Human Resources Analysis

A Data-Driven approach to forecasting the premature leaving of employees

Atzeni Daniele, Abhinav Dwivedi,
Ritacco Antonio, Sbai Saad Eddine

DATA MINING PROJECT, UNIVERSITÀ DI PISA

A.Y. 2017/2018

Contents

1. Data Understanding

1.1 Introduction

The dataset is provided by a company that want to predict how many employees will leave.
It's composed of some indicators of the human resources department. There are 14999 records and 10 attributes :

1. 'Sales': categorical nominal attribute, it represents the department of each employee. We can see in figure 1.1 all the possible values of this variable and their percentages relative to the dataset.

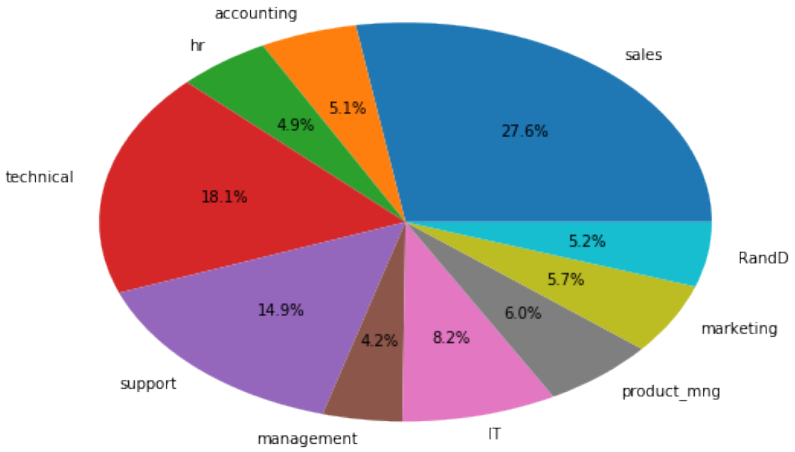

Figure 1.1: Pie chart of attribute 'sales'

2. 'Salary': categorical ordinal attribute, it represents the salary of each employee. Possible values are 'High' (8%), 'Medium' (43%) and 'Low' (49%)
3. 'Work Accident': numerical boolean attribute, its values are 1 (14%) if the employee has had an accident in last 2 years, 0 (86%) otherwise.
4. 'Left': numerical boolean attribute, its values are 1 (24%) if the employee has left the company, 0 (76%) otherwise.
5. 'Promotion Last 5 Years': numerical boolean attribute, its values are 1 (2%) if the employee has had a promotion in the last 5 years, 0 (98%) otherwise.

6. 'Time Spend Company': numerical discrete attribute, it represents the number of years of experience within the company.
7. 'Number of Project': numerical discrete attribute, it represents in how many project employees are involved.
8. 'Satisfaction Level': numerical continuos attribute between 0 and 1. A low value means that the employee is not satisfied.
9. 'Last Evaluation': numerical continous attribute between 0 and 1. A low value means that the employee is not seen as a good worker.
10. 'Average Montly Hours': numerical discrete attribute.

1.2 Data Understanding

In the table 1.1 we can see some statistics about numerical attributes.

Table 1.1: Statistics

	satisfaction_level	last_evaluation	number_project	A.M.H	T.S.C
count	14999	14999	14999	14999	14999
mean	0,612834	0,716102	3,803054	201,0503	3,498233
std	0,248631	0,171169	1,232592	49,9431	1,460136
min	0,09	0,36	2	96	2
25%	0,44	0,56	3	156	3
50%	0,64	0,72	4	200	3
75%	0,82	0,87	5	245	4
max	1	1	7	310	10

We computed the correlation matrix using Pearson correlations (we got similar results with Spearman and Kendall correlations) and we draw a correlation matrix heatmap (figure 1.2). We can see that the biggest values are A.M.H./n.project (0.4172), n.project/last evaluation (0.3493), A.M.H./last evaluation (0.3397). Anyway, because correlations are not very big, we can't elimate or merge any attributes.

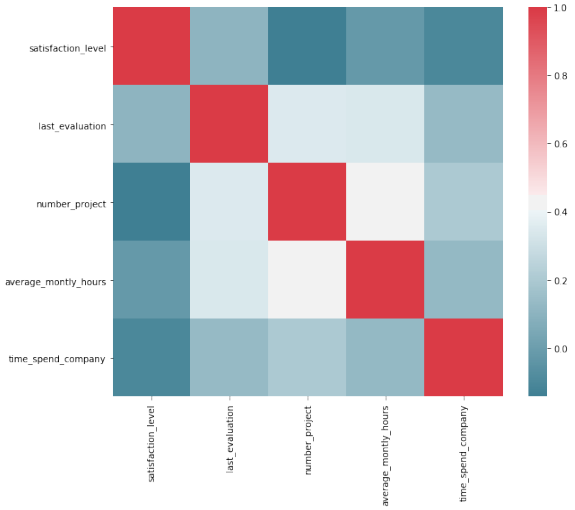

Figure 1.2: Correlation matrix heatmap

As regards data quality we didn't find any missing values and syntactic errors. In fact we looked for that with value_counts method, in this way if there were some typing errors it would have printed strange values, and isnull method.
We checked outliers with boxplot and the John Tukey Empirical Formula (Lower Limit: Q1-1.5IQR , Upper Limit: Q3+1.5IQR). We didn't find any of that concerning the variables 'Satisfaction Level' and 'Last Evaluation' (figure 1.3, while we found some outliers concerning the variable 'Time Spend Company' (figure 1.4. Anyway we decided not to delete this records because we think that this attribute does not have a gaussian distribution, so the boxplot is useless.

