1. Data semantics (3 points)

The issue of keeping one's employees happy and satisfied is a perennial and age-old challenge. If an employee you have invested so much time and money leaves the business, then this would mean that you would have to spend even more time and money to hire somebody else.
The dataset "IBM HR Analytics Employee Attrition & Performance" is a fictional data set created by IBM data scientists that contains all elements to study this phenomenon. Our job is to uncover the factors that lead to employee attrition and study the reasons that lead a worker to leave a company.

Our dataset contains 1470 entries with 33 attributes.
For each attributes let's describe the main features:

No	Name of Attribute	Description	Data Type	Attribute Type	Domain
0	Example	Feature description	Integer	Numerical	<= 100
1	Age	Age of employees	Integer	Discrete	18 - 60
2	Attrition	The reduction of a workforce by employees	Boolean	Categorical	{ Yes, No }
3	BusinessTravel	Frequency of business travels	String	Categorical	{ Non-Travel, Travel_Rarely, Travel_Frequently }
4	DailyRate	The amount of money the company has to pay someone to work for them for a day	Integer	Numerical	102 - 1499
5	Department	Different area of special expertise and responsibility	String	Categorical	{ Research & Development, Sales, Human Resources }
6	DistanceFromHome	Distance from office to home	Integer	Numerical	1 - 29
7	Education	Education level: 1 'Below College' 2 'College' 3 'Bachelor' 4 'Master' 5 'Doctor'	Integer	Ordinal	1 - 5
8	EducationField	Field of education	String	Categorical	{ Medical, Life Sciences, Technical Degree, Other, Human Resources, Marketing }

9	EnvironmentSatisfaction	Environment Satisfaction level: 1 'Low' 2 'Medium' 3 'High' 4 'Very High'	Integer	Ordinal	1 - 4
10	Gender	Sex of the worker	String	Categorical	{ Male, Female }
11	HourlyRate	Remuneration a worker receives for each hour that they work	Integer	Numerical	30 - 100
12	JobInvolvement	Job involvement level: 1 'Low' 2 'Medium' 3 'High' 4 'Very High'	Integer	Ordinal	1 - 4
13	JobLevel	Responsibility level and expectations of roles at your organization	Integer	Ordinal	1 - 5
14	JobRole	Role of employee	String	Categorical	{ Research Director, Manager, Sales Executive, Research Scientist, Laboratory Technician, Sales Representative, Manufacturing Director, Healthcare Representative, Human Resources }
15	JobSatisfaction	Job satisfaction level: 1 'Low' 2 'Medium' 3 'High' 4 'Very High'	Integer	Ordinal	1 - 4
16	MaritalStatus	Marital state	String	Categorical	{ Single, Divorced, Married }
17	MonthlyIncome	Amount paid to an employee within a month	Integer	Numerical	1009 - 19999
18	MonthlyRate	Monthly rate is the	Integer	Numerical	2094 - 26968

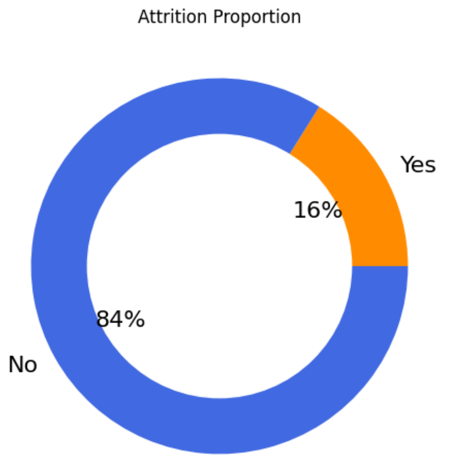
		internal charge out rate which will be used to calculate the cost of each employee monthly			
19	NumCompaniesWorked	Number of companies for which they worked	Integer	Discrete	0 - 9
20	Over18	True if age greater that 18	Boolean	Categorical	{ Y }
21	OverTime	Extra working time	Boolean	Categorical	{ Yes, No }
22	PercentSalaryHike	The amount a salary is increased	Integer	Discrete	11 - 25
23	PerformanceRating	Performance rating level: 1 'Low' 2 'Good' 3 'Excellent' 4 'Outstanding'	Integer	Ordinal	3 - 4
24	RelationshipSatisfaction	Relationship satisfaction level: 1 'Low' 2 'Medium' 3 'High' 4 'Very High'	Integer	Ordinal	1 - 4
25	StandardHours	The amount of work achievable in an hour.	Integer	Categorical	{ 80 }
26	StockOptionLevel	Level of stock compensation granted by companies to their employees and executives	Integer	Ordinal	0 - 3
27	TotalWorkingYears	Years of work in total	Integer	Discrete	0 - 40
28	TrainingTimesLastYear	Times of a particular training activity last year	Integer	Discrete	0 - 6
29	WorkLifeBalance	WorkLife balance level: 1 'Bad' 2 'Good'	Integer	Ordinal	1 - 4

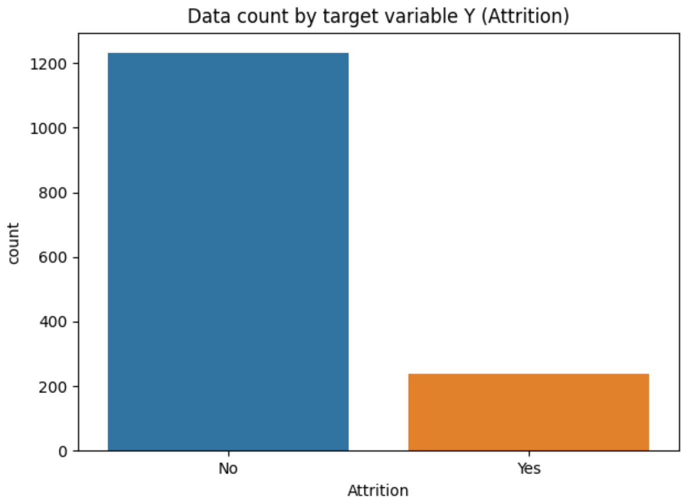
		3 'Better' 4 'Best'			
30	YearsAtCompany	Years in the company	Integer	Discrete	0 - 40
31	YearsInCurrentRole	Years in the current role	Integer	Discrete	0 - 18
32	YearsSinceLastPromotion	Years since last promotion	Integer	Discrete	0 - 15
33	YearsWithCurrManager	Years with current manager	Integer	Discrete	0 - 17

2. **Distribution of the variables and statistics (7 points)**

Let's review the distributions of the variables selected for the attrition study to get a general picture of the situation.

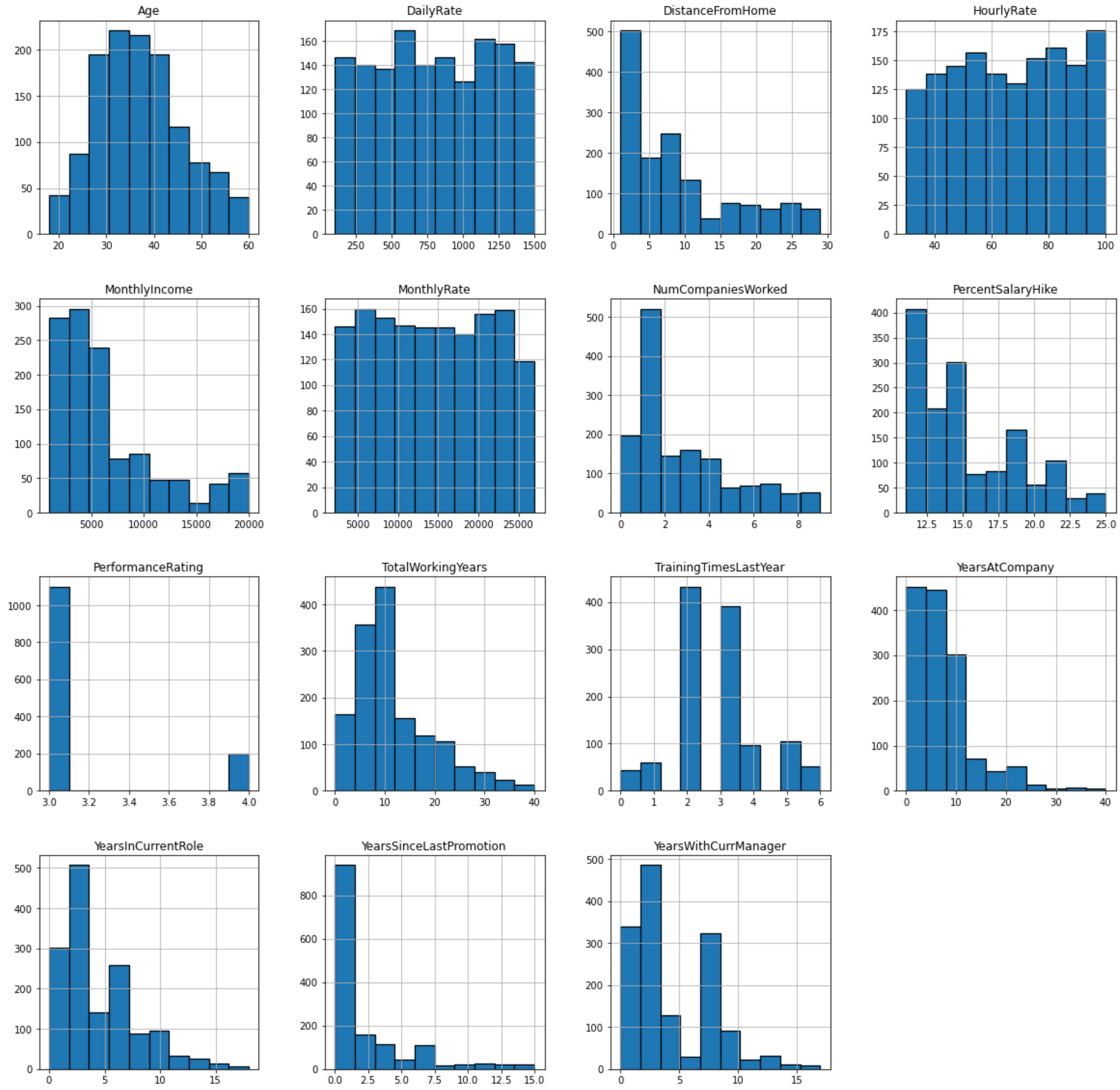
First of all, let’s analyze the main attribute for our study; Attrition:





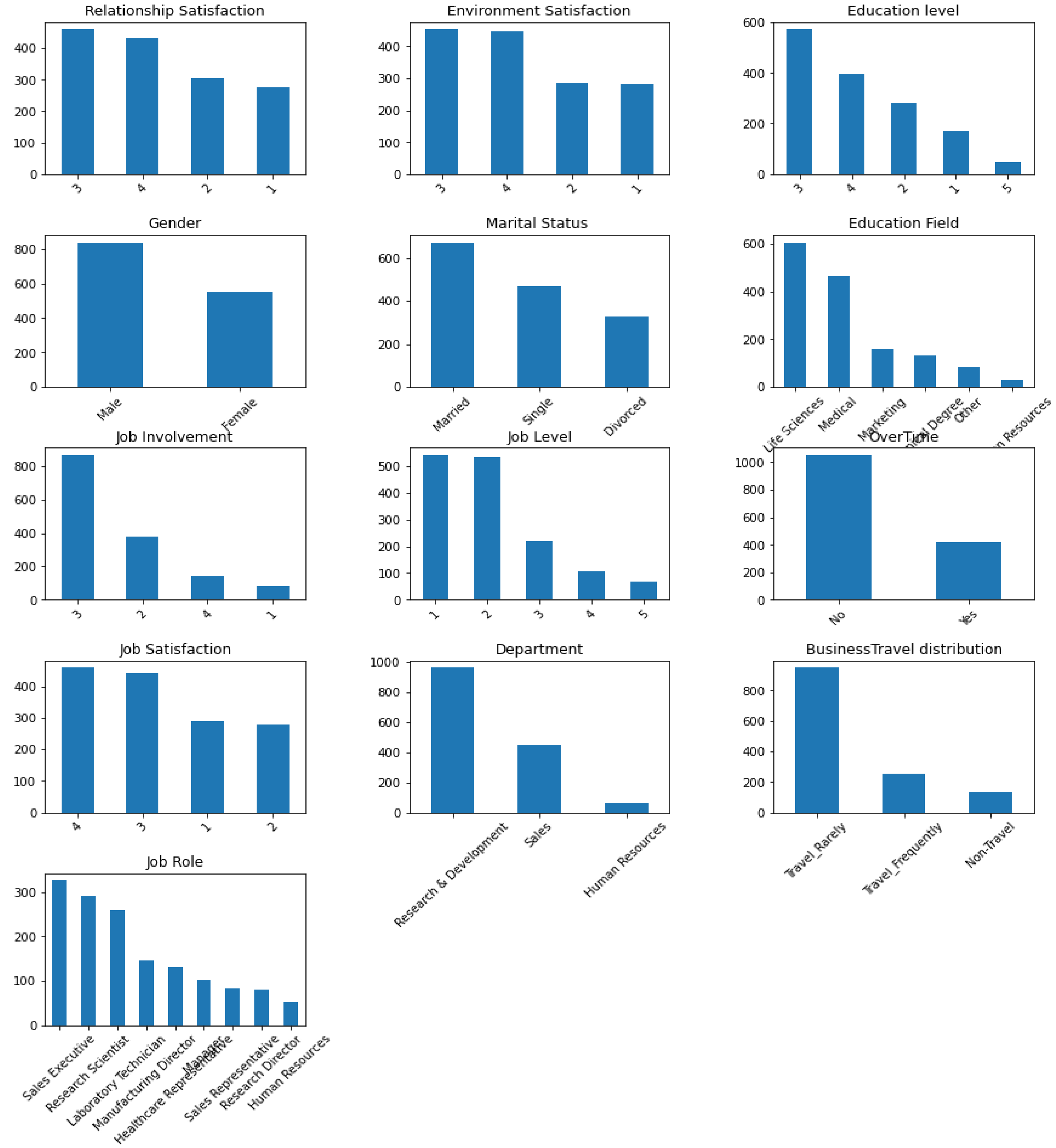
As we can see from the above plots, most of the people don’t go for attrition. 1233 employees over 1470 don’t have any issue at work, which is 84% of the total. Our job is to find out what are the reasons that 237 employees expressed attrition during their work? We can consider our dataset to be imbalanced since more people stay in the organization than they actually leave.

We start from visualizing the distribution of numerical data using histograms:



From the above histograms, we can have an idea of the nature of the employees with respect to that particular attribute. We can see that the average age is between 30-40 years and most of the people in this dataset live in close proximity to the office. Regardless of the daily, hourly or monthly rate, mostly workers are earning less when we see the monthly income. When we see the work history, most of them have changed their company less than 2 times only. A salary hike of 12-14% is common with a performance rate of 3. Furthermore, from YearsAtCompany, YearInCurrentRole, LastPromotion and CurrManager, we can see the behaviour of service of employees like how long they have been working in their respective role and their promotion etc.

