	-0.3792
stockoptionlevel	-0.3665
jobinvolvement	-0.3660
totalworkingyears	-0.3415
stockoptionlevel_squared	0.3360
worklifebalance_cube	0.3158
environmentsatisfaction	-0.3062
joblevel	-0.2641
jobsatisfaction	-0.2404
percentsalaryhike_cube	0.2403
overtime	0.2314
joblevel_squared	0.2293
numcompaniesworked	0.2134
jobinvolvement_squared	0.2092
yearssincelastpromotion	0.2016
jobsatisfaction_squared	0.1958
percentsalaryhike	-0.1912
jobrole_sales_representative	0.1841
jobrole_research_director	-0.1813

FIGURE 17 – Cluster features distribution (part 1)

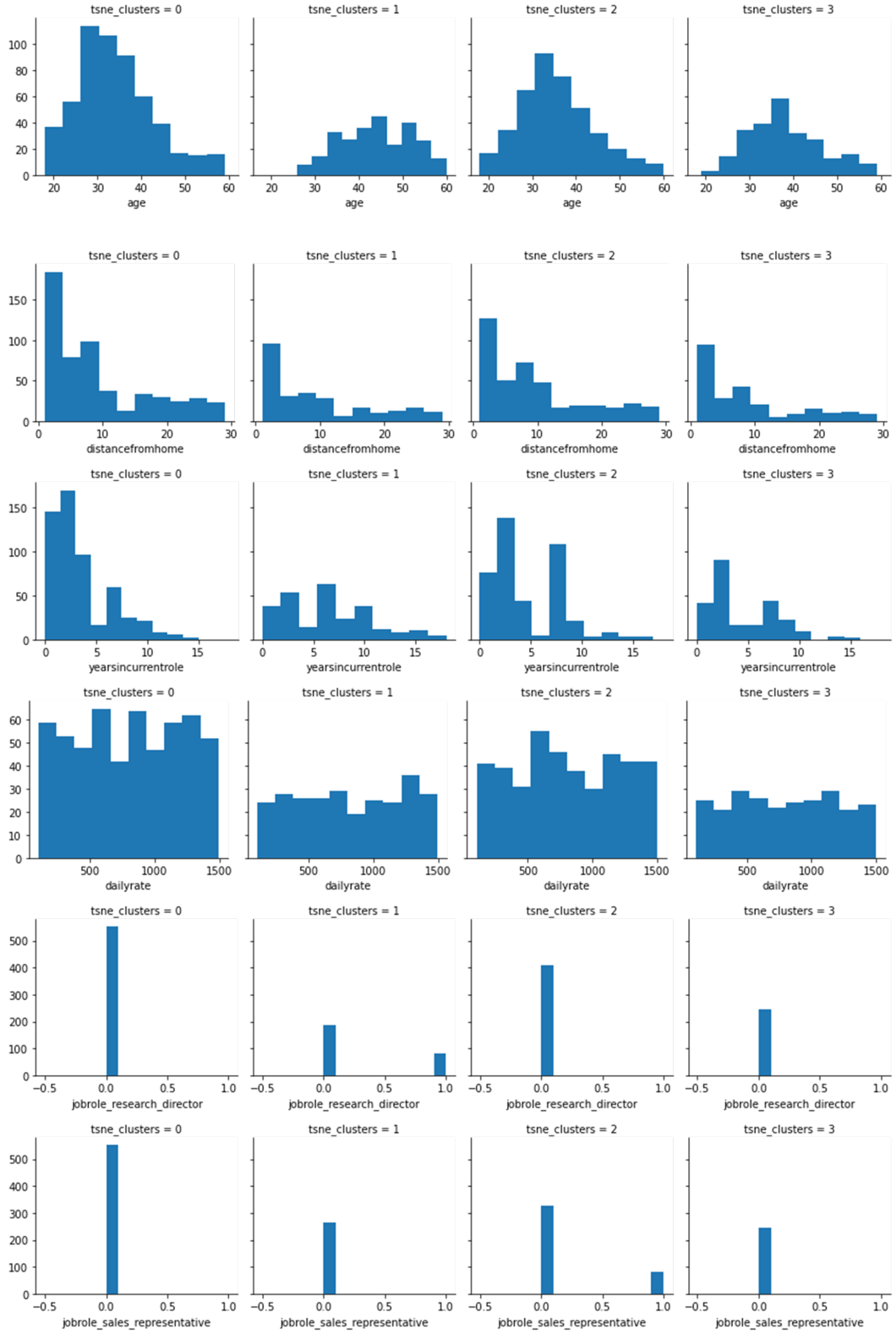


FIGURE 18 – Cluster features distribution (part 2)

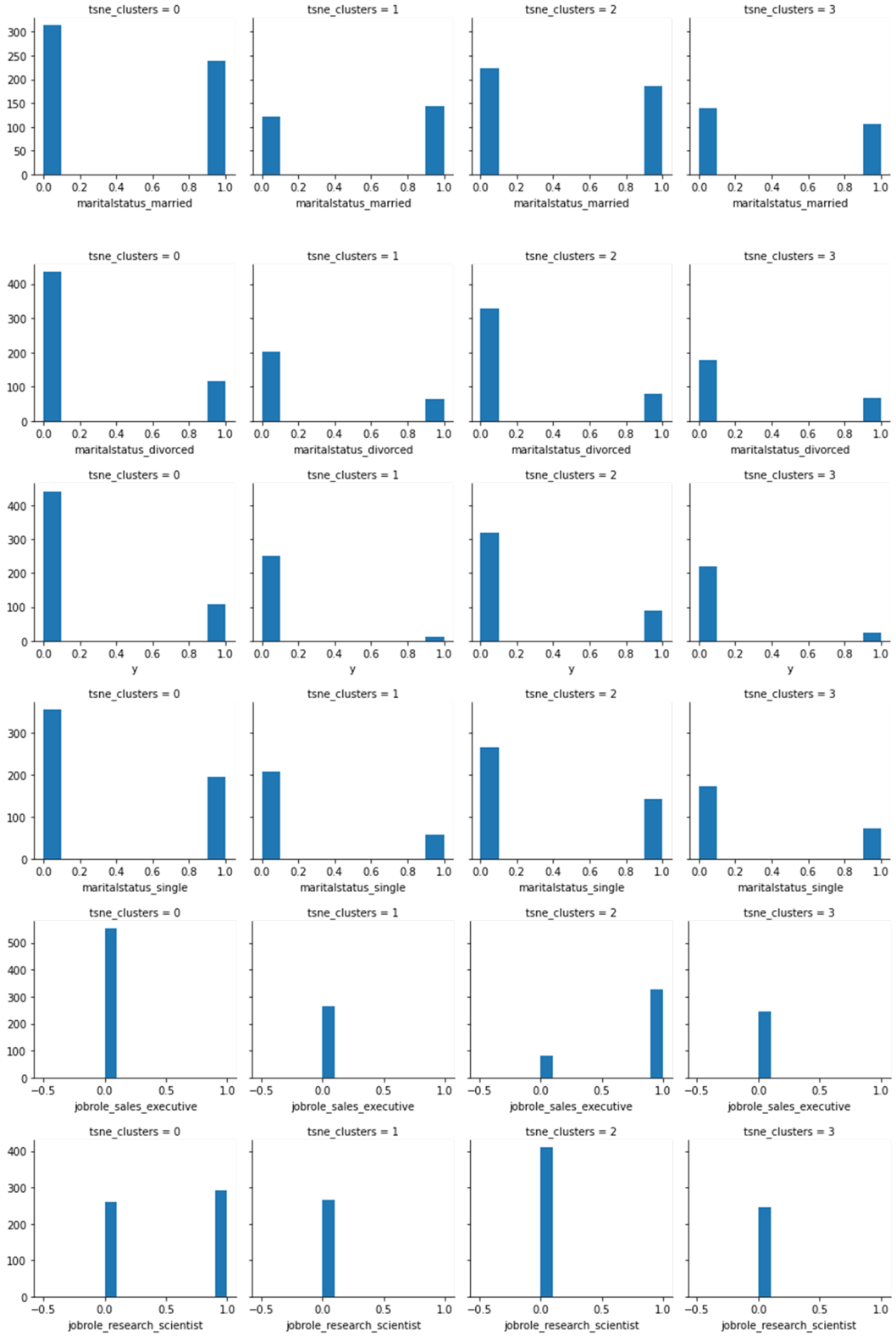


FIGURE 19 – Cluster features distribution (part 3)