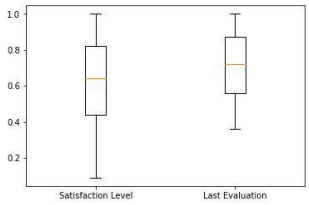

Figure 1.3: Outliers identifications through Boxplot

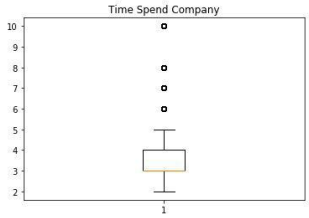

Figure 1.4: Outliers identifications through Boxplot

When we looked at the distribution of numerical attriutes, we didn't find any interesting pattern (figure 1.5); maybe the only variable that has a similar distribution of a known one is 'Average Montly Hours', that seems a bimodal.

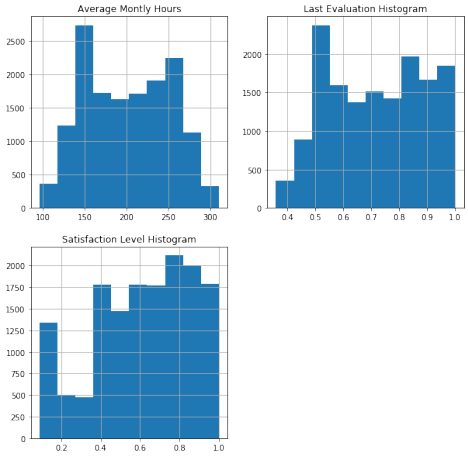

Figure 1.5: Distributions of numerical attributes

On the contrary we found some interesting pattern within the variables 'Average Montly Hours', 'Last Evaluation' and 'Satisfaction Level' when we isolate the people that have left the company, so we decided to plot these variables in a 3D scatter plot and we crearly found 3 different clusters. We can see our result in figure 1.6, but we will discuss it in the newxt chapter.

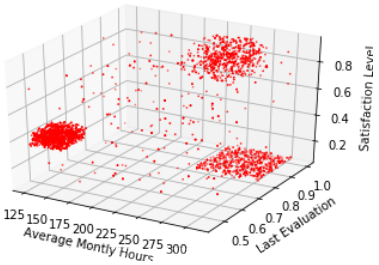

Figure 1.6: 3D scatter plot of people who left the company

Finally we created some crosstab between the variable 'Left' and other variables. We saw that people that has stayed in the company for more than 6 years didn't leave the company (figure 1.7) and that is much less likely that people with high salary or people that has had a promotion leave the company (figure 1.8).

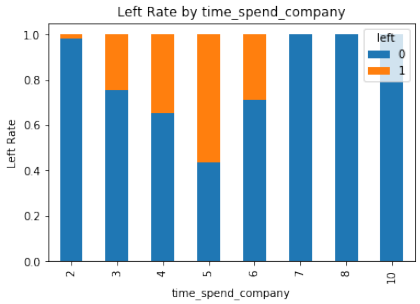

Figure 1.7: Left rate vs Time Spend Company

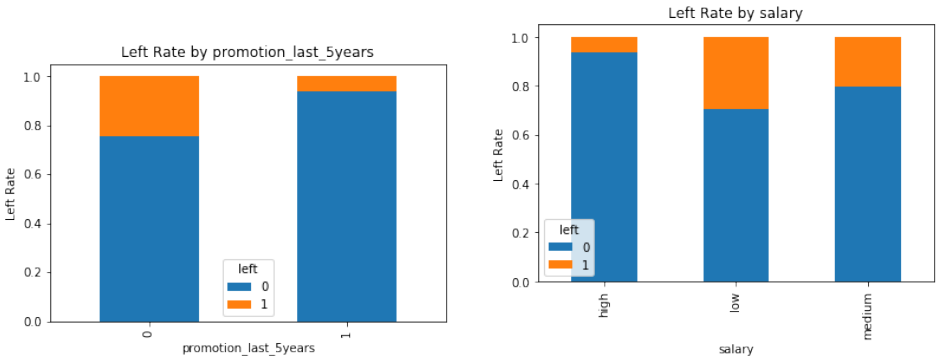

Figure 1.8: Left rate vs Salary

2. Clustering Analysis

2.1 K-Means on people who left

Initially, helped by figure 2.1, we decided to analyze the dataset composed by people who left. So we rescaled the variables 'Average Montly Hours', 'Time Spend Company' and 'Number of Project' to make sure they were within interval [0, 1], we used the K-Means algorithm with different values of K using the three previous variables, 'Last Evaluation' and'Satisfaction Level' and euclidean metric, then we plotted our results. As suggested by the scatter plot, we found that the ideal value of K is 3. In fact we can clearly see in figure 2.1 an elbow for that value of K.

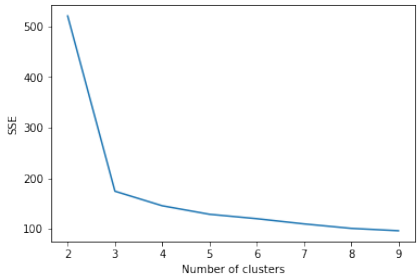

Figure 2.1: SSE as function of K, people who left

With K= 3 we also get a high silhouette, 0.733, and a low SSE, 174. Then we compute the centroids and we plotted again the results in figure 2.2.

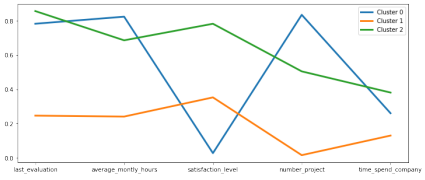

Figure 2.2: Clusters' centroids of people who left

We can do the following observations:

- The biggest cluster is the cluster 1, composed by 1620 records, that is the 45% of the people who left, and formed by 'hard workers', with high number of project, average montly hours

and evaluation, but with a very low satisfaction level. Maybe the company should have had to increase their satisfaction levels with a promotion or an increase in their salary.