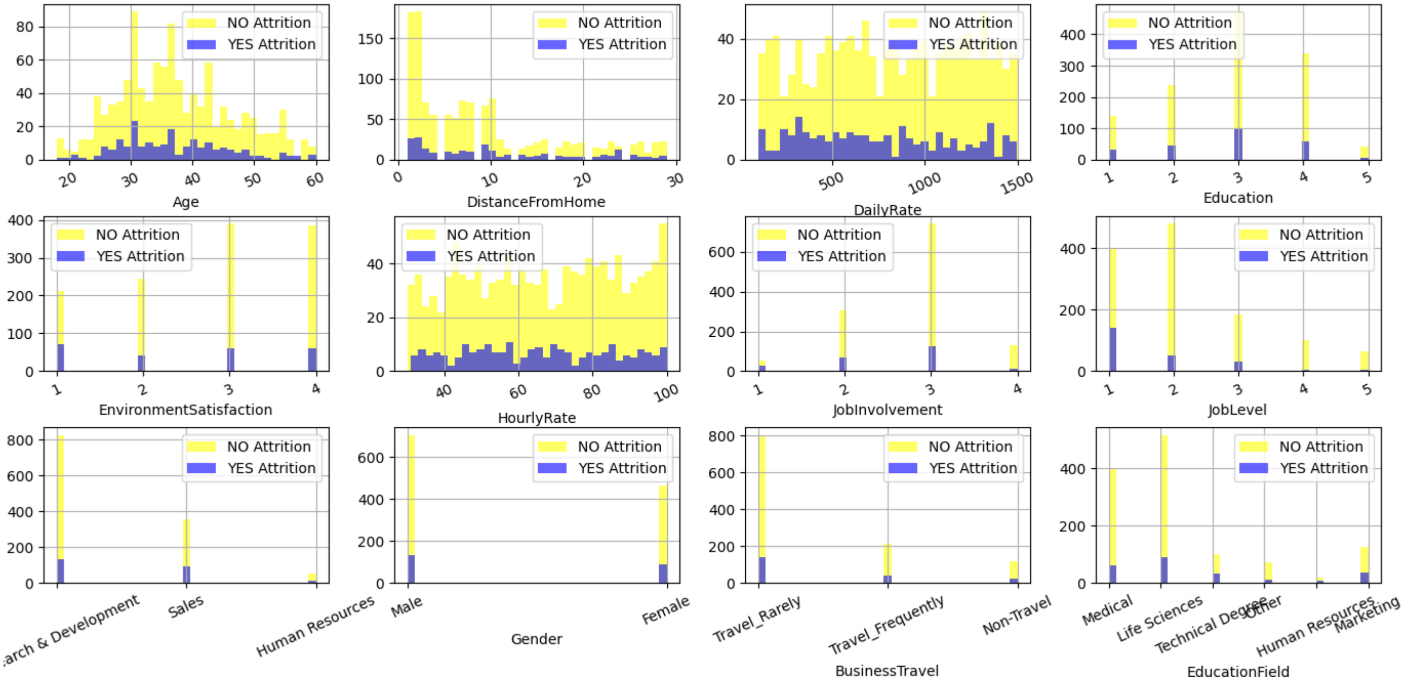
Now let’s plot the categorical variables using histograms:

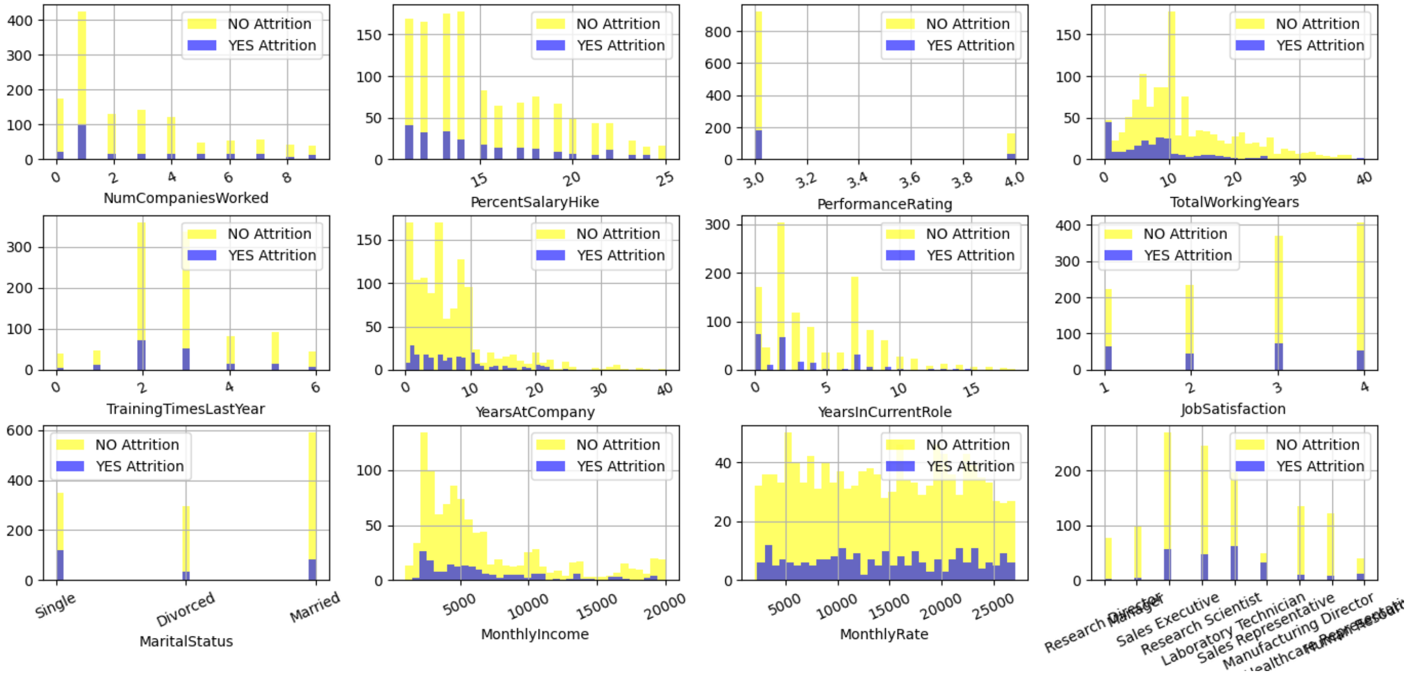


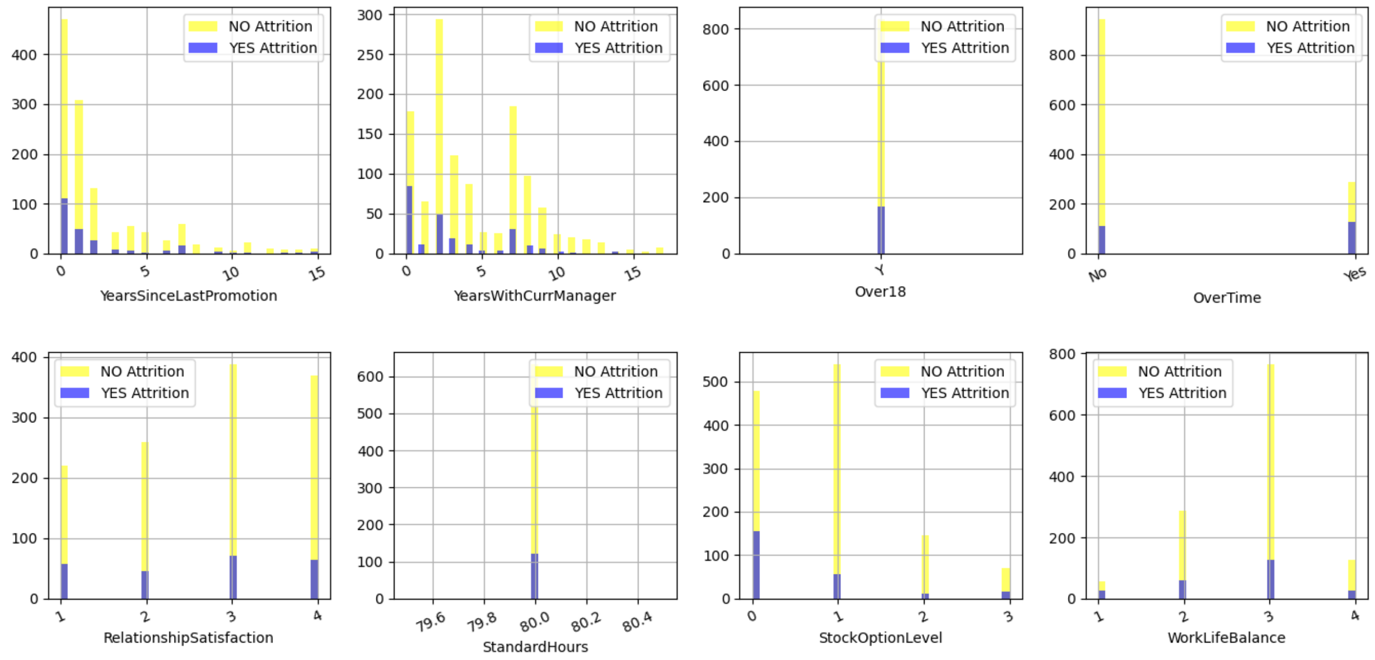
The plots of categorical attributes say that relationship and environmental satisfaction is high in the company. Most of the employees have a bachelor's degree and there are more male employees than females.

Most workers in the dataset are married and rarely travel. Mostly employees have the life sciences and medical background and there are comparatively less people from technical or rest of fields. Furthermore, IBM’s main focus is on research and development as it has the largest team. Most of the employees have high job involvement but low job levels and the company has the highest number of sales executives as compared to other roles and mostly are satisfied with their jobs.

Now let’s analyse each attribute according to our targeted variable; Attrition:







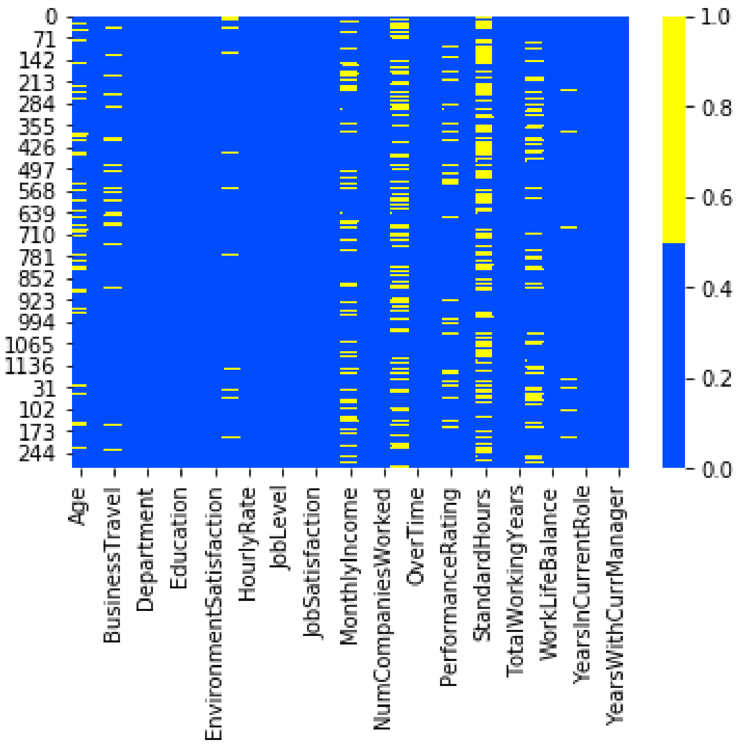
Almost, 25 % of the people who leave are about 30 years old and their percentage falls below 40 years of age whereas a significant percent of people who do not leave the company belong to an age group of 30 to 45 years as described by the plot. There is a higher number of people who reside near to offices and hence the attrition levels are lower. With the increase in distance, the attrition curve overtakes the No attrition curve which is expected. So we can say that distance affects the attrition of employees. When it comes to education people with 3rd level of Education usually go for attrition and meanwhile, Education Field portrays, workers with Human Resources and Technical Degree are more likely to quit then employees from other fields of Education. According to the graph of Gender, males are more likely to quit. As the BusinessTravel plot says, those who travel rarely are more likely to quit then other employees. Department plot says, employees in Research & Development are more likely to stay then the workers in other departments. The workers in Laboratory Technician, Sales Representative, and Human Resources are more likely to quit the workers in other positions, says JobRole plot. When we see the MaritalStatus, singles are more likely to go for attrition than married, and divorced. And finally, OverTime says employees who work more hours are likely to quit then others. Then comes the pay rates i.e hourly, daily or monthly one and the plots above show there is a difference between the attrition with daily rates. Attrition is higher for lesser daily rates and of course no attrition for higher daily rates. It's almost the same with the Hourly Rate, just a slight different. The trend is resonating with monthly income too. After seeing the plot of Total Working Years, people with less experience mostly go for attrition and a significant percentage falls within 10years but the same percentage also doesn't go for attrition. People with high job involvement have higher attrition rates followed by medium involvement people also when training(s) are less attrition is seen highest. Looking at the plots of Overtime and Work life balance, those who are forced to do overtime go for attrition and a work life balance of 3 are more likely to quit their jobs. People with lesser years in current roles mostly go for attrition along with the more like 15years. After summarising the rest of the graphs, it seems that the workers with low JobLevel, MonthlyIncome and YearAtCompany are more likely to quit their jobs. Whereas EnvironmentSatisfaction, JobSatisfaction, PerformanceRating, and RelationshipSatisfaction does not impact much on the determination of Attrition of employees as compared to the other variables.

3. Assessing Data Quality

Missing Values

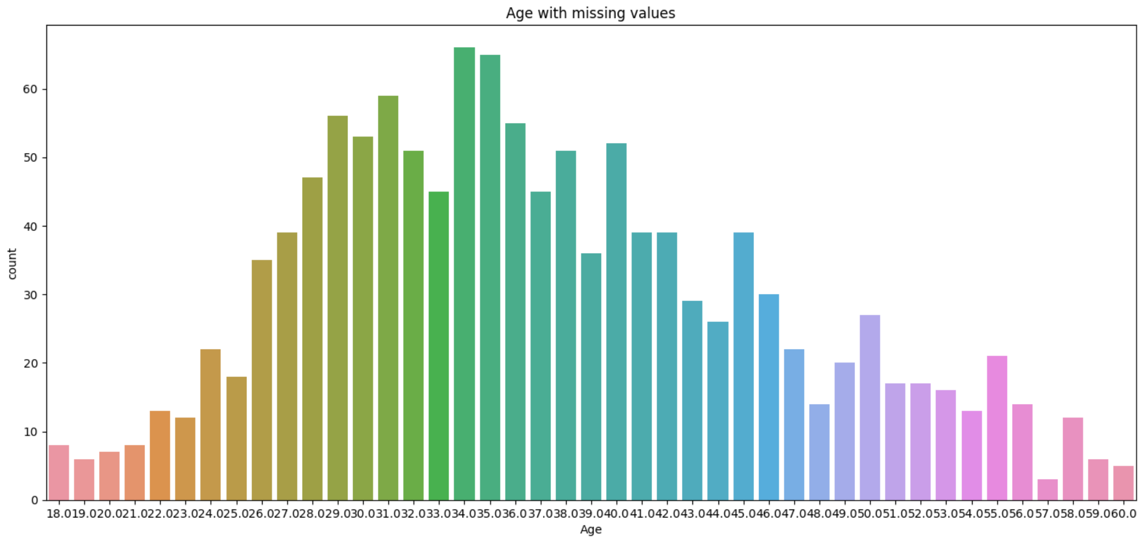
Now let's consider the quality of our available data. First of all we plotted the missing values, stated with yellow, for each variable in the following plot, then we resumed in the table the main variables with missing values.