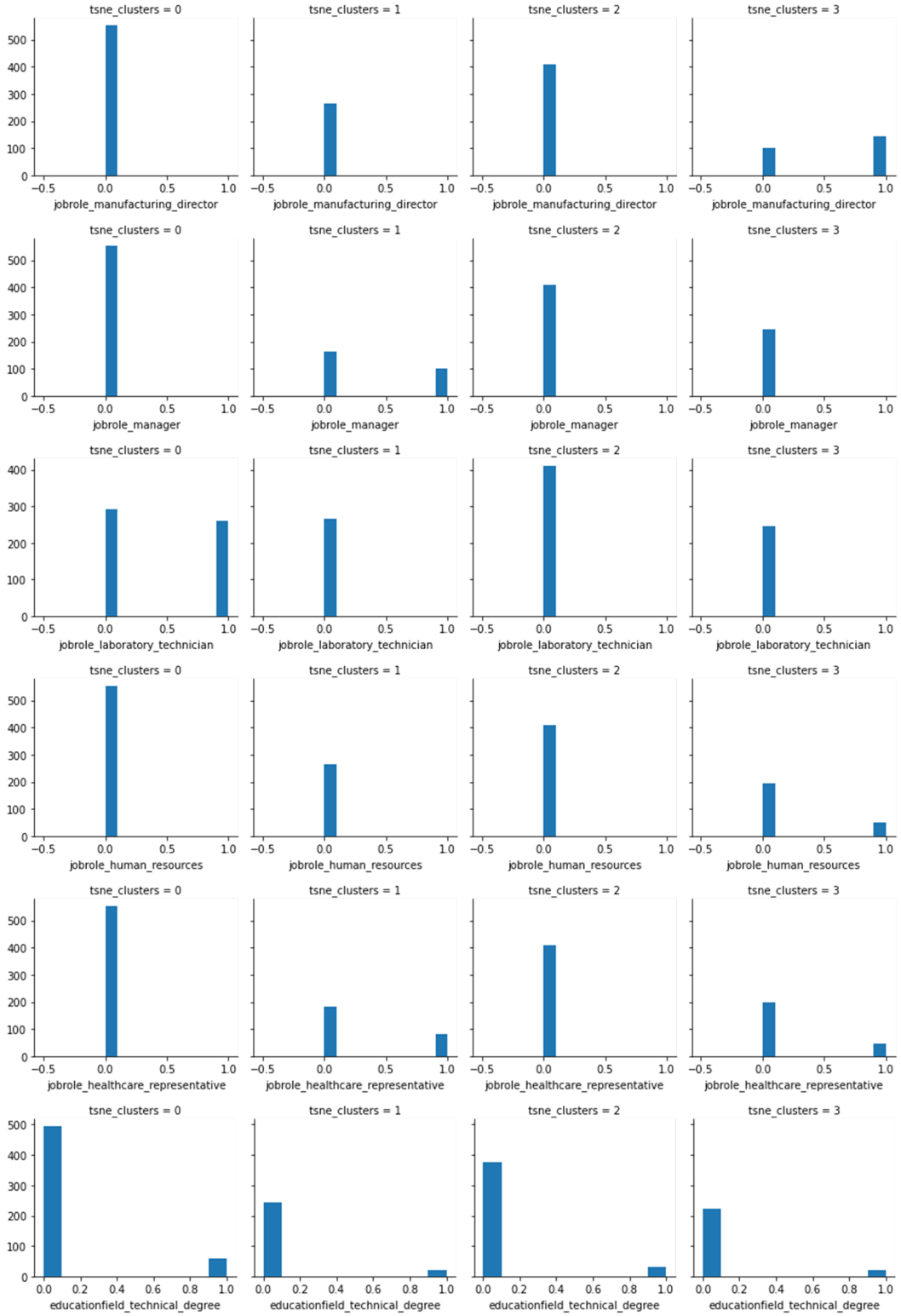


FIGURE 20 – Cluster features distribution (part 4)