- The second biggest cluster is cluster 0, with 993 records (28%), characterized by high satisfaction level and evaluation. It seems that the only plausible reason of their firing is that they were looking for more prestigious works.
- Cluster 2 instead is composed by 'lazy workers', characterized by low evaluation, satisfaction level, average montly hours and number of project. This cluster counts 958 records (27%).

When we compare these cluster with the original dataset of people who left we didn't find out a lot of differences, with the exception of promotions in the last year. In this case we noticed that, despite being very few, the 79% of promoted people are in cluster 1. Maybe these promotions were a failed attempt to stimulate these 'lazy workers'. After that we tried to understand the ideal value of K for the whole dataset, using the same procedure we used in the previous case. Unfortunatly in this case the plot is not very clear, figure 2.3, so we decided to ask for help to hierarchical clustering.

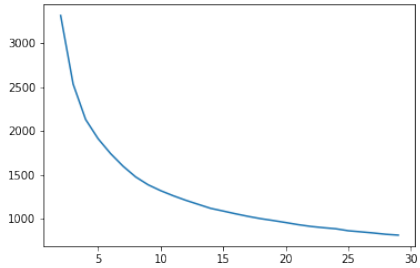

Figure 2.3: SSE as function of K

2.2 Hierarchical Clustering

To begin the clustering we did the same thing we did before, i.e. rescaling variables and using euclidean metric. We first tried with single-linkage clustering, but we didn't get a clear result. So we tried with complete-linkage clustering and we got a little better result than before, whith 11 suggested clusters (figure **??**. But we got the best result with the Ward's method, as we can see in the dendogram in figure 2.5.

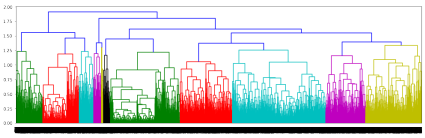

Figure 2.4: Dendogram with complete-linkage clustering

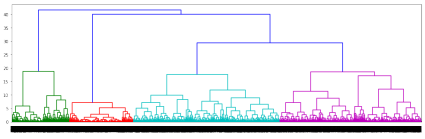

Figure 2.5: Dendogram with Ward's method

We cut the dendogram in order to get 4 clusters, as suggested by different colors, and we got a silhouette of 0.231, that is not very high. Then we calculated the centroids of the clusters, computing the means of the variables we used for this clustering algorithm, to be able to describe better the clusters.

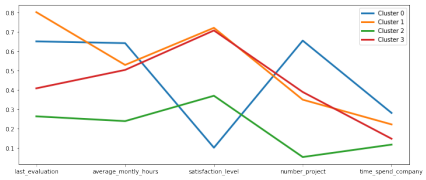

Figure 2.6: Centroids we obtained with hierarchical clustering

We can notice that:

- The centroid of cluster 0 is very similar to the centroid of the cluster 0 we got with K-means algorithm on people who left, characterized by high evaluation and montly hours, but with very low satisfaction level. This cluster is formed by 2113 workers, with a left rate of 47%, much higher than the 24% of the whole dataset.
- The centroid of cluster 2 is very similar to the centroid of on of the previous clusters, precisely prevoius cluster 1. In this cluster there are 2345 records and, as before, there is a very much higher left rate than the whole dataset, i.e. 64%.
- The centroids of cluster 1 and cluster 3 are pretty similar, in fact they both have high satisfaction level and medium montly hours and number of project. The only difference is in last evaluation: for the centroid of cluster 1 is high (~ 0.8), for centroid of cluster 3 is low ($\sim 0,4$). Despite this difference in both clusters the left rate is lower than the rate of the whole dataset, 18% in cluster 1 and just 2% in cluster 3. These two clusters are also the bigger, 5184 in cluster 1 and 5357 in cluster 3. Also in view of all these observations we decided to do the K-Means on the whole dataset with 3 clusters, hoping that the algorithm will merge cluster 1 and cluster 3, preserving cluster 0 and cluster 2.

Anyway we can conclude that, despite a low silhouette, this clustering creates clusters much purer than the whole dataset about the variable 'left'.

2.3 K-Means on the whole dataset

As discussed in the previous section, we decided to use K-Means algorithm with K$=3$. We got a silhouette score equals to 0.283 and SSE equals to 2533. We can see in figure 2.7 we obtained the results we hoped.
In fact we can see that:

- Cluster 0 and cluster 2 very similar to previous cluster 2 and cluster 0, with similar centroids (low satisfaction, evaluation and montly hours in cluster 0, high evaluation and montly hours, low satisfaction in cluster 2) and left rates higher than the whole dataset's rate, 32% and 51% respectively.
- Cluster 1 that is the merger of the previous cluster 1 and cluster 3, with centroid characterized by high satisfaction level, medium montly hours and evaluation near 0.7 (number between 0.4 and 0.8, i.e. previous evaluations) and a left rate much smaller tha the one of the whole dataset, only the 12%.

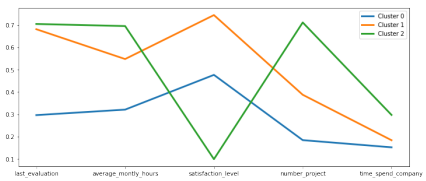

Figure 2.7: Centroids we obtained with K-Mens, K=3

2.4 DBSCAN Clustering

In order to do DBSCAN clustering, we first look at possible values of the parameters min pts and eps. We did the same work on numeric variables we had done in the previous clustering analysis. Then we started fixing min pts= 4 and computing the silhouette score for different values of eps, from 0.05 to 0.4 with a step equal to 0.05, and we repeted this procedure for different values of min pts. In figure 2.8 we can see the results we obtained.