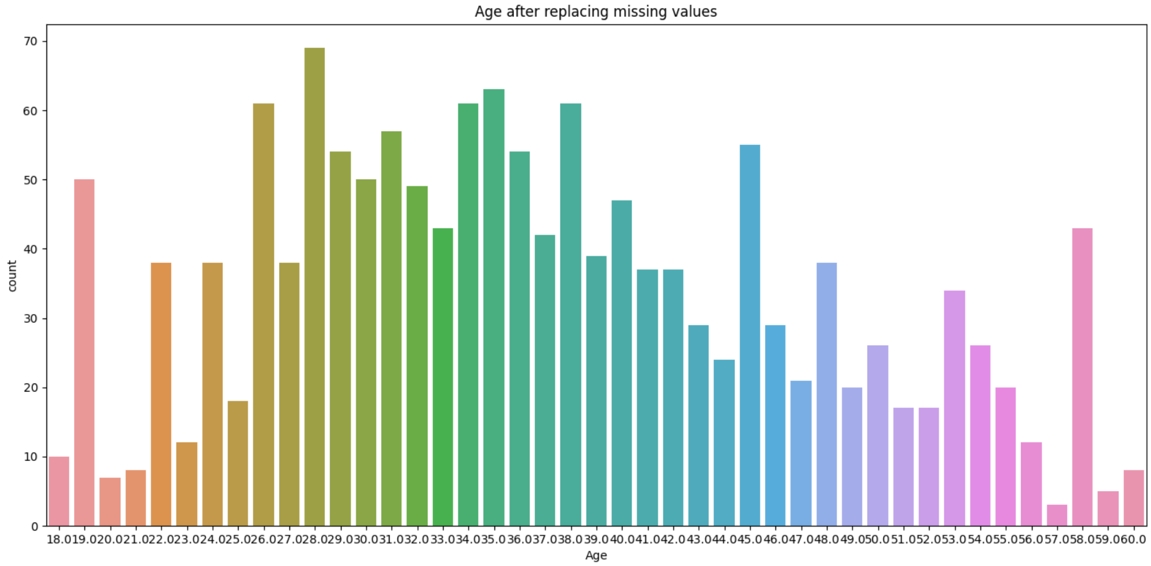
Variables	Missing Values %
Age	14%
BusinessTravel	9%
MonthlyIncome	19%
StandardHours	49%
YearsAtCompany	5%
TrainingTimesLastYear	20%
Gender	5%
PerformanceRating	12%
Over18	32%



We decided to handle the missing values in the previous variables with the following rules:

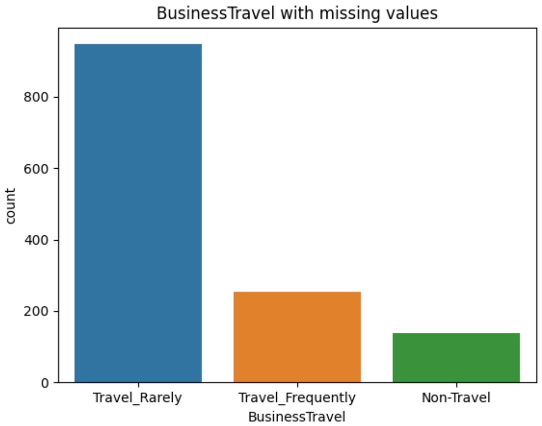
- YearsAtCompany: Since there are only 74 missing values we decided to replace them with with the mean
- PerformanceRating: the distribution of values in these variables is unbalanced, so we decided to add all the few missing values to the most frequent values: “3.0”.
- Gender: the distribution of values in these variables is unbalanced, so we decided to add all the few missing values to the most frequent values: “Male”.
- Age: according to the correlation matrix the age is closely related to “MonthlyRate” and to “YearsAtCompany”, this means that the age of an employee is linked to his monthly salary and the years spent in the company. MonthlyRate has too many missing values and for this reason we estimated the missing values according the values of the variable ‘YearsAtCompany’

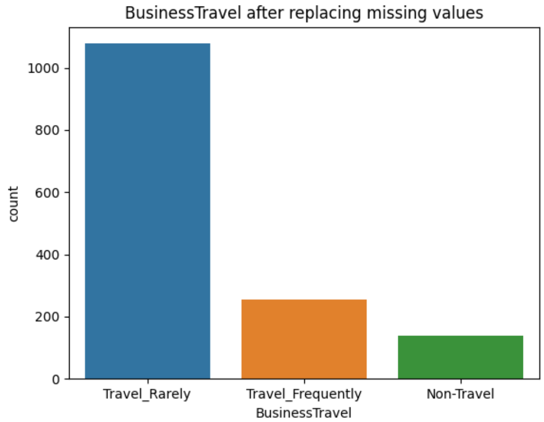




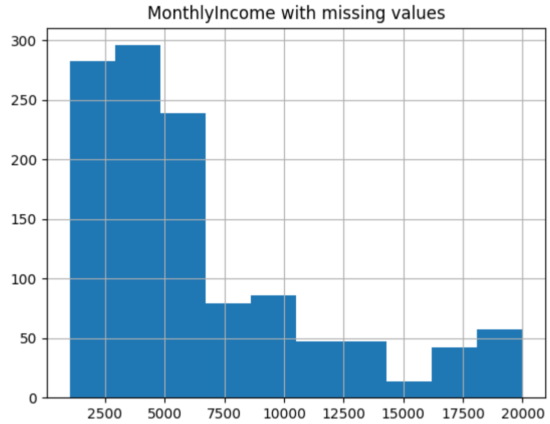
- StandardHours: is the variable with most missing values, almost the half (49%) then we decided to completely eliminate it

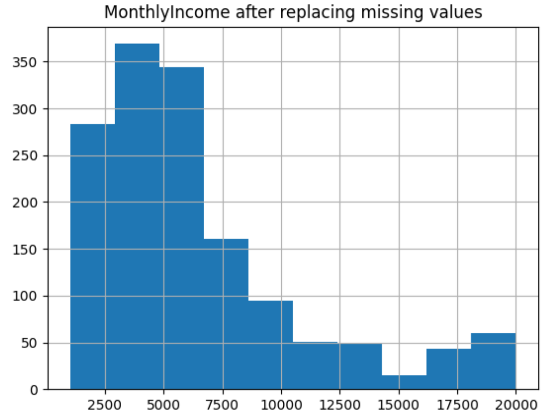
- BusinessTravel: the distribution of values in these variables is really unbalanced, so we decided to add all the missing values (only 9%) to the most frequent values: “Travel_Rarely” that occurs more than 70% of values.





- We decided to fill the missing values of the ‘MonthlyIncome’ variable according to the values of ‘YearsAtCompany’ since we have noticed from the correlation matrix that they are correlated. For each value of ‘YearsAtCompany’ we calculated the mean of Income earned and then we just replaced these means with missing values of ‘MonthlyIncome’. Here the results:

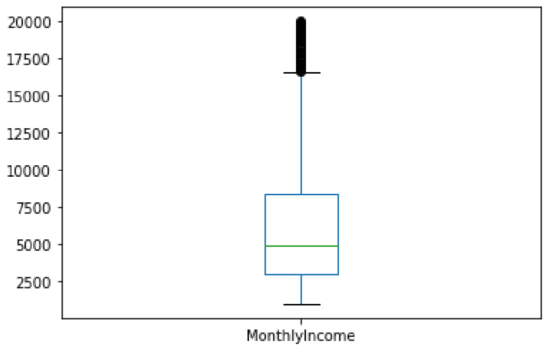


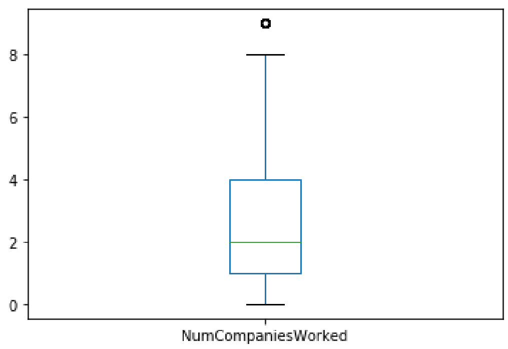


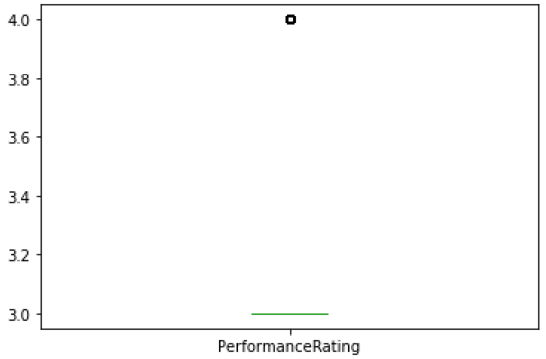
- We decided to do not use the ‘TrainingTimesLastYear’ and ‘Over18’ variables in our study

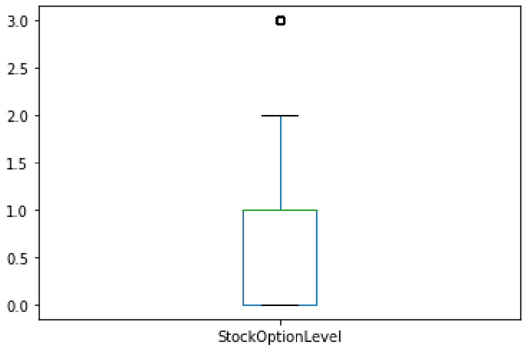
Outliers

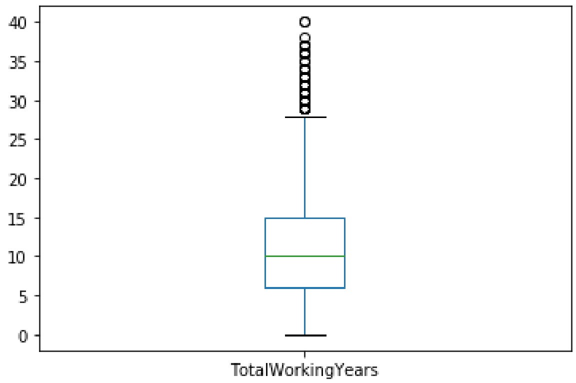
To find out the outliers we decided to use boxplots. We delete outlier values if it is due to data entry error, data processing error or outlier observations are very small in numbers, but deleting the observation is not a good idea when we have a small dataset like this.

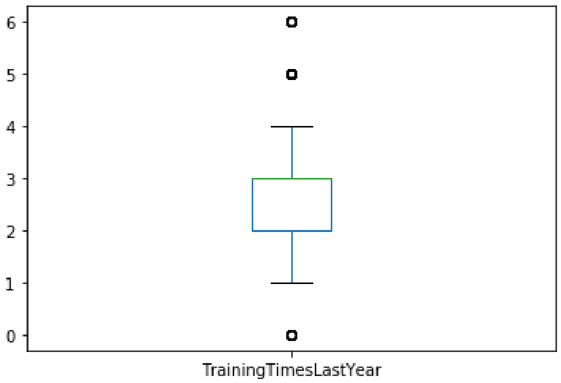

20000
17500
15000
12500
10000
7500
5000
2500
MonthlyIncome

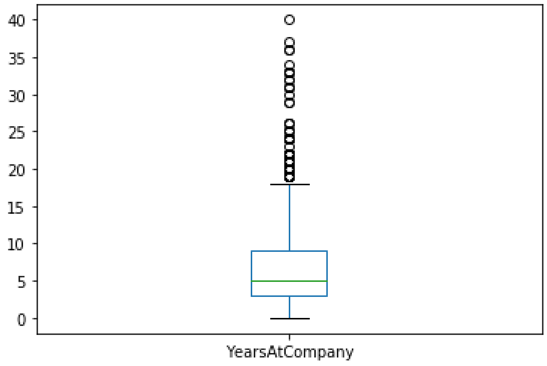

8
6
4
2
0
NumCompaniesWorked

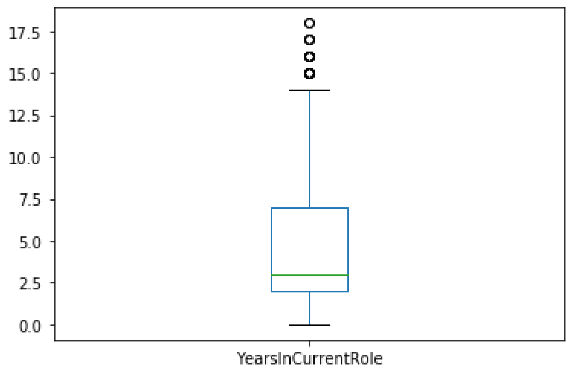

4.0
3.8
3.6
3.4
3.2
3.0
PerformanceRating

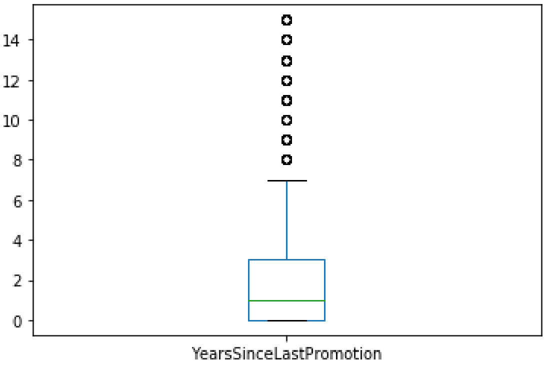

3.0
2.5
2.0
1.5
1.0
0.5
0.0
StockOptionLevel

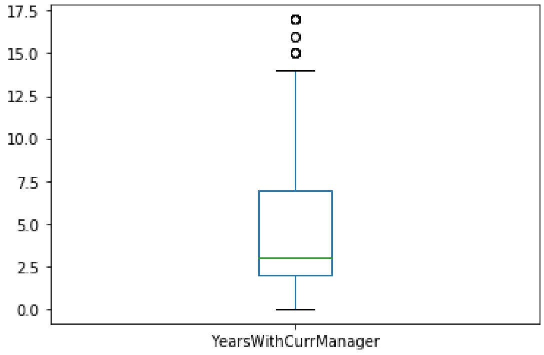

40
35
30
25
20
15
10
5
0
TotalWorkingYears


6
5
4
3
2
1
0
TrainingTimesLastYear









As we can see from the boxplots above, all the variables involved with outliers should have some instances that are different from most of the other data objects in the data set, for instance, is reasonable to think that only few people in the business earn a higher monthly income or that only few people have the higher StockOptionLevel, so these outliers may be of interest in our study.