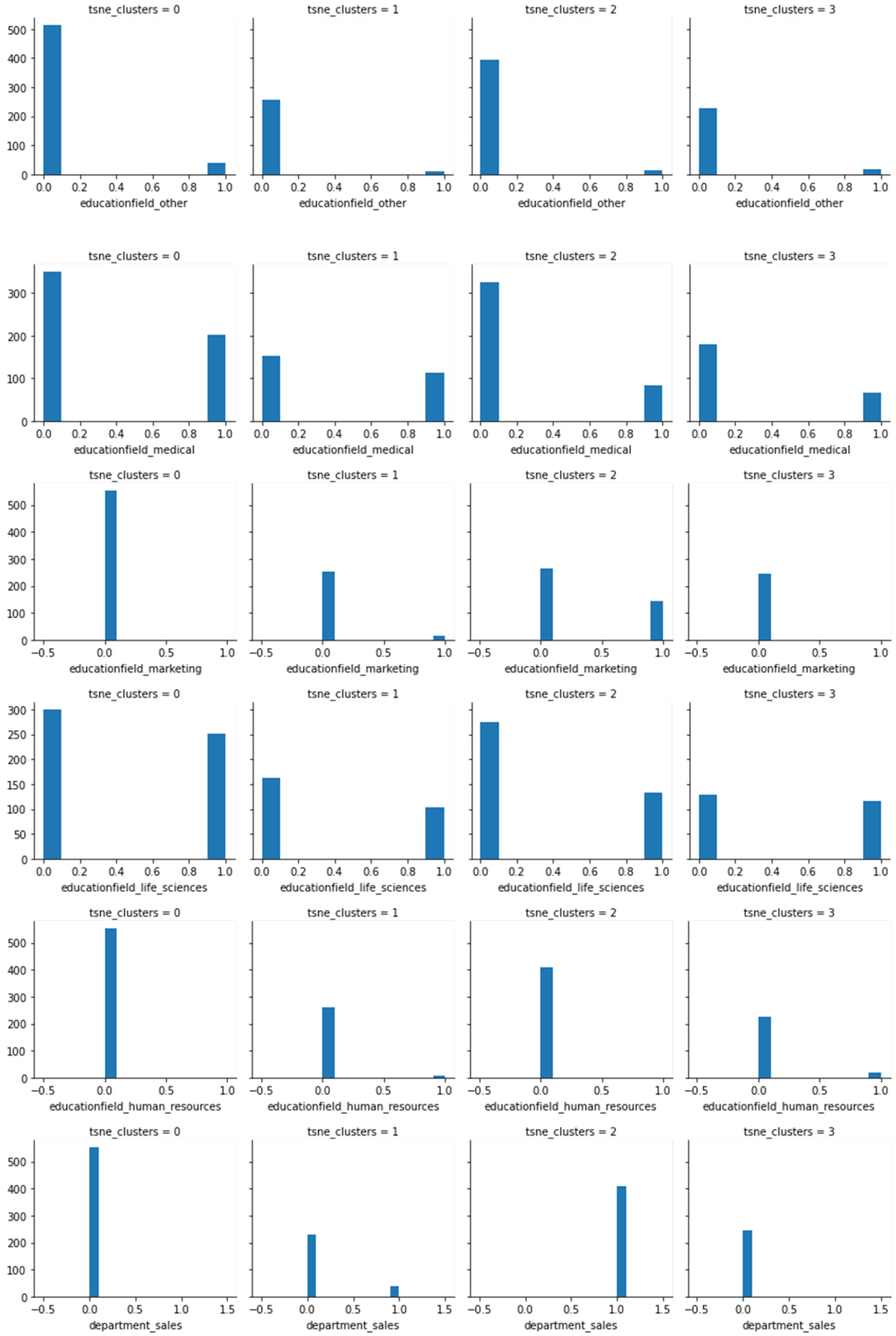


FIGURE 21 – Cluster features distribution (part 5)