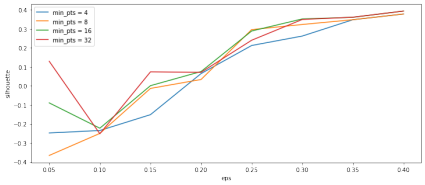

Figure 2.8: Silhouette score in function of min pts and eps

As we can see we don't get high values for silhouette score in any case. We tried DBSCAN algorithm with some different parameters and we saw that we obtained two different cases:

- With high values of eps (~ 0.25) we have obtained two or three very unbalanced clusters, one of them composed by 96-98% of the dataset.
- With lower values of eps (~ 0.2) we have obtained a lot of clusters (in some cases even 27 clusters), most of which are made up of a few elements.

We can conclude that DBSCAN clustering is not very useful in this dataset, even if it can be used to eliminate some outliers, and that the best clustering algorithms are K-Means with K=3 and hierarchical clustering with Ward's method.

3. Association Analysis

3.1 Frequent,Close Maximal Itemsets

In this part of project we are going to consider the frequent ,close maximal itemset and association rules.
The algorithm generates frequent itemsets based on the anti-monotonic properties of the support, which ensures that support for an itemset can never exceed the support of one of its subset. In this way, the algorithm can drastically reduce computational costs by not calculating the supersets of items that are infrequent.
Once the procedure for generating frequent items has been completed, associative rules are generated for each of them.
However, the rule must satisfy a minimum threshold of confidence, i. e. the measure of how often an x attribute appears in transactions where another y attribute appears, also preset by the user. Moreover, since in some cases above all the confidence measurement can be misleading for the purpose of analysing the correlation between two or more attributes, it has been decided to take into account also another type of measure of interest, namely the Lift, calculated as the confidence of a rule on the support of the consequence of the rule itself, and which assumes a value lower than 1 in the case of negative correlation, equal to 1 in the case of statistical independence, and greater than 1 in the case of positive correlation.
Above all we have to transform the dataset in an transiction table. The continuos variable are discretized :

- “Satisfaction Level” and “Last Evaluation” is distributed in an interval between 0.0 to 1.0 with 0.1 interval .
- “Average Montly Hours” is distributed between 90 and 310 with a 10 interval.

Table 3.1: Conversion of attributes

Attributes	**Value in Transiction Table**
Number of Project	value + _NoP
Time Spend Company	value + _TSC
Work Accident	NoAcc or YesAcc
Left	NoAcc or YesLeft
Promotion	NoProm or YesProm
Department	no change
Salary	no change
Satisfaction Level	value + _Sat
Average Hours	value + _AH
Last Evaluation	value + _LE

The most interesting pattern we have found are some frequent itemset with a minsup = 20 %
'low', 'NoLeft', 'NoAcc', 'NoProm' Support : 27.7485
'medium', 'NoLeft', 'NoAcc', 'NoProm' Support : 27.5485
'3TSC', 'NoLeft', 'NoAcc', 'NoProm' Support : 26.3084
Instead , in the maximal frequent itemset we found this pattern : 'NoLeft', 'NoAcc', 'NoProm' Support : 61.3374

3.2 Association Rules

For the association rules we saw that we obtained a lot of rules even with high value of min confident. So we decided to use a priori algorithm with parameteres values of minConf = 90 and minSup= 5 and we imposed that the consequent must be a possible value for the variable 'left'. We obtained a lot of rules and we selected the following:

'0.4Sat', '2NoP', '524AH' –> YesLeft lift 409.4 conf 97.471 sup 6.32709
'0.4Sat', '2NoP', '0.5LE', 'NoAcc' –> YesLeft lift 410.857 conf 97.8178 sup 7.02714
'0.8LE', '3TSC', 'NoAcc' –> NoLeft lift 129.395 conf 98.5879 sup 5.19368
'0.7Sat', '3TSC' –> NoLeft lift 129.441 conf 98.6233 sup 5.32702
'0.4Sat', '2NoP', '0.5LE', '3TSC', 'NoProm' –> YesLeft lift 414.613 conf 98.7121 sup 7.24715
'0.6Sat', '3NoP' –> NoLeft lift 129.611 conf 98.7531 sup 5.34702
'0.8Sat', '3TSC' –> NoLeft lift 129.775 conf 98.8776 sup 6.53377
'0.8Sat', '3TSC' –> NoLeft lift 129.73 conf 98.8433 sup 6.34042
'0.8Sat', '3NoP' –> NoLeft lift 129.685 conf 98.8095 sup 5.04034
'0.6Sat', '3TSC', 'NoAcc' –> NoLeft lift 129.859 conf 98.9418 sup 5.04034
'0.8Sat', '3TSC', 'NoAcc', 'NoProm' –> NoLeft lift 130.065 conf 99.0991 sup 5.18035
'0.6Sat', '3TSC' –> NoLeft lift 130.133 conf 99.1507 sup 6.28042
'0.4Sat', '2NoP', '524AH', '3TSC', 'NoAcc', 'NoProm' –> YesLeft lift 416.651 conf 99.1972 sup 5.81372
'1.0Sat' –> NoLeft lift 131.248 conf 100.0 sup 6.84046

We noticed that there are a lot of rules with consequent 'NoLeft' has in the antecedent a satisfaction level greater than 0.5 and with consequent 'YesLeft' and in the antecedent a satisfaction level lower than 0.5.
So we decided to do a new discretization of satisfaction level in two groups: lowsat if the sat_lev$\leq$0.5, highsat otherwise. We think that the most interesting rules are the following:

’3NoP’, ’3TSC’, ’highsat’, ’NoAcc’ –> NoLeft lift 131.05 conf 99.8489 sup 8.82726
’0.6LE’, ’3TSC’, ’highsat’, ’NoProm’ –> NoLeft lift 130.672 conf 99.5609 sup 6.07374
’technical’, ’3TSC’, ’highsat’ –> NoLeft lift 130.593 conf 99.5012 sup 5.34702
’3TSC’, ’highsat’ –> NoLeft lift 130.181 conf 99.187 sup 28.7019
’2TSC’, ’3NoP’, ’highsat’, ’NoAcc’ –> NoLeft lift 129.984 conf 99.0374 sup 6.23375
’0.5LE’, ’524AH’, ’lowsat’ –> YesLeft lift 392.311 conf 93.4023 sup 5.15368
’2NoP’, ’0.5LE’, ’lowsat’, ’3TSC’, ’NoProm’ –> YesLeft lift 410.468 conf 97.7253 sup 7.62051
’2NoP’, ’lowsat’ –> YesLeft lift 358.55 conf 85.3645 sup 11.9808
’2NoP’, ’524AH’, ’lowsat’ –> YesLeft lift 398.346 conf 94.8393 sup 6.84712
’0.5LE’, ’lowsat’, ’low’ –> YesLeft lift 347.519 conf 82.7381 sup 5.60037.