For this reason we decided to treat outliers by transforming variables. These transformed values reduce the variation caused by extreme values.

4. Variable Transformation

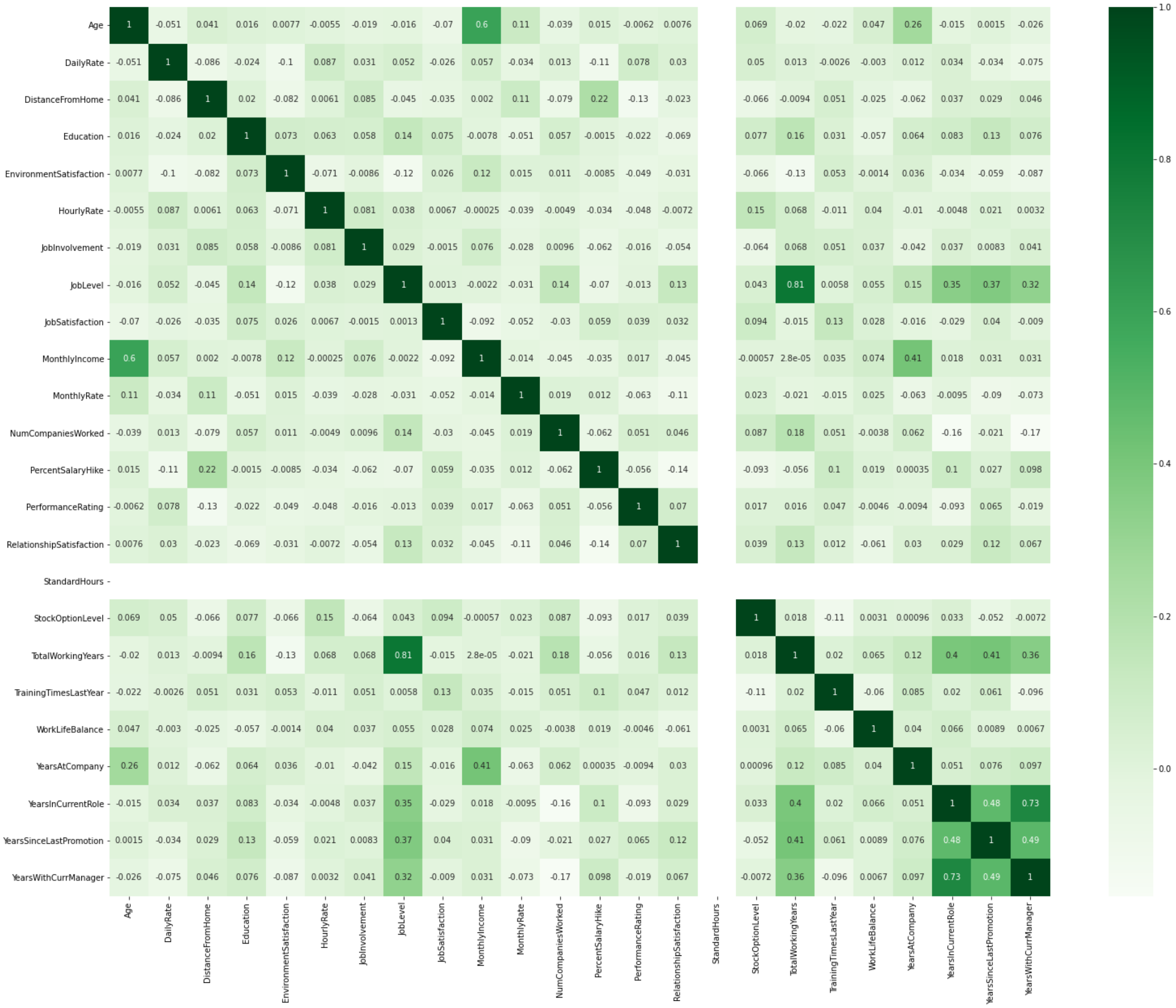
In a real world dataset, you can not and should not expect proper distribution of data because of the presence of outliers, errors, missing values and such. Hence, to make sure that this is not a problem when we use the data to make predictions, it is necessary to transform them into data that fits a defined statistical model reducing skew from the data distribution.

For our dataset, we use the 'fit_transform' method provided by pandas for its data frame objects to normalize the data. The method basically calculates the mean and sd of the given data and uses it to normalize the given set. For this, it is necessary to run the transformation for the entire data set including the training, test, and any new data we may encounter. In our case, pandas calculates these values using the 'fit' method, then later applies the transformation using the 'transform' method.

We have applied this function on the data set for the clustering and KNN classification since the underlying algorithms for these use the euclidean distance which is useless if the data is not scaled or transformed well. As for the rest of our project, we have used the original data set without any global transformation, except for the fact that we have filled the missing values.

5. Pairwise correlations and eventual elimination of redundant variables

We can see that job level is highly correlated with age and monthly income and so as monthly income is highly correlated with age and job level. Thus all are positively correlated. We can also see that performance rating is highly correlated to percent salary hike. Moreover, total working years, years at company, years in current role, years since last promotion and years with current manager are inter correlated.



As can be seen from the correlation matrix, we find a correlation among the following variables which we have categorised into two groups based on the correlation score.

Strong correlation (0.80+) - JobLevel and TotalWorkingYears

Weak correlation (0.60 - 0.80) - Monthly Income and Age, YearsInCurrentRole and YearsWithCurrManager

Hence, we drop JobLevel.

Clustering

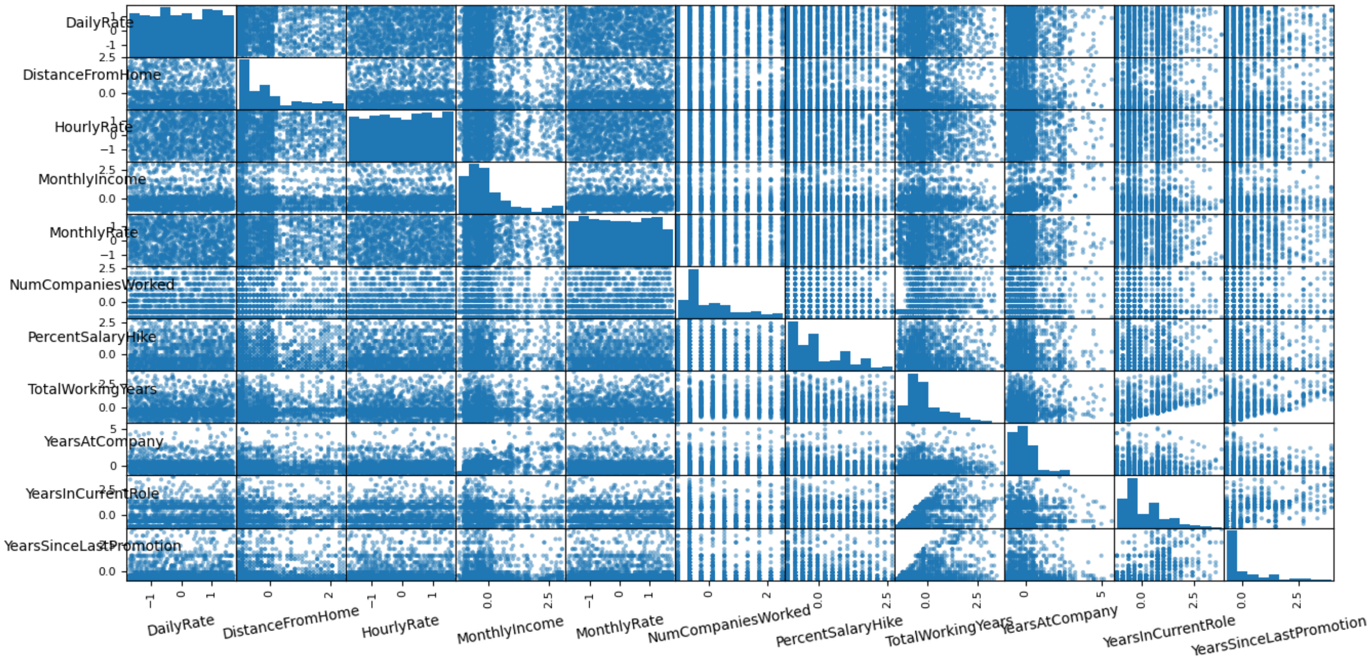
Kmeans

For the kmeans clustering analysis we decided to use only the numerical and discrete variables because the cluster analysis is strictly correlated to the notion of similarity. The similarity between two objects is a numerical measure of the degree to which the two objects are alike. For categorical and ordinal variables it's hard to define an order of similarity between two objects e.g. Is the difference between the values *fair* and *good* really the same as that between the values *OK* and *wonderful*?

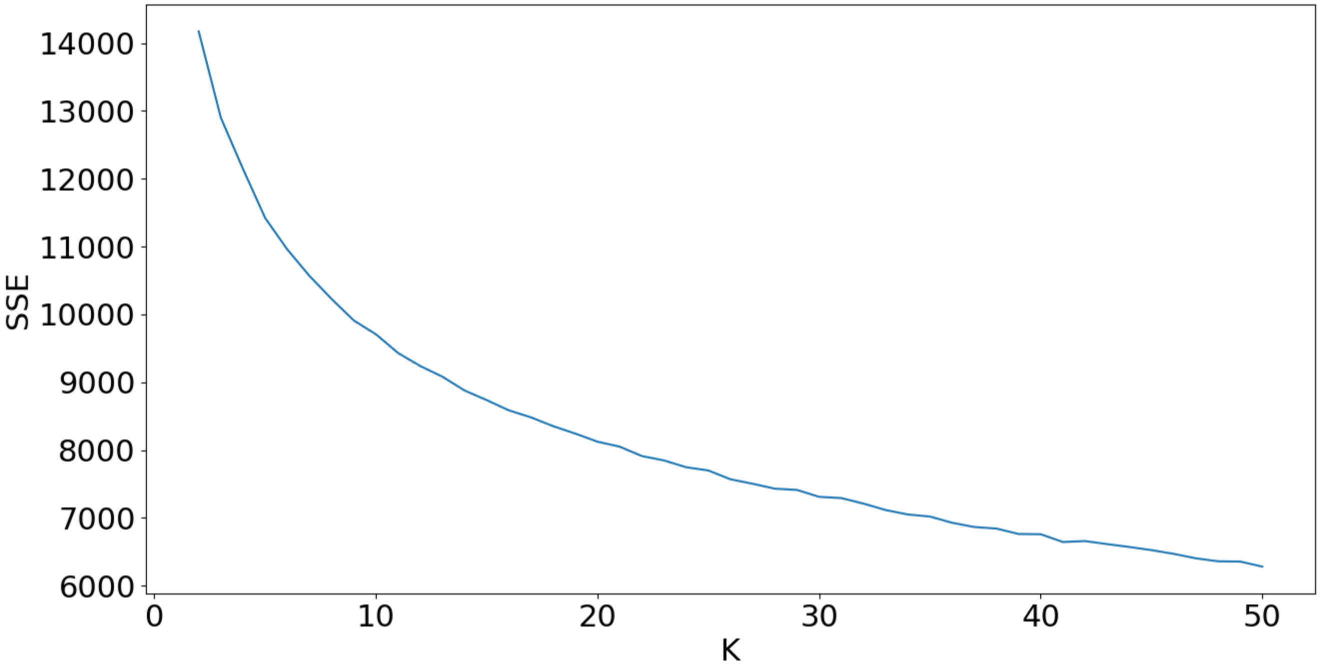
The variables selected for our analysis are: DailyRate, DistanceFromHome, HourlyRate, MonthlyIncome, MonthlyRate, NumCompaniesWorked, PercentSalaryHike, TotalWorkingYears, YearsAtCompany YearsInCurrentRole, YearsSinceLastPromotion.
We decided to remove Age and YearsWithCurrManager, weakly correlated to respectively MonthlyIncome and YearsInCurrentRole.

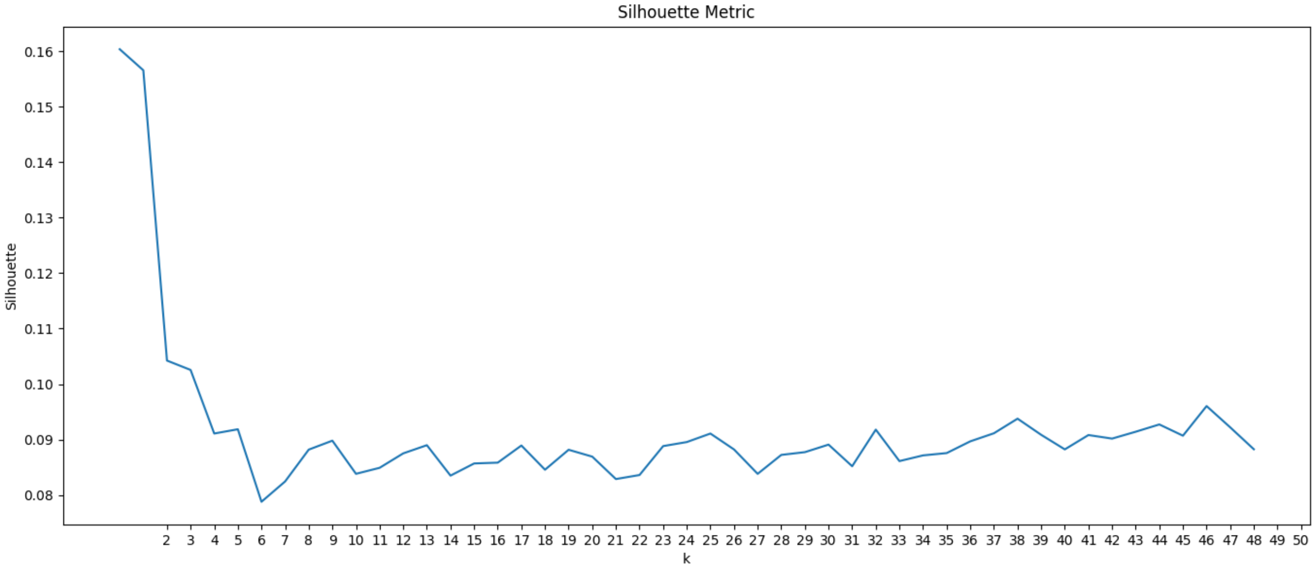
The distance function chosen is the classic euclidean distances.