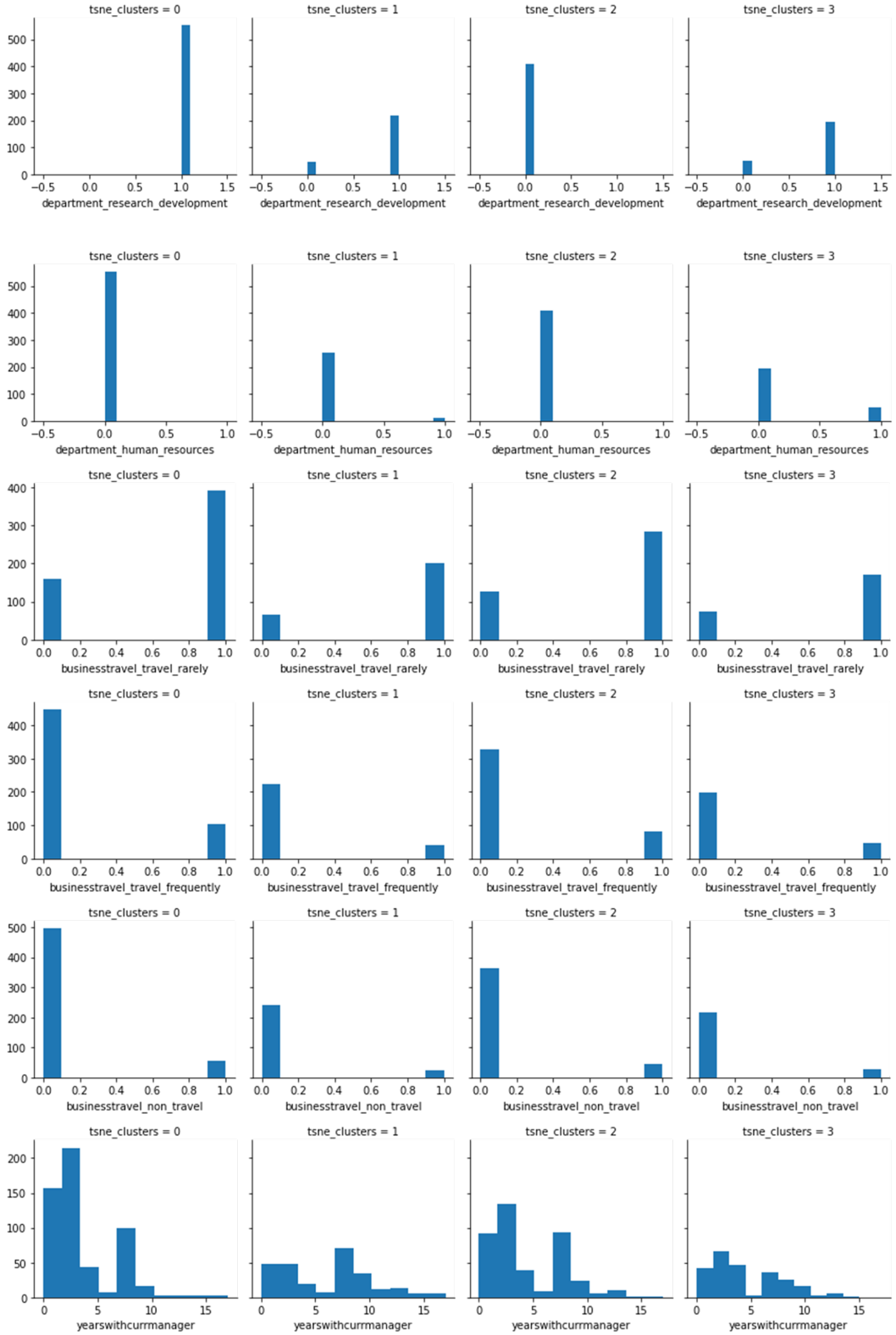


FIGURE 22 – Cluster features distribution (part 6)