4. Classification

4.1 Decision Trees

The dataset is shuffled and divided in Training and Test Set (80% and 20% respectively) and each model is fitted on training set and evaluated on test set. The hyperparameters needed to be tuned were the minimum number of leafs in each node and the maximum depth of the tree. Each not numerical attribute are then trasformed in categorical values. The chart in figure 4.1 represents on

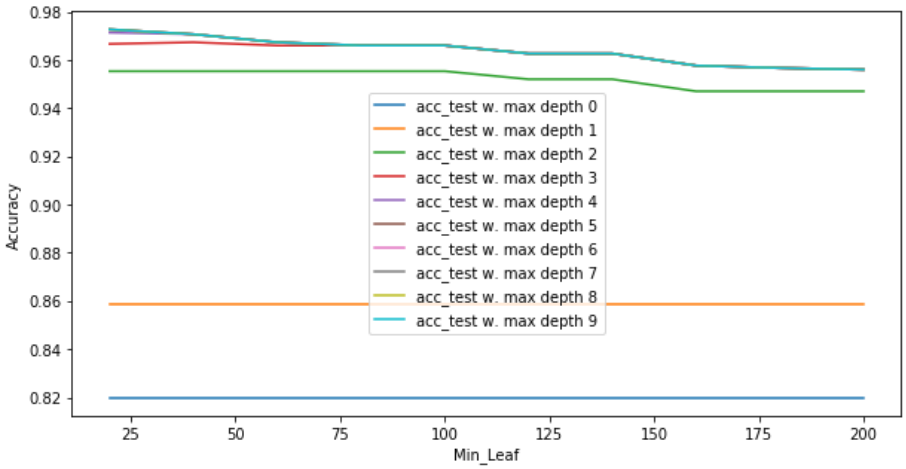

Figure 4.1: Accuracy on MinLeaf and MaxDepth

the X axis the minimum number of leaf to be present in each node and on the Y axis the accuracy. We tested different model with increasing max depth and the best are the ones with max depth ≥ 3 and with min leaf < 25. The Accuracy stops increasing after choosing depth = 6.
Figure 4.2 shows how choosing deep > 6 do not corrisponds to a big increment of accuracy once fixing min leaf = 20.
As we can see in figure 4.3 once fixed 6 as maximum deep, the accuracy decrease accordingly with increasing of Min Leaf and most important the accuracy on test set is not decreasing accordingly with respect to the accuracy on Training set. If the model was overfitted the accuracy on test set would be much lower with respect an high accuracy on training set. Moreover we have noticed that there's a point beetween 150 and 250 as values for min leaf where the accuracy on Test set started

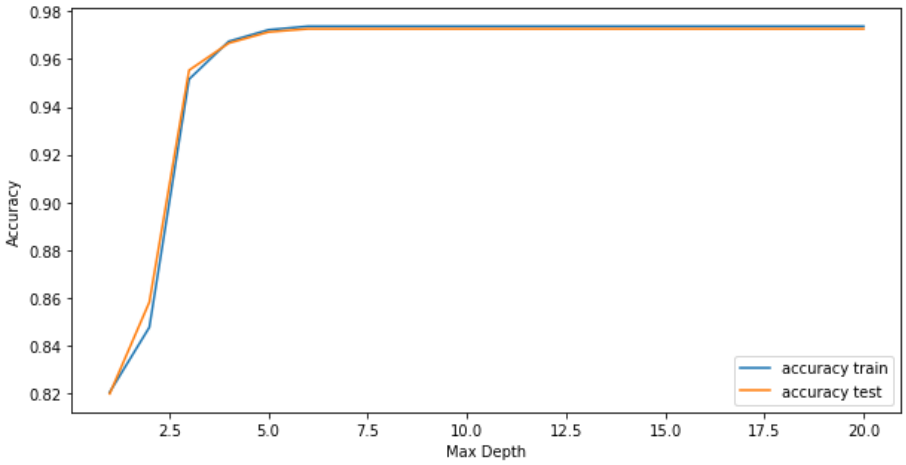

Figure 4.2: Accuracy on increasing Depth

decreasing and then started increasing again.

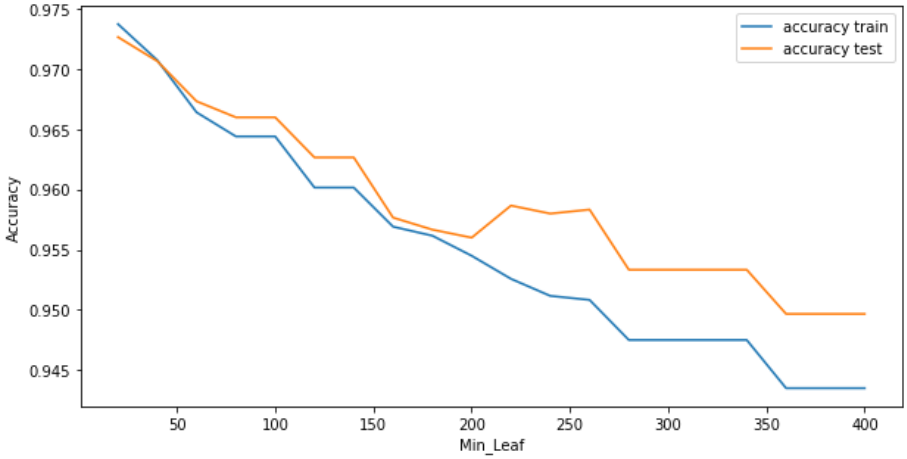

Figure 4.3: Accuracy on increasing MinLeaf

We have invastigated this by plotting values for the confusion matrix obtained by different models with different values for min leaf starting from 20 up to 360 and max depth fixed to 6.

