Stockholm University
DSV dept.
Academic year 2017/18

Human Resources Analytics: providing useful insights for employee resignation prediction.

final project for the course of Big Data with NoSQL

Presented by:

- Giorgos Ntymenos
- Giacomo Bartoli

1. Analytical Problem

Human resources management plays an essential role in developing a company's strategy as well as handling the employee-centered activities of an organization. A good human resources staff can increase the understanding of how important human capital is to the company’s bottom line. If someone resigns, it leaves the company with a gap to fill and a potential threat to the company’s profitability. Moreover, all the expertise gained during years will benefit another company, probably a rival.

Imagine yourself being the CEO and hear more and more frequently that some of your company’s best employees resign and you are forced to spend time and resources to hire new ones and train them. You would probably wish you knew better, so that you could have tried to prevent. In this project, machine learning techniques will be applied on a dataset provided by HR department of a company and aiming not only try to find who are the people that leave the company and why, but also to predict who is going to want to leave from the people that currently work in the company. If such information can be known in advance, some valuable actions can be taken in order to avoid resignations that can affect the company’s outcome and, consequently, company’s profit.

2. Data

The dataset is a comma separated value file, where columns are:

Column_name	**Description**	**Data type**
satisfaction_level	Level of satisfaction	Numeric
last_evaluation	Grade of last evaluation	Numeric
number_project	Number of project that currently are working with	Numeric
average_monthly_hours	Average monthly hours at workplace	Numeric
time_spend_company	Number of years spent in the company	Numeric
work_accident	Whether the employee had a workplace accident	Numeric
left	Whether the employee left the workplace or not (1/0)	Numeric

promotion_last_5years	Whether the employee was promoted in the last 5 years	Numeric
sales	Department in which they work for	String
salary	Level of salary	String

The dataset is public and available at the following link:
https://www.kaggle.com/ludobenistant/hr-analytics/downloads/HR_comma_sep.csv

The dataset is already given, but, supposing that we are in a company, it is possible to extract all these data from the ERP system, especially the HR module.

These data are not so big to suggest a classic big data approach. However, this dataset can be taken as example and suppose that such analysis is done over a database containing data coming from different companies. In that case the dataset will easily reach a size typical of big data. Further considerations can be done over the V's of big data. Concerning velocity, these data are batch processed. This means that before storing them on a database, they are cleaned and normalized. Even if they will not provide real-time insights, they will be extremely accurate.

Furthermore, since our vision is that we will have data from different companies, it is very likely that they are going to have a different format, other criteria to evaluate performance etc. Though, there are solutions like preprocessing the data to make them fit our dataset up to that point. For instance, with different ways of evaluating the performance of the employees, it is clear that the classifier would not be so accurate anymore due to different ways of calculation. As a solution, it is possible to calculate the mean of ratings of each company and then normalise everything and force them to have as mean the global mean that would be the mean of means. Then, this attribute could be converted as categorical (high,medium,low).

Considering all these characteristics, we will firstly focus on a solution suitable for our dataset then we will imagine a more scalable solution, where data change often, and then scalability and fault-tolerance are prioritized.

3. Method

Storing Method:

For storing method we chose HDFS. It is a distributed file system that allows access to files from multiple hosts across a network. Its two main advantages are its ability of handling variety and volume. It is an ideal choice for analyzing massive amounts of data, so it is a pretty solid solution when we need to scale. Moreover, it can accept data in about any format which saves a lot of time from data transformation and further processing. The main issue is that we should be able to execute long-running analytics and queries over the data in the same time.

Therefore, HDFS is the best option in comparison with a NoSQL database. We have considered as alternative NoSQL databases like Mongodb for example. Mongodb is capable of adding far more flexibility than Hadoop, but the latter, and consequently HDFS, is built to handle big data. A solution like that is preferable in real time analytic tasks and even if it is quite effective in working with large volume of data, HDFS is known to perform better.

Analysis method:

The problem is based on labeled data, so it is clear that it should be used a method for solving a classification task.