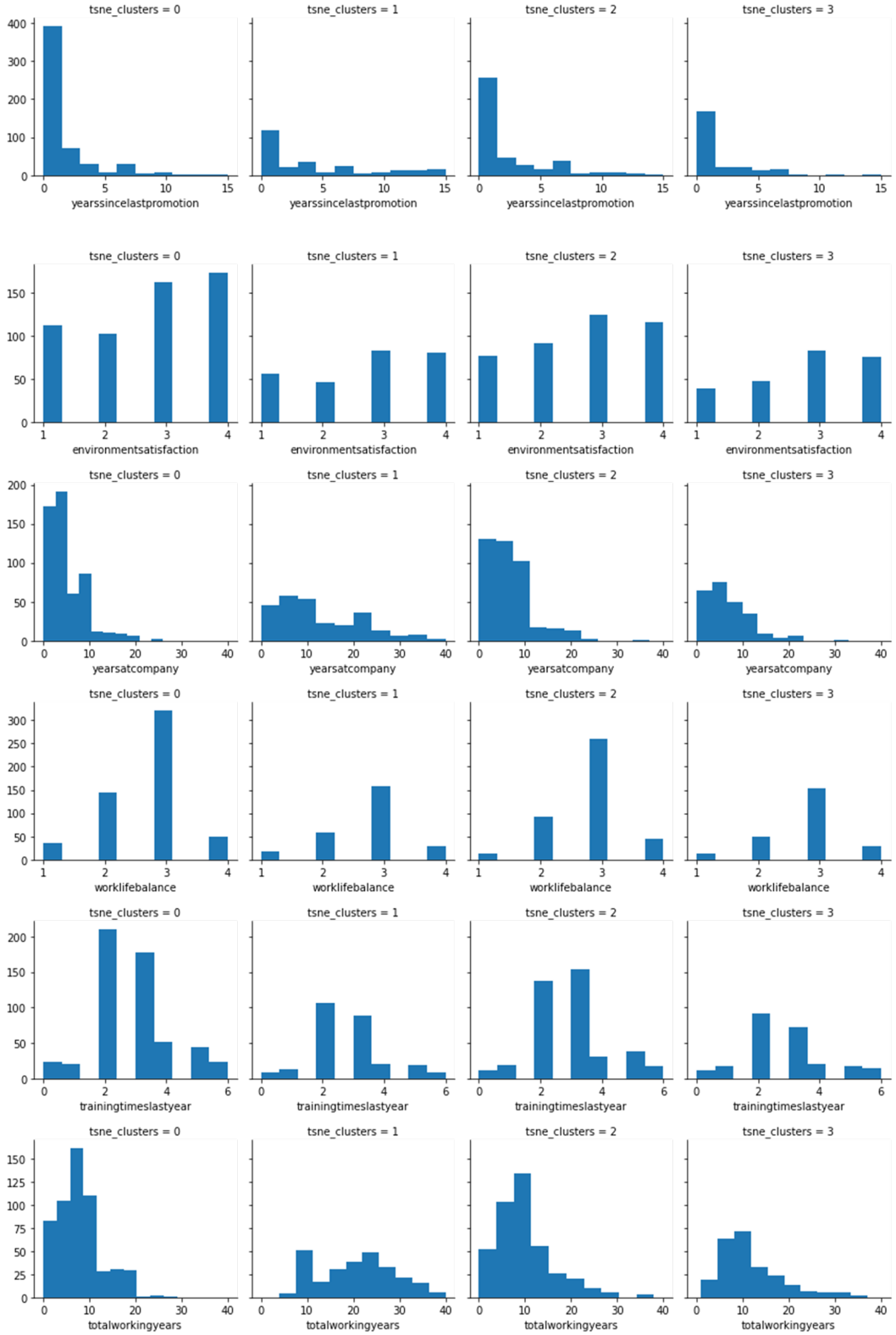


FIGURE 23 – Cluster features distribution (part 7)