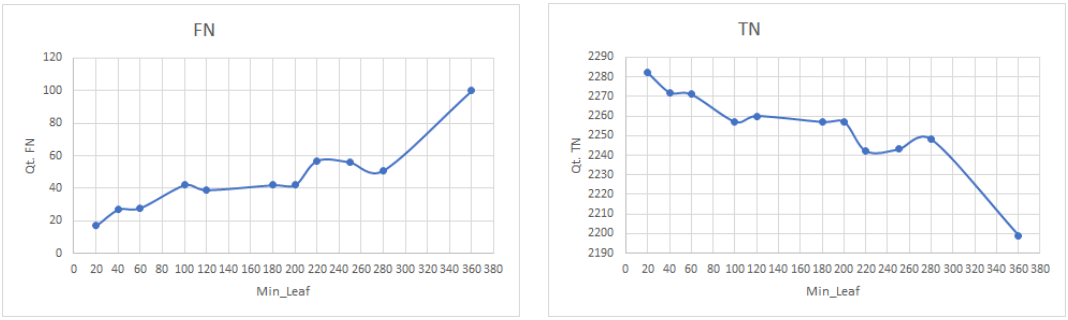

Figure 4.4: Number of False Negatives(FN) and True Negatives(TN)

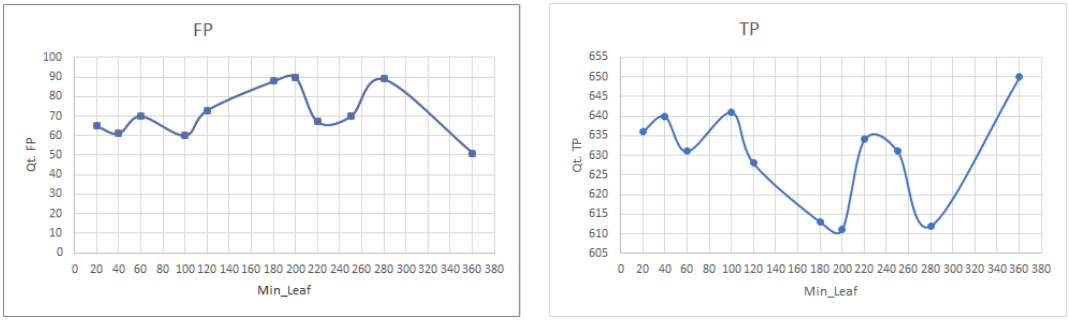

Figure 4.5: Number of False Positives(FP) and True Positives(TP)

In figure 4.4 we can see how between 200 and 220 there's a small increase in good classification of false negatives in true negatives, while there's an increasing of good classification of false positives in true positives as shown in figure 4.5.