We resumed the relationship between our variables in the following scatter plot:



To choose the number of clusters we used the “Elbow Method”. We iterated the kmean algorithm over our selected variables 50 times by varying the number of clusters every time. We saved all SSE (sum of the squared distance) obtained for each time, the result is the following plot:



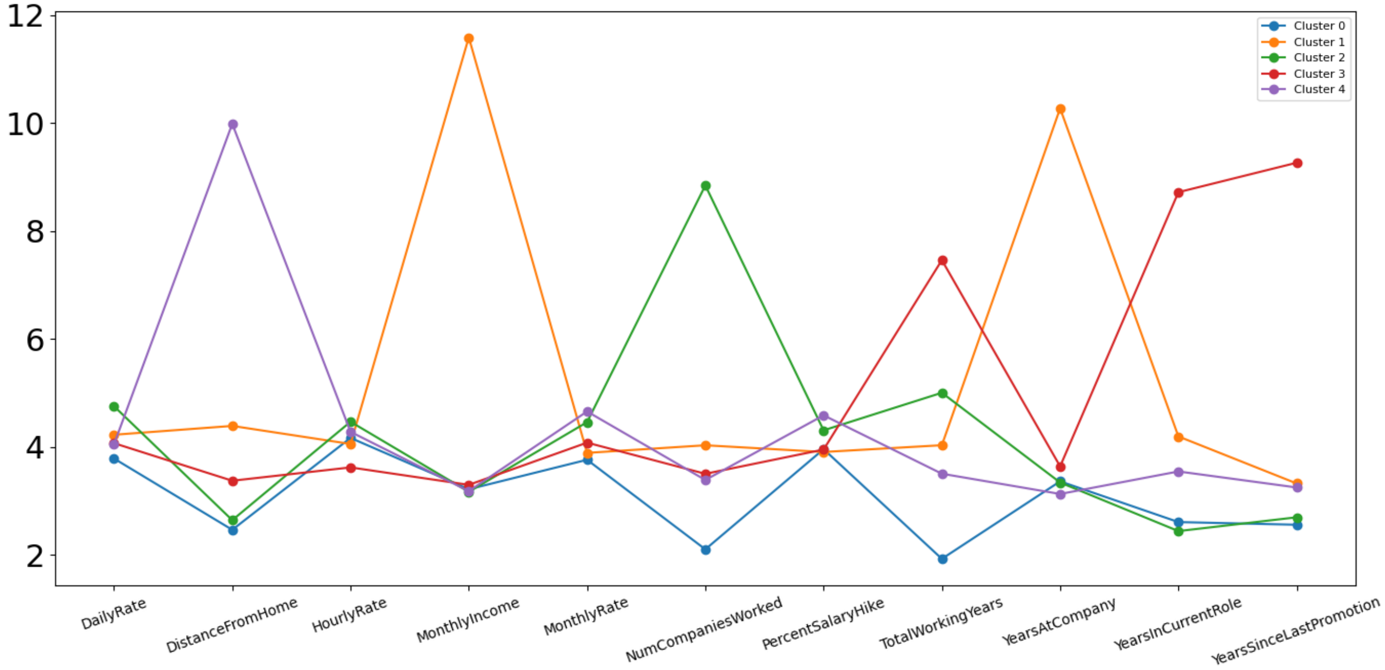


The intuition is that increasing the number of clusters will naturally improve the fit (SSE decrease), since there are more clusters to use, but that at some point this is over-fitting, and the elbow reflects this. To help in this decision we calculated also the silhouette coefficients that combine both cohesion and separation. As a compromise we decided to use k = 5.

We applied the K means method and we obtained the following results:

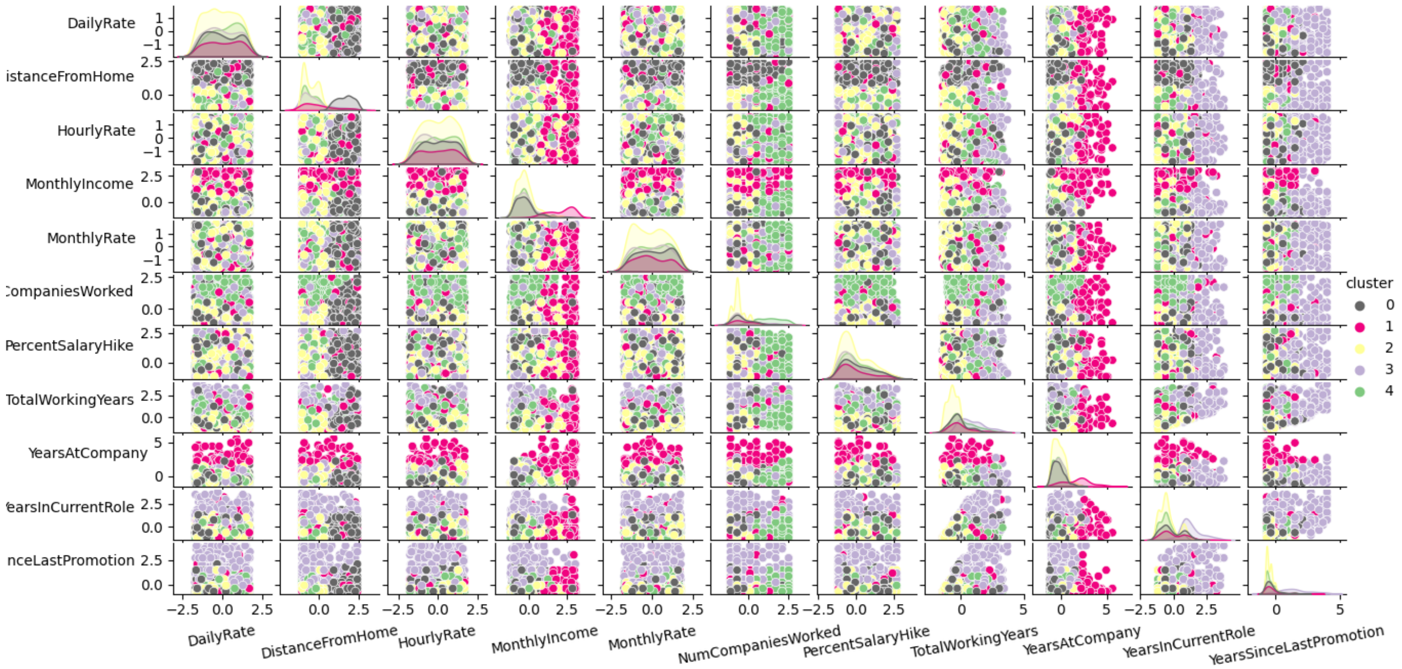
Dimensions of clusters: Cluster 0: 243, Cluster 1: 158, Cluster 2: 487, Cluster 3: 289, Cluster 4: 293

We plotted the centroid of each cluster for all our variables, obtaining the following plot:



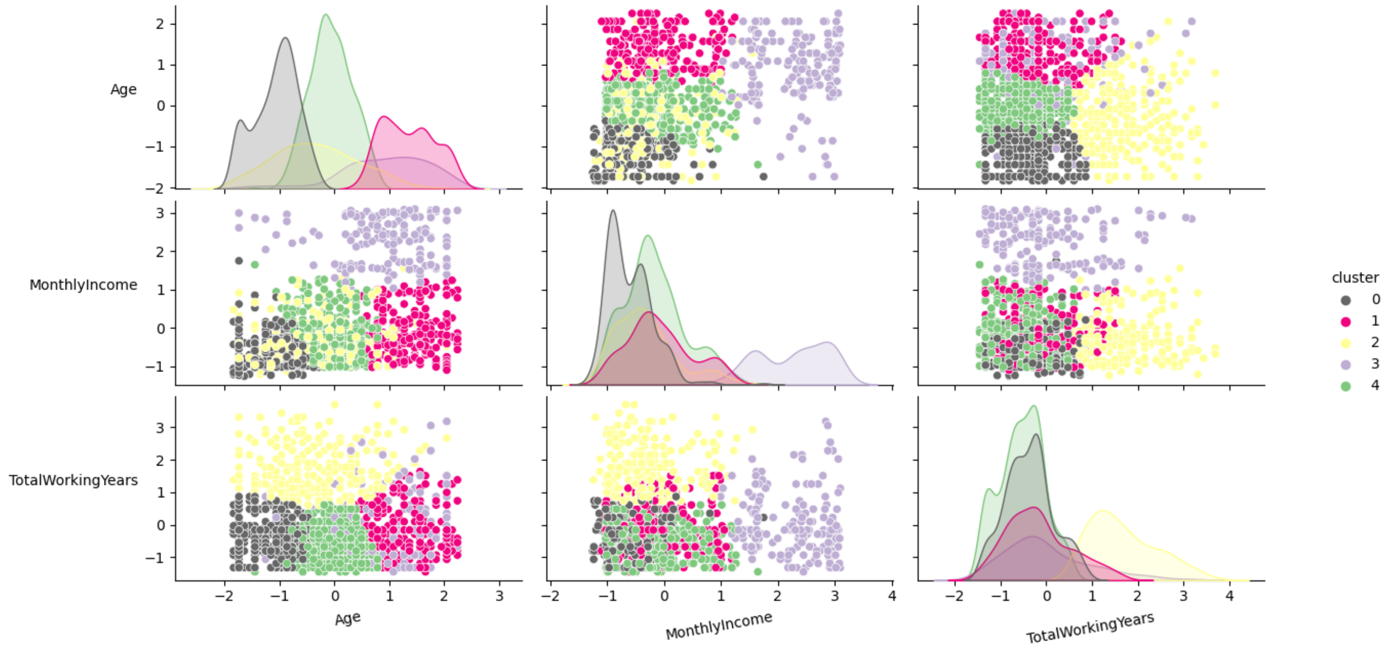
As we expected MonthlyIncome, NumCompaniesWorked, YearsAtCompany, YearsInCurrentRole and YearsSinceLastPromotion have been affected by outliers. Some centroids tend to be close while others really far apart. Cluster 1 took all the outliers of MonthlyIncome and YearsAtCompany while other centroids seem really close because data are concentrated in a small zone since most people have low income and worked for a few years.

DailyRate, HourlyRate and MonthlyRate have centroids equally distributed as we can see from their distribution.



Final SSE: 11420.680471384638
Final silhouette: 0.10239654849719229

All these dimensions makes the k means algorithm really complicated to interpret, for this reason we decided to apply it again, to a smaller set of features that showed interesting features in the distribution analysis. The variables are: Age, TotalWorkingYears and MonthlyIncome. We obtained the following scatter plot:



The new set of clusters have different numbers of instances: Cluster 0: 361, Cluster 1: 247, Cluster 2: 245, Cluster 3: 174, Cluster 4: 443.

Let's analyze the distribution of Yes Attrition among those clusters since our main aim is to obtain more information about the attrition phenomenon:

- Cluster 0: 61 instances in 351 (61/361=0.169)
- Cluster 1: 39 instances in 173 (39/247=0.158)
- Cluster 2: 20 instances in 247 (20/245=0.082)
- Cluster 3: 22 instances in 247 (22/174=0.126)
- Cluster 4: 95 instances in 452 (95/443=0.214)

We can see that the cluster that has the most Yes Attrition values is the cluster 4. As we can see from the scatter plot the cluster 4 took all the instances that have lower MonthlyIncome, low value of TotalWorkingYears and middle values of Age. We can say that most of the employees that found attrition might have these characteristics.

Another important observation can be done on cluster 2, that has the higher rate of No Attrition instances (225 in 245), all the instances in this cluster seem to have the same characteristic, they have a high value of TotalWorkingYears.

Final SSE: 1439.8383859309097

Final silhouette: 0.2728201725682409

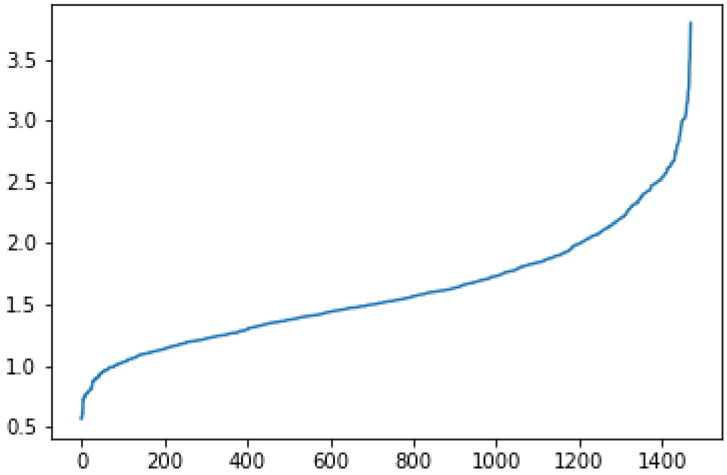
DBSCAN

The choice of attributes for DBSCAN is the same as the ones for kmeans, since we wanted to maintain a uniformity between the chosen attributes for the three different clustering so as to compare them easily later. The chosen attributes are : DailyRate, DistanceFromHome, HourlyRate, MonthlyIncome, MonthlyRate, NumCompaniesWorked, PercentSalaryHike, TotalWorkingYears, YearsAtCompany YearsInCurrentRole, YearsSinceLastPromotion.

As for the distance function, we decided to go with the euclidean distance since this gives us the regular distance between two points and is helpful and simple to identify the distance in an n dimensional space. Furthermore, we ran all the iterations described below with six different distance functions, namely 'cityblock', 'cosine', 'euclidean', 'l1', 'l2', and, 'manhattan', and the best result was given by the euclidean distance.

For the DBSCAN, we used the implementation of 'sklearn.cluster' package. The DBSCAN algorithm has two hyper parameters, namely epsilon (ε) and min_samples . Basically, the algorithm starts with a point in the dataset and retrieves all the points with euclidean distance less than or equal to epsilon from the starting point. Then it checks if the number of points retrieved is greater than or equal to min_samples, if it is, the points are considered part of the same cluster, or else. Hence, these parameters determine how many and what falls into a cluster and is of critical importance to the algorithm.

As per the paper cited in the footnote of this page, we know, if the data has more than 2 dimensions, choose MinPts = 2*dim, where dim= the dimensions of your data set (Sander et al., 1998). Since the dimension of our dataset is 11, we set the parameter minimum points to 22 (2 * 11). Using this value as the number of neighbors for the KNN algorithm, we are able to generate a k-dist plot which should help us get a suitable value for epsilon using the elbow method. The generated k-dist plot is as follows:



We see that the elbow (maximum slope) is in the vicinity of 2.4 to 2.8. After running some tests around this value, we realize that the optimal value for epsilon is 2.6.