The label is represented by the column “left”. We decided to use decision trees, since they are very simple, they do not need too much time to built and they can scale easily. Of course, it is a classifier that has high variety and tends to overfit, but these problems can be dealt with different techniques like pre or post pruning. In this case an easy way to implement is build the tree and then prune it until the performance stops improving.

We did considered alternatives like K-NN and SVM. They are interesting classifiers with many advantages. However, the criteria of interpretability is what we consider as the most important. Decision trees’ behaviour is easy to interpret and the knowledge produced can be understood clearly also by people who have a lack of technical background. This happens, since k-NN and SVM are black box methods in contradiction to decision trees that are a white box method. Finally, k-NN is a lazy learner, so it would be highly time consuming using this model to perform a lot of queries.

Evaluation Method:

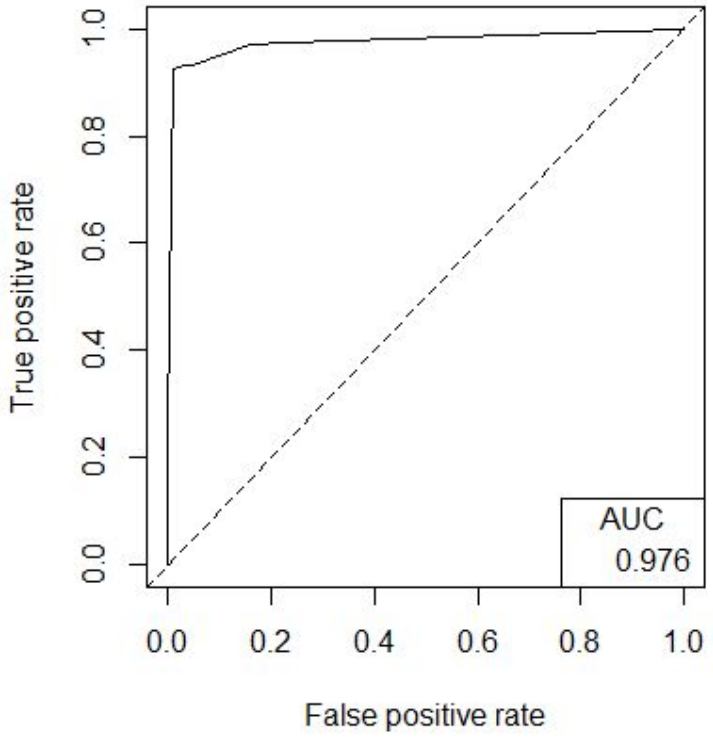
Accuracy:

dtPred	0	1
0	2235	55
1	25	685

Accuracy = (TP+TN)/(TP+TN+FP+FN) =
Accuracy = (2235+685)/(2235+55+25+685) = 0.97

Accuracy is very high, which means that the classifier is very close to ideal.

ROC Curve:
This is a better evaluation method, because accuracy has some flaws that can make you think you have built a good classifier, while this is not the case at all. For example, if you have 100 items, 99 belonging to class A and 1 of class B your classifier will classify everything A and you will get 99% accuracy.