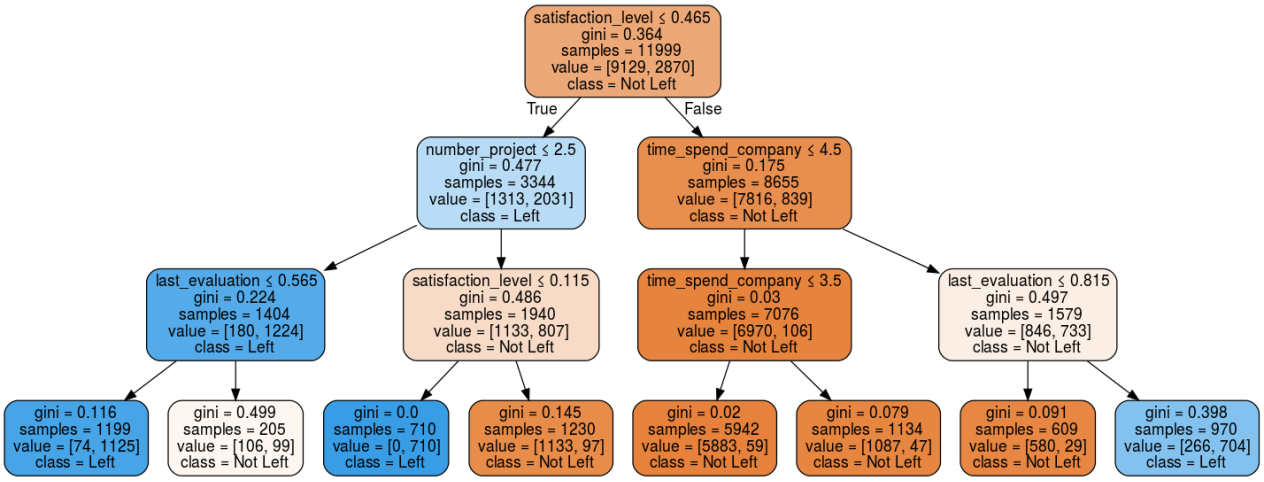

Figure 4.6: Tree Plot with min Leaf = 200 and Max Depth = 3

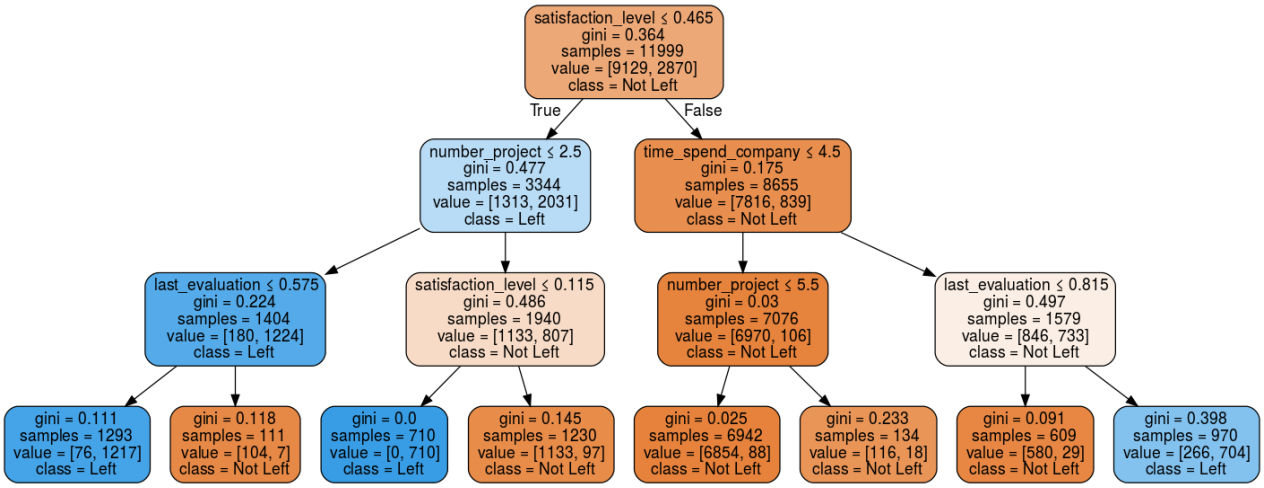

Figure 4.7: Tree Plot with min Leaf = 20 and Max Depth = 3

As you can see in figure 4.6 there's one leaf with a Gini index = 0.499 while in figure 4.7 the same leaf is replaced with one with Gini Index of 0.118 that is a very good improvement. So we choose to set min leaf to 20.

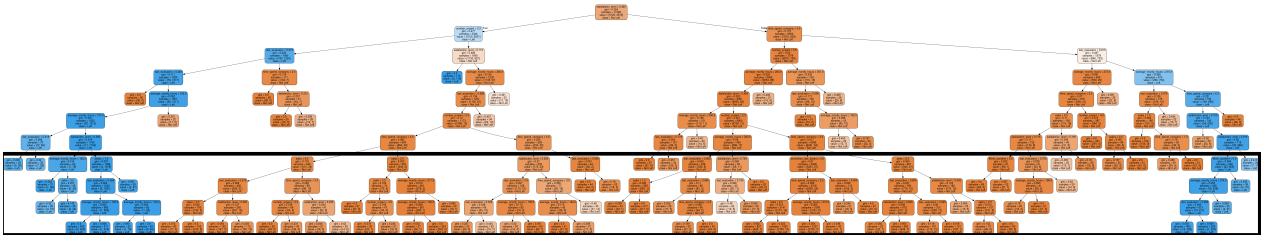

Figure 4.8: Tree Plot with min Leaf = 20 and Max Depth = 10

In figure 4.8 at page 16 is shown how Gini Index improving with respect to increasing the depth of the tree. In particular, we can see how there's no improvement in classification after a depth of 6, so we decided to choose 6 as best depth.