Hence, we are able to determine the optimal values of epsilon and minimum samples as 2.6 and 22, respectively. We use these values to run the DBSCAN algorithm and get a the following clusters:

Cluster 1 : 230 points, and, Cluster 2 : 1240 points

From this clustering, we are able to see the following observations:

Cluster 1:

Number of Yes values for Attrition = 35

Percentage of total Yes values for Attrition in dataset = 14.77

Number of No values for Attrition = 195

Percentage of total No values for Attrition in dataset = 15.81

Cluster 2:

Number of Yes values for Attrition = 202

Percentage of total Yes values for Attrition in dataset = 16.40

Number of No values for Attrition = 1038

Percentage of total No values for Attrition in dataset = 84.18

Final silhouette score: 0.2230466206252839

Citation for this page:

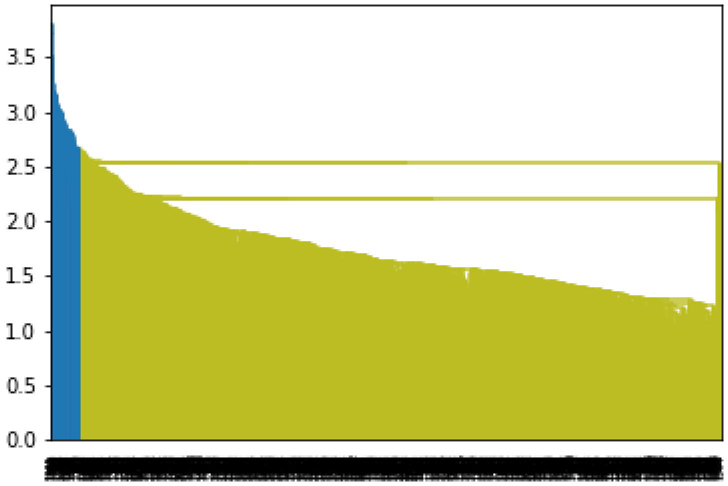
Sander, J., Ester, M., Kriegel, HP. et al. Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications. Data Mining and Knowledge Discovery 2, 169–194 (1998). https://doi.org/10.1023/A:1009745219419

Hierarchical Clustering

Hierarchical Clustering can be implemented using different metrics to define the inter-cluster similarity/difference. For this report, we have explored some of the most popular ones, namely 'Ward's method', 'average linkage', 'complete linkage', and 'single linkage'. There are two types of hierarchical clustering depending on the way you move: bottom up, or, top down. The bottom up way is also called Agglomerative Clustering, in which we start with every point as an individual cluster and we move forward merging any two clusters given by the chosen similarity score into one. In our analysis, this is the method that we have chosen to use.

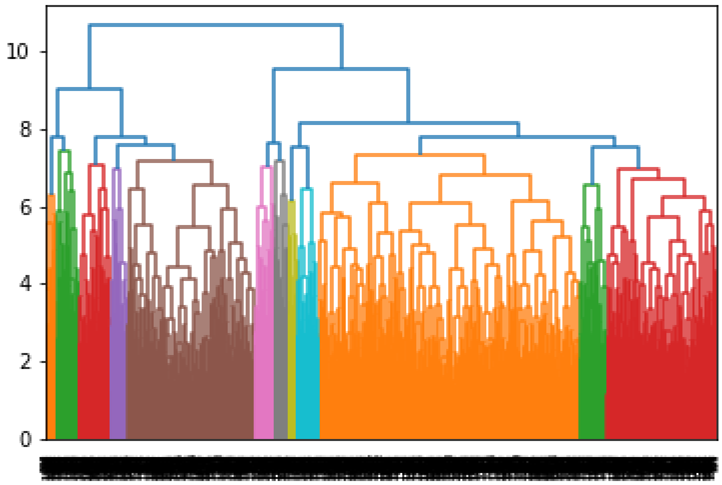
Principally, there are two parameters that the algorithm depends on: number of clusters, and linkage heuristic. We will be analyzing all four of the aforementioned linkage heuristics in our report, and for every one of them, we will be plotting out a dendogram to try and find the optimal number of clusters.

a. Single Linkage:



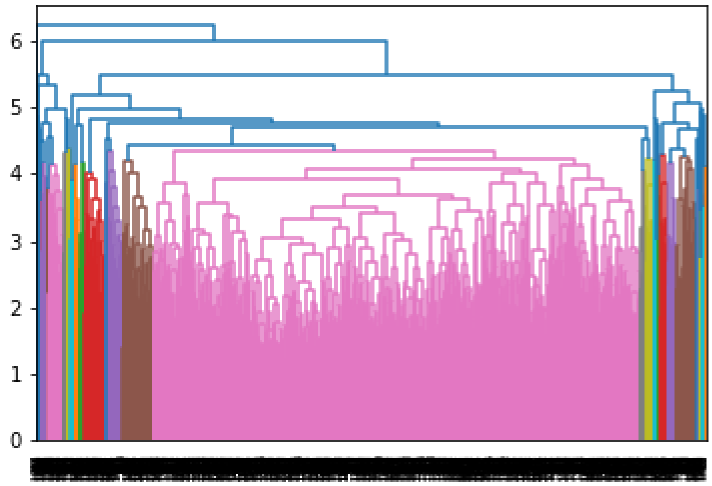
From the dendrogram, we can immediately conclude that single linkage is a poor heuristic as we are not able to get any clear distinction from the plot. We can see two separate clusters but one of them is disproportionately large and skewed, providing little entropy.

b. Complete Linkage:



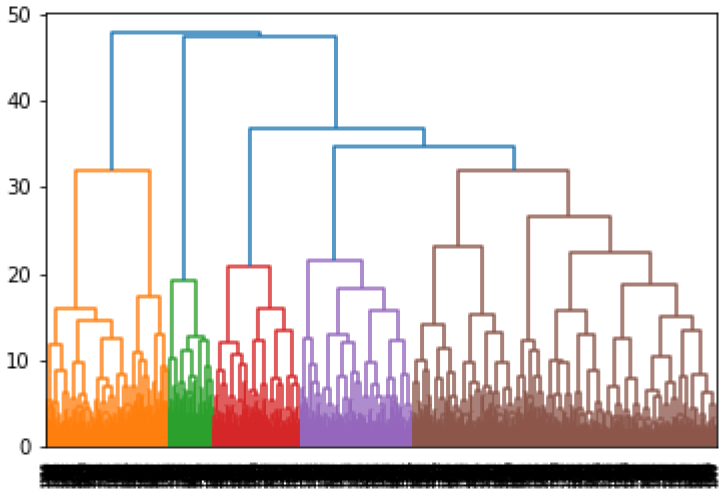
From this dendrogram, we are able to see that there are around 12 clusters that can be formed (based on the color). Running the AgglomerativeClustering algorithm provided with the sklearn package, with parameters ‘complete’ for linkage and ‘12’ for number of clusters, we get a silhouette score of 0.048.

c. Average Linkage:



From this dendrogram, we are able to see that there are around 20 clusters that can be formed (based on the color). Running the AgglomerativeClustering algorithm provided with the sklearn package, with parameters ‘average’ for linkage and ‘20’ for number of clusters, we get a silhouette score of 0.065.

d. Ward’s method:



From this dendrogram, we are able to see that there are around 5 clusters that can be formed (based on the color). Running the AgglomerativeClustering algorithm provided with the sklearn package, with parameters 'ward' for linkage and '5' for number of clusters, we get a silhouette score of 0.067.

Evaluation of the clustering approaches and comparison of the clustering obtained

First of all, we can say that DBSCAN does not seem to be a good clustering approach from our analysis. This may be because of the huge dimension of the dataset, which is a known problem for DBSCAN. If we look at the clusters for the parameters obtained from tuning, we see there are just two clusters formed and both of them have around 18 percent of the values that say Yes to attrition. This indicates that the clustering does not result in much entropy i.e. information gain.

The other approaches are K-means and hierarchical clustering. We ran both of them for the same set of attributes so as to maintain uniformity in the results. For the hierarchical, we ran it using four different methods and except one, the others showed similar performance when evaluated using their silhouette score.

The silhouette scores are listed below:

	Approach	**Silhouette Score**
1	K-means clustering	0.102
2	Hierarchical clustering : complete linkage	0.048
3	Hierarchical clustering : average linkage	0.065
4	Hierarchical clustering : Ward's method	0.067

As we can see from the table, k-means clustering yields the best score among the analysed algorithms. Furthermore, we decided to select just three attributes which seemed the most relevant from the data distribution and ran a k-means clustering on them. The results obtained are much more interesting with us being able to observe good distinction between the values for the chosen attributes. The results are discussed in the k-means clustering part, and the silhouette score obtained was 0.273 which is significantly better than the other approaches. However, it has its limitations as we cannot compare this result to the result of the other approaches since they consider a much higher dimensional data and are prone to the curse of dimensionality.

Classification

We are going to use our 'IBM HR Analytics Employee Attrition & Performance' dataset to build some classification models to represent the relationship between the attribute set and the class label. We can consider our dataset as a collection of records. Each such instance is characterized by the tuple (x,y) where x is the set of attribute values that describe the instance and y is our binary class label attrition. Our aim is to try to generalize the concept and try to obtain the best model to represent this relationship.
Our dataset is composed of 1470 entries, with a really skewed distribution of our class labels, in fact, there are 1233 entries with NO ATTRITION and 237 YES ATTRITION. We decided to use 80% of these entries (1176) for learning the model and the rest 20% (294) for the model evaluation. The distribution of our class label is preserved.

Some variables have been grouped in some ranges for best fit our dataset for the classification task. We transformed the following numerical and discrete variables into categorical (low, medium, high values): 'DistanceFromHome', 'YearsInCurrentRole', 'YearsWithCurrManager','DistanceFromHome', 'YearsInCurrentRole', 'YearsWithCurrManager', 'YearsSinceLastPromotion', 'YearsAtCompany'.
For tuning the parameters for the building of the model phase, we used the "RandomizedSearchCV" from sklearn. Our dataset has a lot of variables and we are not sure about the influence of each on the classification model, for this reason, we used a random search for its two benefits: a budget can be chosen independent of the number of parameters and adding parameters that do not influence the performance does not decrease efficiency.

Decision Trees

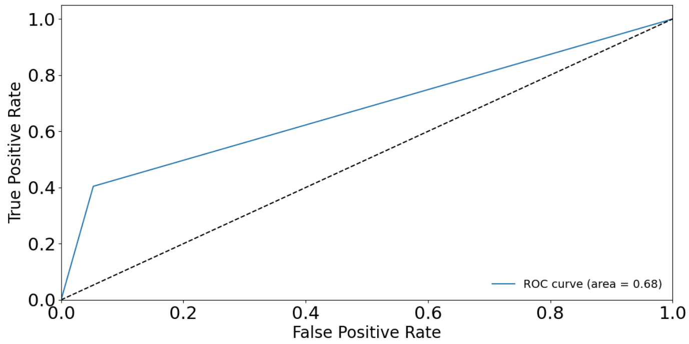
We built 4 classification trees, using two different impurity measures (entropy and gini index) with different sets of input values. We built two trees with the same parameters but different impurity measures (1-2) and other two trees with two different impurity measures, similar parameters but one with a small number of maximum depth and the other with no limitation (3-4).

We can resume the models in the following tables:

Impurity measure: Entropy		Impurity measure: Gini	
1 Parameters: {'min_samples_split': 5, 'min_samples_leaf': 20, 'max_depth': 7}	Training set: Accuracy 88% F1-score 0.87 Precision 0.87 Recall 0.88	2 Parameters: {'min_samples_split': 5, 'min_samples_leaf': 20, 'max_depth': 7}	Training set: Accuracy 88% F1-score 0.87 Precision 0.87 Recall 0.88
	Test set: Accuracy 86% F1-score 0.85 Precision 0.85 Recall 0.86 ROC curve 0.68		Test set: Accuracy 86% F1-score 0.85 Precision 0.85 Recall 0.86 ROC curve 0.68
3 Parameters: {'min_samples_split': 15, 'min_samples_leaf': 15, 'max_depth': 4}	Training set: Accuracy 87% F1-score 0.85 Precision 0.86 Recall 0.87	4 Parameters: {'min_samples_split': 15, 'min_samples_leaf': 15, 'max_depth': none}	Training set: Accuracy 88% F1-score 0.87 Precision 0.87 Recall 0.88
	Test set: Accuracy 86% F1-score 0.84 Precision 0.84 Recall 0.86 ROC curve 0.63		Test set: Accuracy 85% F1-score 0.84 Precision 0.83 Recall 0.85 ROC curve 0.64

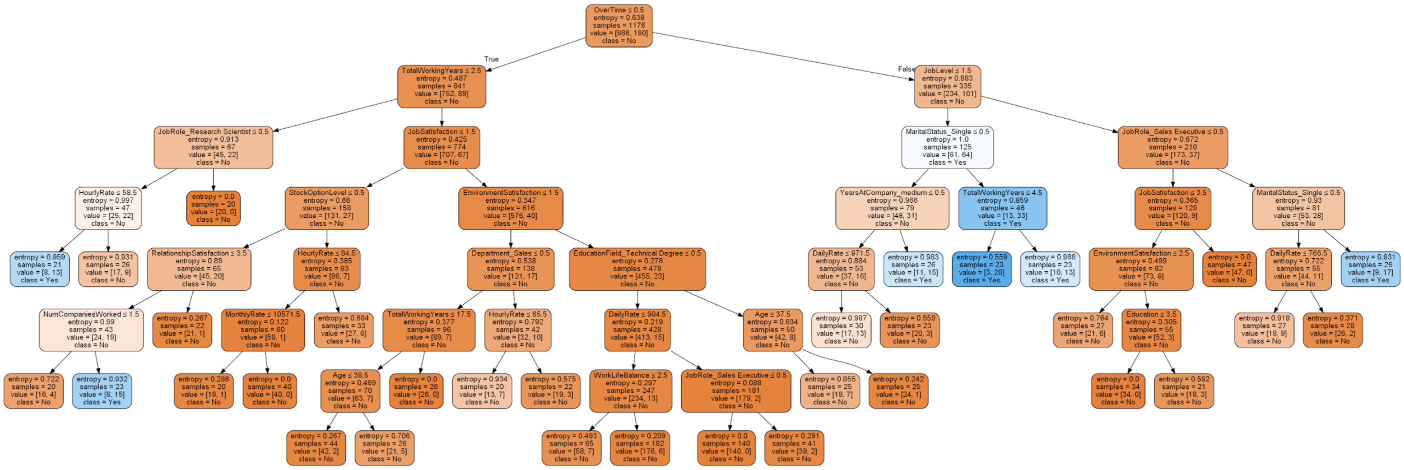
The most interesting results are reached by the first two trees. They both have the same results so the impurity measure didn't change the model selection at all. We reached a really high accuracy of 86% in the test set, on instances that we didn't use for the model training, but if we analyze the precision and recall values we can denote some anomalies. The results are in the following table:

Attrition	Precision	Recall
No	0.89	0.95
Yes	0.59	0.40



As we can notice the precision of No Attrition instances is definitely bigger than the precision of Yes Attrition instances, this denote a misleading value of accuracy because our model is able to classify really good the No Attrition Instances but it's not the same with the Yes Attrition Instances, in

which our model is slightly better than a random classifier.
This happened because the skewed distribution of Yes and No values of attrition and the limited training set for Yes Attrition instances.
Decision trees are really popular also because it is possible to see the reason why a classifier gives a certain answer. Let's check the sequence of rules that lead to the purest Yes attrition to try to generalize the concept.