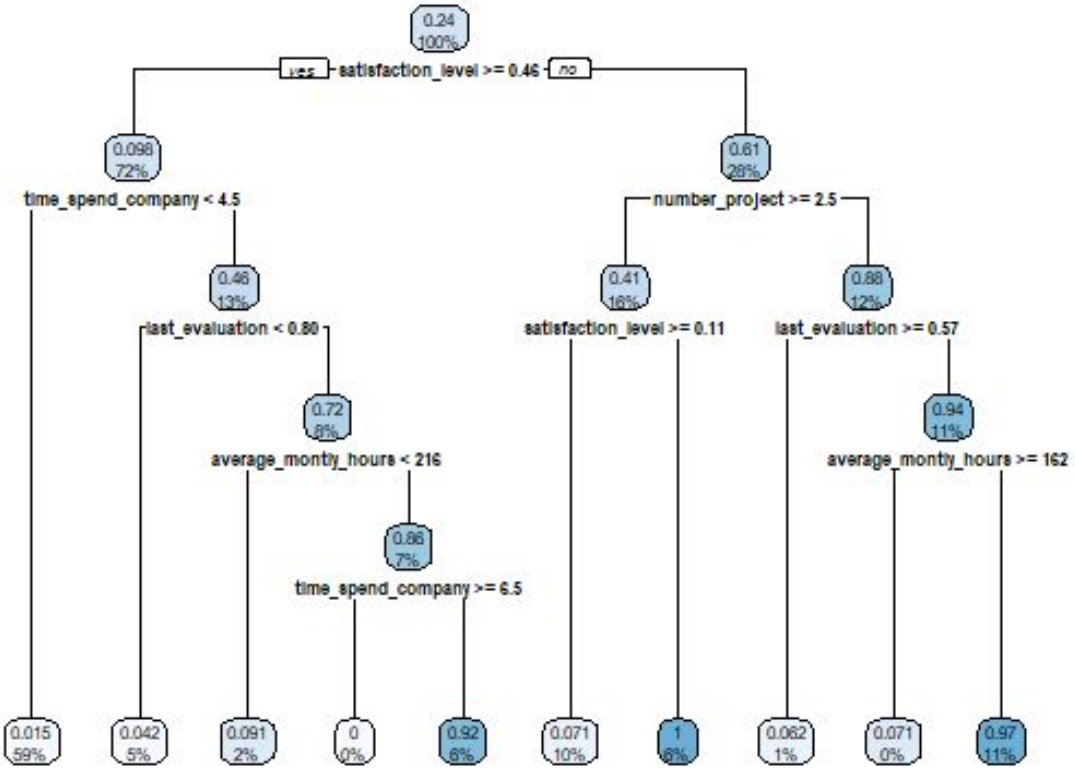


The closer it is to the upper left corner, the better the classifier is.
In this case the classifier appears to perform really well.

The code was written in R and it is available on the following URL:
https://github.com/giacomobartoli/BIGDATA/blob/master/final_project/FP.R

4. Results and analysis

Here we can see the tree extracted from training data.
From that picture valuable insights are provided concerning people who tend to leave.



Each node shows - the predicted value, - the percentage of observations in the node

We can see that the most important factor, which is also the factor that we make the first split, because it gives the most information gain, is the **satisfaction level.**

If an employee is in general satisfied(more than,or equal to 0,46/1) it is very unlikely that they will leave.The chance that they will resign is 10%. If this is the case,mostly those who will leave are those who are evaluated very high(more than 0,8/1 and they leave with probability of 47%).When we add to this set,those who work many hours(over 216/month) and have spent a lot of time in the company the probability for them to leave increases 92%. It is worth noting that these people are only the 6% of our whole data. In order for an employee to be in this category they are supposed to be very efficient and being an hard worker, but these skills make him/her attractive to other companies, which can offer better job opportunities. Probably this is the reason why they choose to resign.

On the other hand, if employees are not satisfied with the company they are likely to leave with a probability of 60%. Obviously it makes sense, because since they don't feel satisfied they will try to find something else.

The next factor our classifier investigates, is the number of projects they currently work on. If it is higher than 2.5 the chances of not leaving increase. This probably means that some people feel committed and they do not want to leave the company while working on a lot of projects. Then, it checks the satisfaction level again: if it is really low(less than 0.11) it means that these people are extremely unhappy and they are going to leave. Otherwise they will probably stay.

But, if the projects are less than 2.5 the chance of an employee to leave is 0.87. So what the decision tree does afterwords, is to check their evaluation and working hours and almost every one of those employees not only are evaluated under 0.57 but also don't work many hours. This means that they are not so efficient and they are likely to leave the company, mostly due to their insatisfaction

To sum up,there are three basic cases when people leave:

- When they perform really well and although they are generally satisfied they feel underestimated,or they find better job opportunities due to their skills.
- When they are not satisfied from the company at all.
- When they are not satisfied but not very effective either.