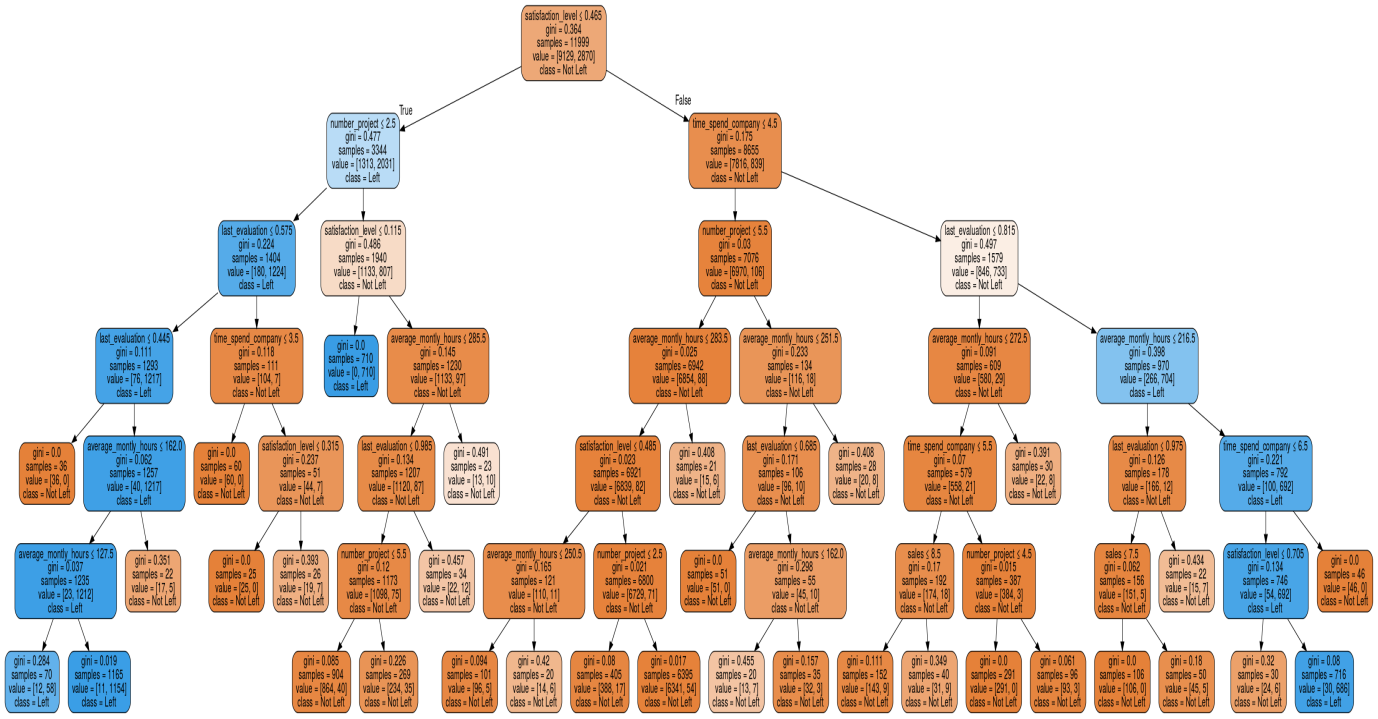

Figure 4.9: Tree Plot pruned with min Leaf = 20 and Max Depth = 6

In figure 4.9 is show the model after the post pruning operation based on max depth. In table 4.1 are showed all the metrics for the selected model. In figure 4.9 at page 16 there're a lot of

Table 4.1: Measures

Accuracy	0.972666666667
Precision	0.973966309342
Recall	0.90727532097
F1_Score	0.939438700148

reduntant nodes so we tried to prune the tree with respect to the gini index fixing the minimum impurity split to 0,15. In this case we didn't lost any usefull information and moreover the new model has an accuracy only 0.01 less than the previous one and the information loss only regards some false positives. Figure 4.10 show how the final model appears and in table 4.2 are showed the same metrics used in 4.1

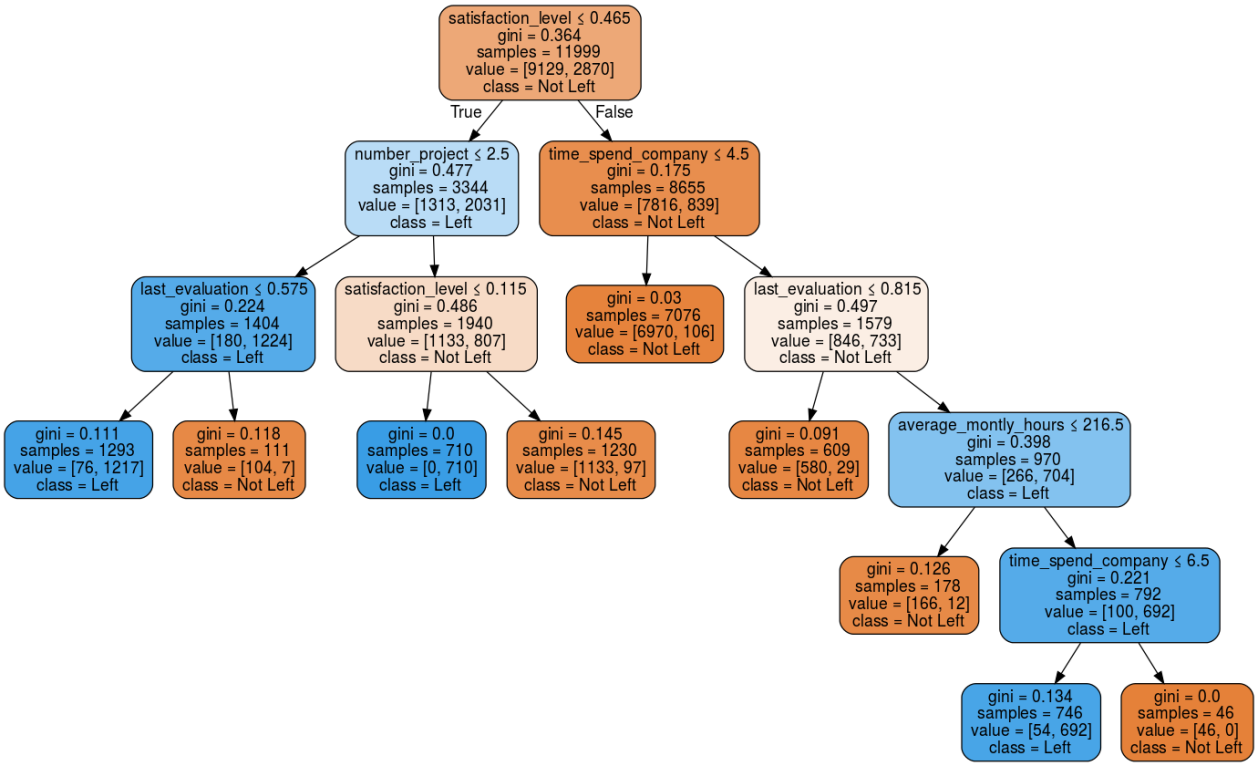

Figure 4.10: FINAL MODEL: Tree Pruning with min Gini = 0.15

Table 4.2: Measures after pruning on Gini

Accuracy	0.969333333333
Precision	0.952451708767
Recall	0.914407988588
F1_Score	0.933042212518

4.1.1 Conclusion

In coclusion, we can observe that :

- The model choosen seems to be not overfitted with an high accuracy, so we did't use other algorithms like random forest for model selection.
- Looking at the model in 4.10 at page 17 We can see that the attribute 'Time Spend Company' seems to be really important when it assumes values greater than 6 (see figure 1.7 at page 4). It seems that if you're good enough ($\geq 0.81\%$ of last-evaluation) and you've been at the company for a mid-long time (between 4.5 and 6.5 years), you're very likely to leave (see also cluster 2 of figure 2.2 at 5).
  Anyway, the most meaningful attribute seems to be 'Satisfaction Level'. In fact we can see that when satisfaction level is above 0.47 is very likely to stay in the company, meanwhile when satisfaction level is very low it's almost sure that you will leave.
- The model agrees with a lot of rules found in chapter 3, for example:
  " '3TSC', 'highsat' –> NoLeft " reflects part the right branch of the tree;
  " '2NoP', '0.5LE', 'lowsat', '3TSC', 'NoProm' –> YesLeft " reflects part of the left branch.