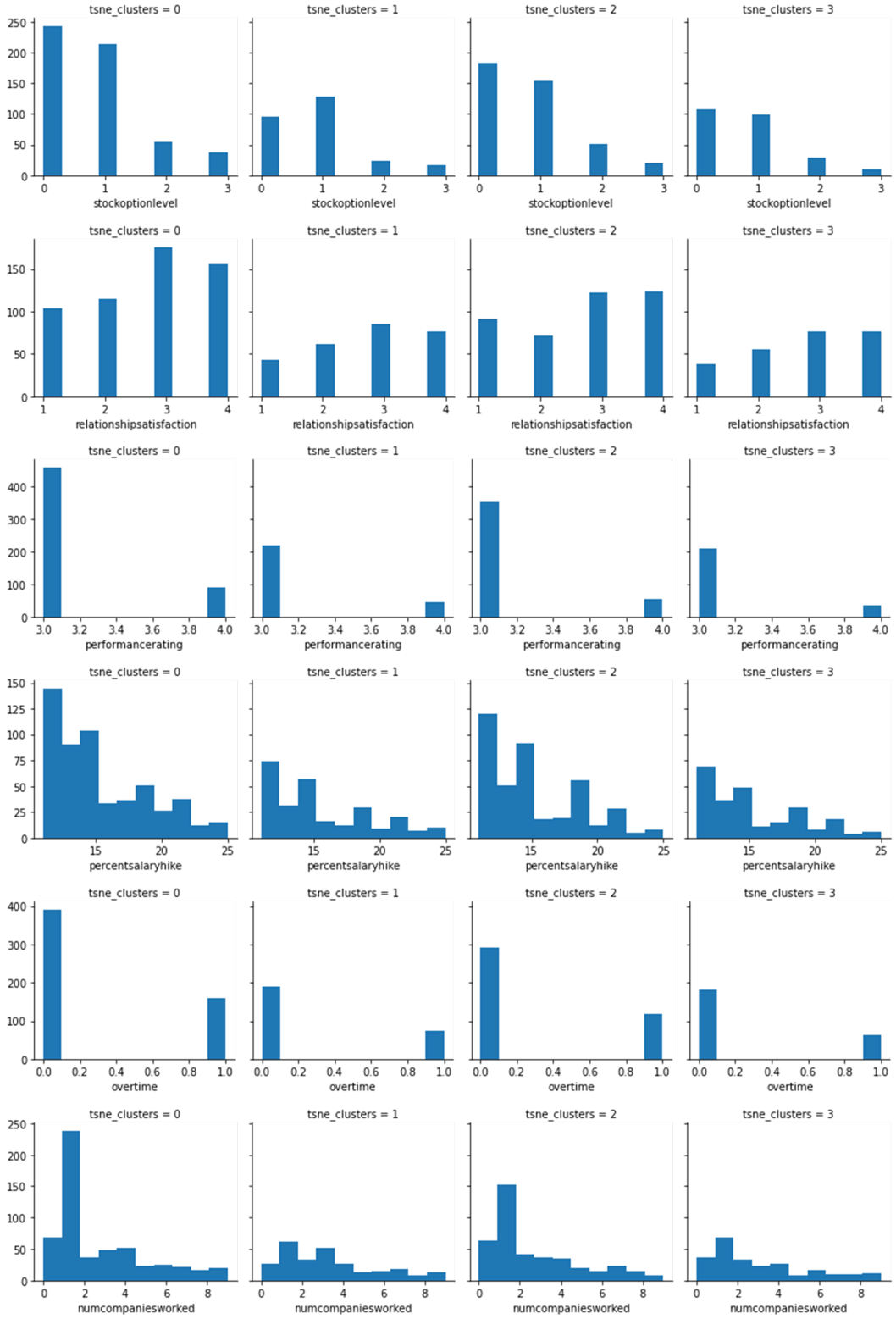


FIGURE 24 – Cluster features distribution (part 8)