OverTime ≤ 0.5 entropy = 0.638 samples = 1176 value = [986, 190] class = No
True
False
TotalWorkingYears ≤ 2.5 entropy = 0.487 samples = 841 value = [752, 89] class = No
JobLevel ≤ 1.5 entropy = 0.883 samples = 335 value = [234, 101] class = No
JobRole_Research Scientist ≤ 0.5 entropy = 0.913 samples = 67 value = [45, 22] class = No
JobSatisfaction ≤ 1.5 entropy = 0.425 samples = 774 value = [707, 67] class = No
MaritalStatus_Single ≤ 0.5 entropy = 1.0 samples = 125 value = [61, 64] class = Yes
JobRole_Sales Executive ≤ 0.5 entropy = 0.672 samples = 210 value = [173, 37] class = No
HourlyRate ≤ 58.5 entropy = 0.997 samples = 47 value = [25, 22] class = No
entropy = 0.0 samples = 20 value = [20, 0] class = No
StockOptionLevel ≤ 0.5 entropy = 0.66 samples = 158 value = [131, 27] class = No
EnvironmentSatisfaction ≤ 1.5 entropy = 0.347 samples = 616 value = [576, 40] class = No
YearsAtCompany_medium ≤ 0.5 entropy = 0.966 samples = 79 value = [48, 31] class = No
TotalWorkingYears ≤ 4.5 entropy = 0.859 samples = 46 value = [13, 33] class = Yes
JobSatisfaction ≤ 3.5 entropy = 0.365 samples = 129 value = [120, 9] class = No
MaritalStatus_Single ≤ 0.5 entropy = 0.93 samples = 81 value = [53, 28] class = No
entropy = 0.959 samples = 21 value = [8, 13] class = Yes
entropy = 0.931 samples = 26 value = [17, 9] class = No
RelationshipSatisfaction ≤ 3.5 entropy = 0.89 samples = 65 value = [45, 20] class = No
HourlyRate ≤ 84.5 entropy = 0.385 samples = 93 value = [86, 7] class = No
Department_Sales ≤ 0.5 entropy = 0.538 samples = 138 value = [121, 17] class = No
EducationField_Technical Degree ≤ 0.5 entropy = 0.278 samples = 478 value = [455, 23] class = No
DailyRate ≤ 971.5 entropy = 0.884 samples = 53 value = [37, 16] class = No
entropy = 0.983 samples = 26 value = [11, 15] class = Yes
entropy = 0.559 samples = 23 value = [3, 20] class = Yes
entropy = 0.988 samples = 23 value = [10, 13] class = Yes
EnvironmentSatisfaction ≤ 2.5 entropy = 0.499 samples = 82 value = [73, 9] class = No
entropy = 0.0 samples = 47 value = [47, 0] class = No
DailyRate ≤ 766.5 entropy = 0.722 samples = 55 value = [44, 11] class = No
entropy = 0.931 samples = 26 value = [9, 17] class = Yes
NumCompaniesWorked ≤ 1.5 entropy = 0.99 samples = 43 value = [24, 19] class = No
entropy = 0.267 samples = 22 value = [21, 1] class = No
MonthlyRate ≤ 10571.5 entropy = 0.122 samples = 60 value = [59, 1] class = No
entropy = 0.684 samples = 33 value = [27, 6] class = No
TotalWorkingYears ≤ 17.5 entropy = 0.377 samples = 96 value = [89, 7] class = No
HourlyRate ≤ 65.5 entropy = 0.792 samples = 42 value = [32, 10] class = No
DailyRate ≤ 904.5 entropy = 0.219 samples = 428 value = [413, 15] class = No
Age ≤ 37.5 entropy = 0.634 samples = 50 value = [42, 8] class = No
entropy = 0.987 samples = 30 value = [17, 13] class = No
entropy = 0.559 samples = 23 value = [20, 3] class = No
entropy = 0.764 samples = 27 value = [21, 6] class = No
Education ≤ 3.5 entropy = 0.305 samples = 55 value = [52, 3] class = No
entropy = 0.918 samples = 27 value = [18, 9] class = No
entropy = 0.371 samples = 28 value = [26, 2] class = No
entropy = 0.722 samples = 20 value = [16, 4] class = No
entropy = 0.932 samples = 23 value = [8, 15] class = Yes
entropy = 0.286 samples = 20 value = [19, 1] class = No
entropy = 0.0 samples = 40 value = [40, 0] class = No
Age ≤ 38.5 entropy = 0.469 samples = 70 value = [63, 7] class = No
entropy = 0.0 samples = 26 value = [26, 0] class = No
entropy = 0.934 samples = 20 value = [13, 7] class = No
entropy = 0.575 samples = 22 value = [19, 3] class = No
WorkLifeBalance ≤ 2.5 entropy = 0.297 samples = 247 value = [234, 13] class = No
JobRole_Sales Executive ≤ 0.5 entropy = 0.088 samples = 181 value = [179, 2] class = No
entropy = 0.855 samples = 25 value = [18, 7] class = No
entropy = 0.242 samples = 25 value = [24, 1] class = No
entropy = 0.0 samples = 34 value = [34, 0] class = No
entropy = 0.592 samples = 21 value = [18, 3] class = No
entropy = 0.267 samples = 44 value = [42, 2] class = No
entropy = 0.706 samples = 26 value = [21, 5] class = No
entropy = 0.493 samples = 65 value = [58, 7] class = No
entropy = 0.209 samples = 182 value = [176, 6] class = No
entropy = 0.0 samples = 140 value = [140, 0] class = No
entropy = 0.281 samples = 41 value = [39, 2] class = No

OverTime > 0.5 , JobLevel <=1.5, MaritalStatus_Single <=0.5, TotalWorkingYears <=4.5.

We can say that extra working time, a lower job level of responsibility and a marital status of married or divorced leads to attrition to work.

For the 3 and 4 classification trees we decided to change only the max_depth parameter, from a low value to a no limitation one. The tree with a low value of max depth showed better results on the test set rather than the one with no limitation, this shows that with a tree with lower depth generalized the Attrition concept better than a higher number of leafs due to the overfitting.

Random Forest

We tried to build another model in an attempt to get a better result, for this purpose we decided to build a random forest. We used the same division of training and test with a random generator method to find the best set of parameters. We chose the following parameters: {'min_samples_split': 5, 'min_samples_leaf': 1, 'max_depth': 30, 'criterion': 'entropy'}. We obtained an accuracy of 85%, slightly lower than the previous better decision tree. However if we observe the precision and recall value we can see that our classifier tend to answer no at the most of our instances, as we can see from the following table:

Attrition	Precision	Recall
No	0.86	0.99

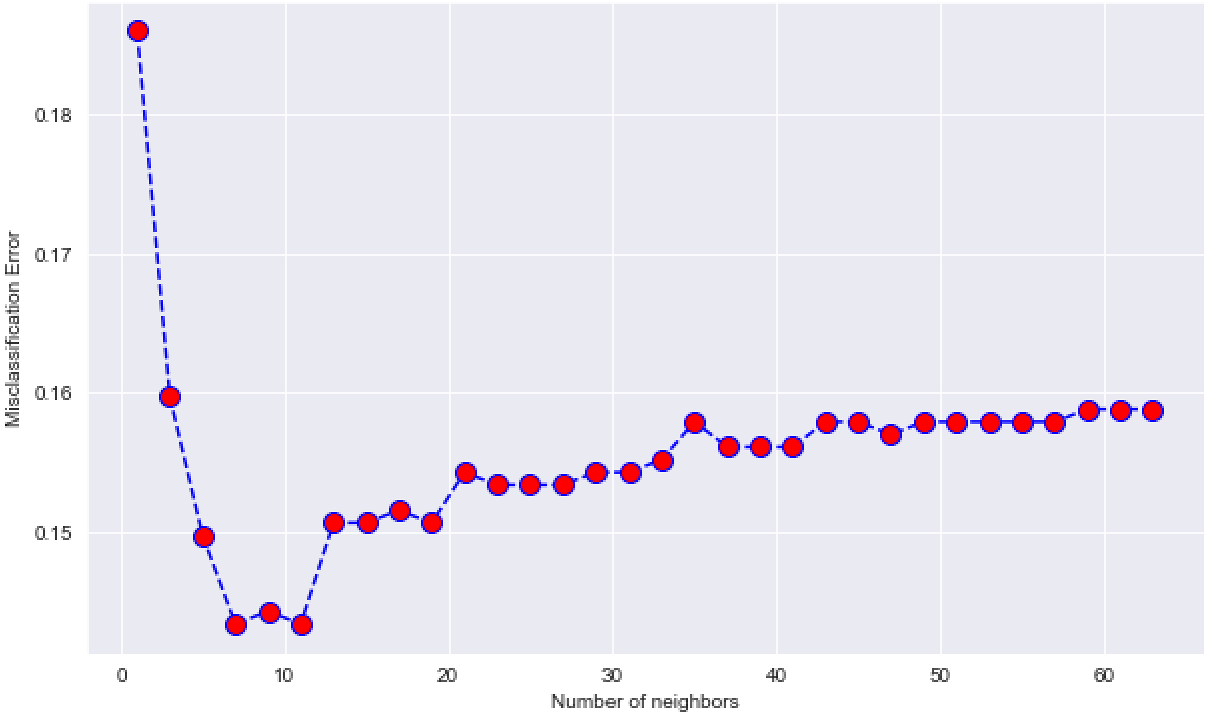
Yes	0.75	0.13

Yes attrition precision is definitely higher than our previous tree but as we can see its recall it’s too low to be considered reliable. This is the result of the voting majority system of random forest algorithm. More trees are involved in this classification model consequently the probability that more trees agree with the No class label is higher as the more precision obtained for the Yes class label.

Confusion matrix		True class	
		Positive	Negative
Predicted class	Positive	245	2
	Negative	41	6

KNN CLASSIFICATION

As the last classification method we applied to our dataset an instance-based learning, Nearest Neighbor classifiers.
We used the same division of the dataset that we use for the previous classification models, but this time we used it just to check the validity of our results since there is not a model building.
We applied this method with the traditional euclidean distance and we used the entire set of our variables. Before applying the algorithm, we have splitted all the categorical values in binary variables (e.g. BusinessTravel is splitted into BusinessTravel_Non-Travel, BusinessTravel_Travel_Frequently, BusinessTravel_Travel_Rarely) in doing so make it sense to include them all to this process. Furthermore Nearest-neighbor classifiers can produce wrong predictions unless variables have really different variability, for this reason we normalized our dataset.
We iterated our knn algorithm with different values of neighbors as we can see from the plot below:



This plot associates a misclassification error value to each neighbor number and helps us find the optimal number of neighbors. After a certain number (around 50 number of neighbors) we can see that the curve tends to stabilize, this is the case when our neighborhood is so big that the skewed distribution of No Attrition has the best. We resumed our most interesting values in the following table:

Number of neighbors	Precision	Recall	F1-Score	Average Accuracy
3	Test Result No Attr: 0.85 Yes Attr: 0.52	Test Result No Attr: 0.97 Yes Attr: 0.18	Test Result No Attr: 0.91 Yes Attr: 0.27	TestSet: 0.8342
7	Test Result No Attr: 0.85 Yes Attr: 0.58	Test Result No Attr: 0.98 Yes Attr: 0.11	Test Result No Attr: 0.91 Yes Attr: 0.19	TestSet: 0.8370
11	Test Result No Attr: 0.84 Yes Attr: 0.71	Test Result No Attr: 0.99 Yes Attr: 0.08	Test Result No Attr: 0.91 Yes Attr: 0.14	TestSet: 0.8397

13	Test Result No Attr: 0.84 Yes Attr: 0.80	Test Result No Attr: 1.00 Yes Attr: 0.06	Test Result No Attr: 0.91 Yes Attr: 0.12	TestSet: 0.8397

As we can notice from the table, if we set the number of neighbors of 11 or 13 we obtain a high precision value for the Yes Attr but the recall is really bad and since our main aim is to retrieve instances of employees with Yes Attrition we have to discard this option. If we set a value of neighbour lower or equal to 7 the recall for the Yes Attrition instances increase but the precision is really low, really close to a random classifier. Finally we can say that Knn gives us the worst results of classification.

Discussion of the best prediction model

From the discussion of KNN classification, we can conclude that KNN is not a good approach for this dataset. Either we gain in precision and lose in recall, or vice versa. Hence, right off the bat, we would like to eliminate this possibility in comparing the most suitable prediction model.

This leaves us with: Decision Trees and Random Forest. Basically at heart, both of the algorithms are similar since random forest is but a cluster of decision trees whose output is average to reach a consensus. However, as similar as it may seem in concept, the prediction can be significantly different.

The biggest challenge we face with this dataset, setting aside the low number of data points or the high dimensionality, is the skewness of the variable to be predicted: Attrition. With around 85 percent of the rows labeled ‘No’, it is not hard to gain a high confidence model, even though it may just guess ‘No’ regardless of the input. In a random forest model, we see this behaviour reinforced. This is also the reason we can not really look at the accuracy of the models to get a good idea of their capabilities, and hence, we are going to look at their precision and recall values below.

Attrition	Precision		Recall	
	Yes	No	Yes	No
Decision Tree	0.59	0.89	0.40	0.95
Random Forest	0.75	0.86	0.13	0.99

As we can see from the table above, even though Decision Tree has a lower precision value for yes attrition, compared to the Random Forest, it has a greater recall value for yes. Given that, among the values, it is more important for a model to be able to predict the yes values better, we would like to choose the Decision Tree classifier as the better of the suggested models.

ASSOCIATION ANALYSIS

As our last analysis we're going to do an association analysis in order to uncover hidden relationships, in particular relationships that involve the Attrition. In order to apply an association analysis we have to process our dataset in a form commonly known as market basket transaction. For these reasons we divided all the numerical variables in some certain bins according to their distributions and added some labels to the single values for a better understanding of the patterns extraction.