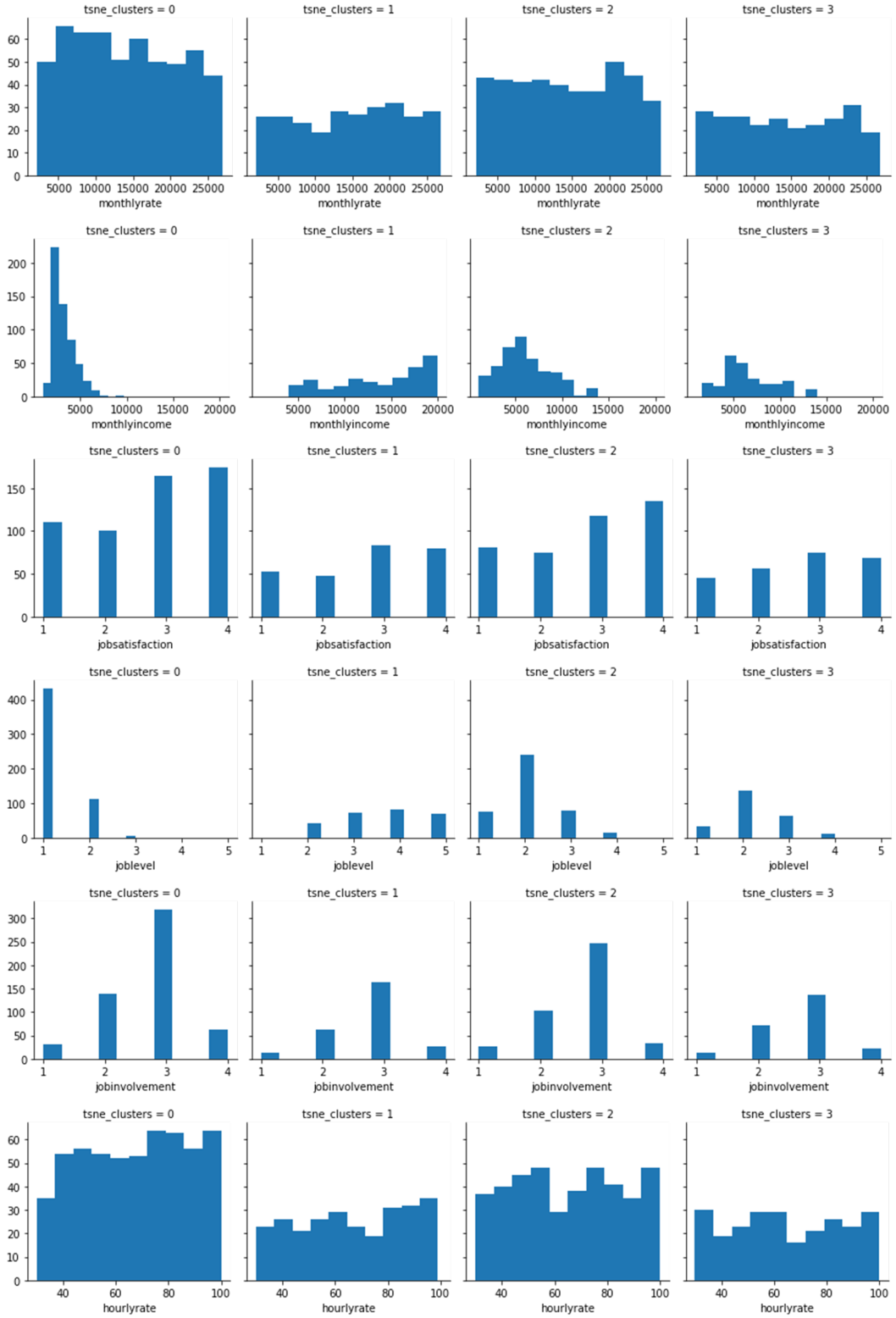


FIGURE 25 – Cluster features distribution (part 9)