As the first step we extracted frequent patterns. We extracted them using a minimum value of support equal to 60% and with a minimum number of elements equal to 2, because it's useless to extract patterns with only one value. We obtained the following list, composed of 10 couple of elements, followed by the support value:

1. ('No_Attrition', 'Excellent_PerformanceRating'), 0.728
2. ('[0.0, 3.75)_YearsSinceLastPromotionRange', 'Excellent_PerformanceRating'), 0.675
3. ([0.0, 10.0)_YearsAtCompanyRange', 'Excellent_PerformanceRating'), 0.662
4. ('[0.0, 10.0)_YearsAtCompanyRange', 'No_Attrition'), 0.650
5. ('[0.0, 3.75)_YearsSinceLastPromotionRange', 'No_Attrition'), 0.649
6. ('No_OverTime', 'No_Attrition'), 0.642
7. ('Travel_Rarely', 'Excellent_PerformanceRating'), 0.637
8. ('No_OverTime', 'Excellent_PerformanceRating'), 0.622
9. ('Travel_Rarely', 'No_Attrition'), 0.616
10. ([0.0, 10.0)_YearsAtCompanyRange', '[0.0, 3.75)_YearsSinceLastPromotionRange, 0.60

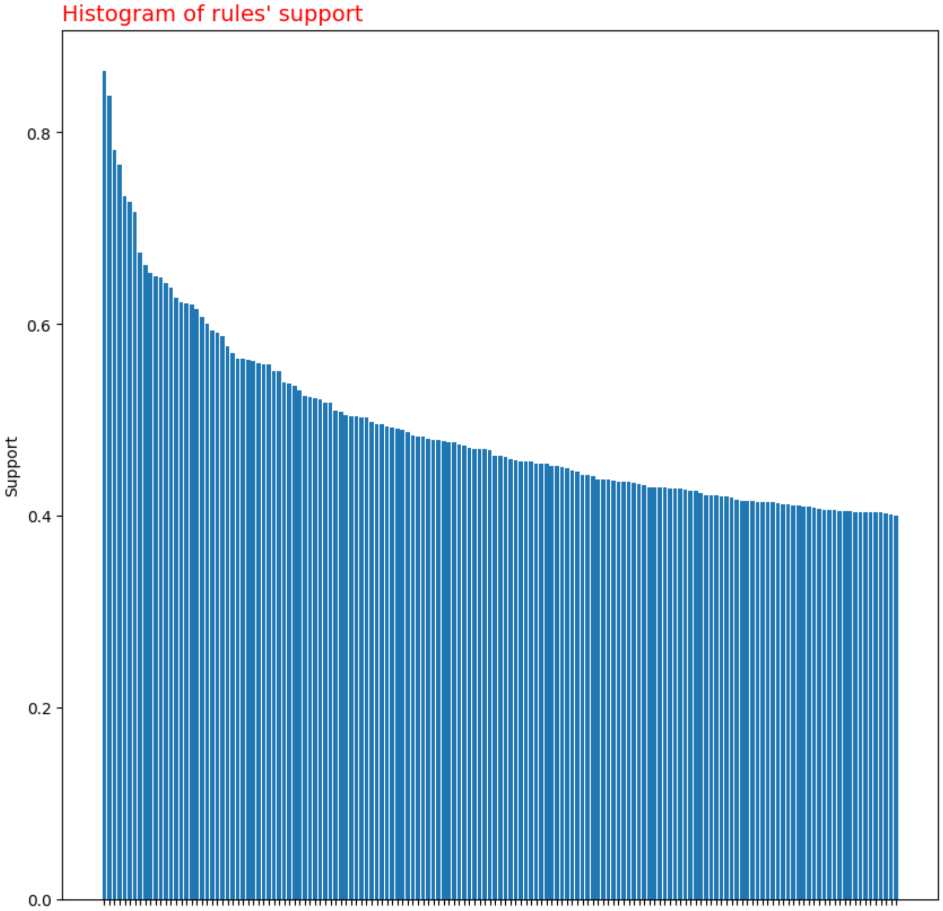
As we can see from the list above we discovered some importants patterns, let's analyze patterns that involve attrition values first. The value No_Attrition, appears in 5 out of 10 patterns. If we join all these sets we obtain that No_Attrition values frequently can be found with: Excellent_PerformanceRating, [0.0, 10.0)_YearsAtCompanyRange, [0.0, 3.75)_YearsSinceLastPromotionRange', No_OverTime, 'Travel_Rarely. If we include in our frequent extraction singular elements, with the same minimum support, we obtain:

1. (**'Excellent_PerformanceRating'**), 0.8639455782312925)
2. (**'No_Attrition'**), 0.839
3. (**[0.0, 3.75)_YearsSinceLastPromotionRange**), 0.782
4. (**'[0.0, 10.0)_YearsAtCompanyRange'**,), 0.766
5. (**'Travel_Rarely'**,), 0.7333333333333333
6. (**'No_OverTime**',), 0.7170068027210884

This list includes all those elements, they are all often and for this reason they appear together. If we want to observe the highest support of a set with at least 3 elements we have to decrease our support of 50%. The rules with 3 frequent elements with the most support is: ('[0.0, 3.75)_YearsSinceLastPromotionRange', 'No_Attrition', 'Excellent_PerformanceRating') with 0.563 of support. Not surprisingly it is composed of the first 3 elements of our list.

We can say that most of the employees have an excellent performance rating, they do not have attrition on the work, they travel rarely, don't do over time and they are in the company less than 10 years. Also interesting to note that most employees have a value of a year since the last promotion range lower than 3.75 years, this means that only a small set of employees wait more than 4 years to obtain a promotion.

If we set the support of 35% (really low) we obtain a set of 276 frequent patterns, if we plot the support level of them we obtain the following result:



If we extract the frequent pattern again with the minimum support equal to 30% we obtain a set of 515 instances, almost the double of the set with minimum support of 35%. As we can see there is a small set with a high level of support and then the curve tends to stabilize itself as more elements have the same value of support.

Let's now extract the main association rules, as on the first attempt we fixed the minimum confidence level at 80% and support level to 40%. We obtained 163 association rules. The rule with the highest confidence is:
('[0.0, 4.5)_YearsInCurrentRole', '[0.0, 4.25)_YearsWithCurrManager') ->([0.0, 3.75)_YearsSinceLastPromotion)
This rule has a support of 0.55, a confidence level of 0.98, and a lift of 1.25. The value of confidence is really high and the Interest Factor (also known as lift) states a positive relationship between the antecedents set and the consequent. However this rule is quite trivial because bind employees with the lower level of YearsInCurrentRole and YearsWithCurrManager to the lower values of YearsSinceLastPromotion. In other words employees that are in a current role and know a certain manager, responsible for promotions, since a few years; usually wait at least 4 years to obtain a promotion.

Since the first rules are trivial as the one above we're going to write just the more interesting ones, followed by values of confidence and lift value:

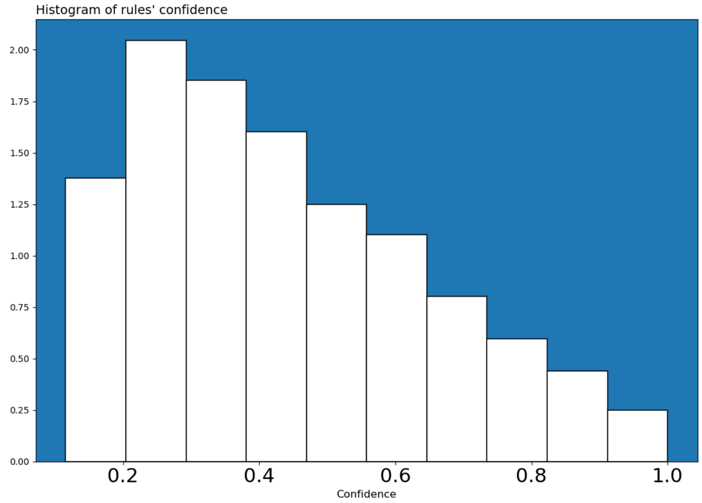
- (No_OverTime) -> (No_Attrition), 0.89, 1.07
- (Married) -> (No_Attrition), 0.87, 1.04
- (Research & Development_Department) -> (Excellent_PerformanceRating),0.87,1.00
- (Travel_Rarely) -> (Excellent_PerformanceRating),0.87,1.00
- (No_Attrition) -> (Excellent_PerformanceRating),0.86,1.00
- (No_OverTime) -> (Excellent_PerformanceRating),0.86,1.00
- ([1.0, 6.6)_DistanceFromHomeRange) -> (No_Attrition),0.86,1.03
- (Male) -> (Excellent_PerformanceRating),0.86,0.99826
- (Research & Development_Department) -> (No_Attrition), 0.86,1.02
- (Better_WorkLifeBalance) -> (No_Attrition),0.85,1.02
- (Male) -> (No_Attrition), 0.83,0.999

As we can see the most interesting rules involve "No_Attrition" and "Excellent_PerfomanceRating", as we expected from the frequent pattern analysis. We can observe that employees that do not have attrition at work might have these characteristics: don't do an over time job, married, closer from home, have a better work life balance or work in the Research & Development_Department. An Excellent performance rating (a value of 4 over 5) is associated with these characteristics: work in the Research & Development_Department, travel rarely, don't do over time job, male and they do not have attrition at work.

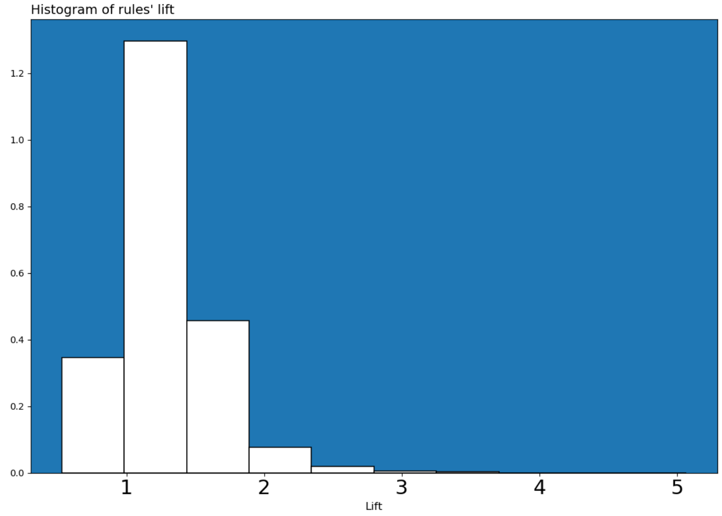
Since the frequency of Yes Attrition is low we couldn't find any rule that involves our main target, for this reason we decided to filter our list and find the higher confidence rule for consequents as Yes Attrition. We obtained the following rule: (0_StockOptionLevel) -> (Yes_Attrition)
support =0.10, confidence = 0.24, lift = 1.51

Support and confidence values are really affected by the skewed distribution of Yes and No Attrition, the high value of lift measures underline the fact that they are positively related.

For a further analysis we have plotted all the frequency of rules sorted by confidence value and lift measure value in the following plots:



We can see that only a few association rules have a high value of confidence. The peak of frequency is reached around a confidence value of 30% but then there are less rules with confidence less than 20%.



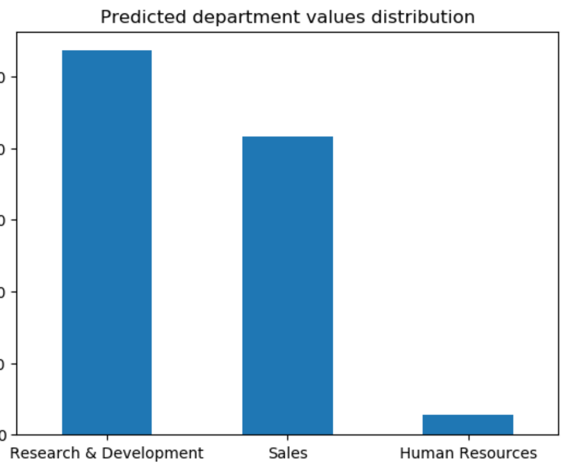
From the histograms of rules' lift we can see that more association rules in our dataset have a lift value greater than 1, this denotes a positive correlation between antecedent and decedent. The association rules with the highest value of lift measure is: (Sales_Department) -> (Sales Executive) with a lift value of 3.30. This rule is really trivial because it denotes a really high relationship between employees that do as job role Sales Executive and its Department.

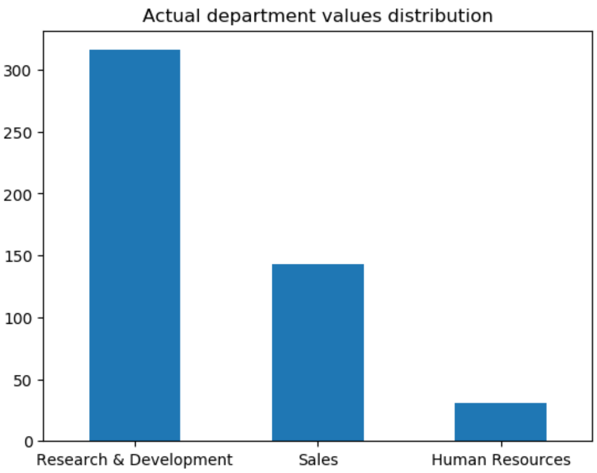
As a final step we will try to use the most meaningful rules to replace missing values and to predict the target variable and evaluate the accuracy.

We decided to apply the missing values replacing using meaningful association rules to the variable Department, as we have noticed that it has a relationship with the JobRole variable. On this purpose we removed ⅓ of the values in this row, so now we have 490 missing values over 1470. We are going to build a simple way to replace missing value using those rules:

- For “Research & Development” we are going to use:
    - (Research Scientist) -> (Research & Development_Department) with conf= 1.00 supp=0.20
    - (Laboratory Technician) -> (Research & Development_Department) with conf=1.00 supp=0.18
    - (1_JobLevelRange) -> (Research & Development_Department) with conf= 0.80 supp=0.30;
    - (Medical_EducationField) -> (Research & Development_Department) with conf = 0.78 supp=0.25;
    -
- For “Sales” we are going to use:
    - (Sales Executive) -> (Sales_Department) with conf=1 supp=0.22;
    - (Marketing_EducationField) -> (Sales_Department) with supp=0.11 conf=1.00
    - (2_JobLevelRange) -> (Sales) with supp= 0.16 conf=0.44;
    - ([0.0, 10.0)_YearsAtCompanyRange) -> (Sales_Department) with supp= 0.24 conf =0.32;
- If a certain distance doesn’t match any of the previous rule we would target it as “Human Resources”

Rules will be applied in order of confidence, if a certain instance has a particular value of antecedent we will use the consequent as a label of missing value.
To improve the evaluation of the accuracy in our algorithm, let’s plot the distribution those 490 row (⅓ of the entire dataset) of actual data versus predicted data:


Predicted department values distribution


Actual department values distribution

As we can see the main difference is given by the sales instances. It predicted 352 right instances in 490 total instances, so it has an accuracy of 71.8% .

Finally we are going to create a classifier using the most meaningful rules to predict the target variable. Association rules will be retrieved in order of confidence and support and our target will be to use the ones with the highest value of confidence and support.

- (No_Overtime) ->(No_Attrition) with supp=0.64 ,conf= 0.89
- (No_OverTime, Excellent_PerformanceRating) ->(No_Attrition) with supp=0.55, conf=0.89
- (Research & Development_Department) -> (No_Attrition) with supp= 0.56, conf=0.86
- (Better_WorkLifeBalance) -> (No_Attrition) with supp=0.52, conf=0.85

- (0_StockOptionLevel)->(Yes_Attrition) with support 0.10 , conf=0.244
- ([0.0, 3.75)_YearsSinceLastPromotionRange), ([0.0,10.0)_TotalWorkingYearsRange) -> Yes_Attrition with supp = 0.10, conf=0.218
- ([0.0, 10.0)_TotalWorkingYearsRange) -> Yes_Attrition with supp= 0.10 , conf=0.217
- ([0.0, 3.75)_YearsSinceLastPromotionRange) -> Yes_Attrition with supp= 0.10, conf=0.208

We can notice that the extraction process for the Yes_Attrition target variable only generated rules with a low confidence value(the rule hasn't been found often) and a low support value(the itemset doesn't appear often in the dataset). That is reasonable because of the Attrition variable distribution. We'll choose the relationship (No_OverTime, Excellent_PerformanceRating) ->(No_Attrition) with low confidence and support value but the highest ones at the same time.

Differently from the yes_Attrition values, the No_Attrition variable has higher values. The relationship (No_OverTime, Excellent_PerformanceRating) ->(No_Attrition) can be chosen as the most meaningful rule for the No_Attrition because both the itemsets forming the rule and the rule itself appear more often than the others.

The classifier managed to classify 1220 items correctly in 1470 instances, with an accuracy of 83%.