5. Discussion

- Scaling-Generalization:

Our method can scale using data from a lot of companies and as a result produce more accurate results based on the market as a whole rather than address only to one company. Of course, this is a challenging task since the data might not be in the same form. A preprocessing phase is needed in order to integrate everything into a global scheme. This scenario can lead to different issues. For example, it is possible to reach a point where different metrics are used but not all of them would be useful. As a result we would have very sparse data with not so much useful information. This problem can be fixed using dimensionality reduction techniques,like PCA.

According to what we studied during the course, we need some mechanism/architecture for assuring consistency and availability of data in case of disasters.

So, one possible solution could be replicate data over different servers. In this case, all the users that need access to data will not be connected to the same server. However, everytime data are updated we must propagate the update to all peers of the distributed system. In this way traffic on the network is balanced and servers can achieve a faster response time. We would like our system to guarantee consistency, availability and partition tolerance, but, as the CAP theorem states, we can only choose two of these properties in a distributed architecture. Thus, we choose availability and partition tolerance over consistency, aiming to enhance performances, avoiding loss of data.

Value of method/analysis and results:

-Method:

Using decision tree for the analysis of this specific dataset we can gain a huge advantage, which is fast calculations, since attributes with missing values or sparse data are not taken into consideration.

Even if accuracy and AUC seems optimal, there is another reason why decision trees are one of the best options, while choosing a classifier. It is fundamental to consider that businesses and companies continuously change. People are hired while other just resign and the internal processes leaded by the company are constantly revised by managers. This means that also the decision tree

that is used to predict who is going to leave has to be updated and maintained according to the way the company evolves.

A lot of queries will be required to be performed, but in case that there is already a decision tree, this tree can be used to classify new data avoiding re-training the model from scratch every time we have an insertion or an update. The training phase will take place when the IT department team that is working on the project believes it is required.

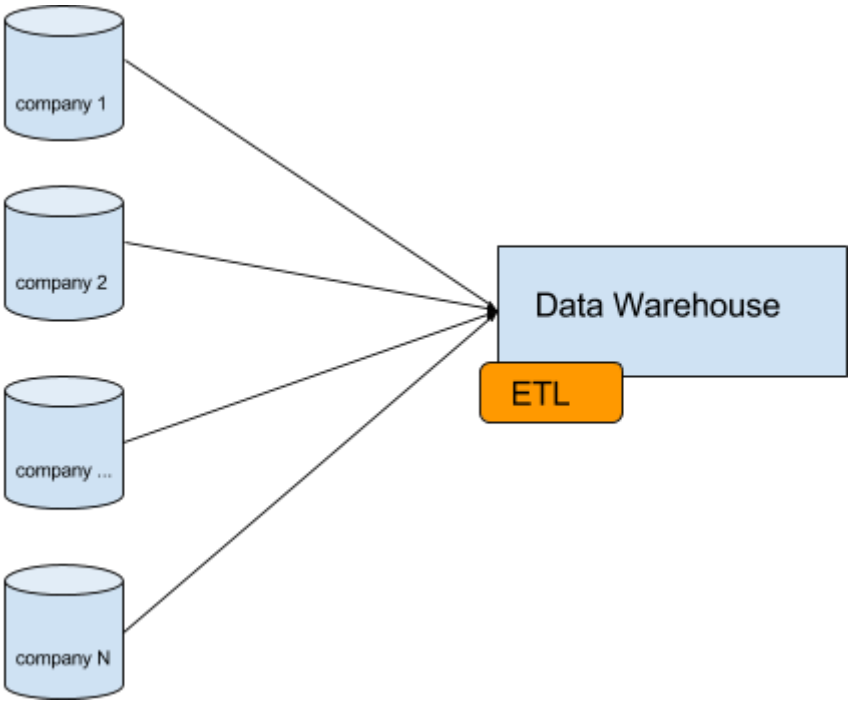
Results:

From the results it emerges that some employees that leave are among the best ones and the company would not want to lose them, so it would be very useful to find out before they get another job opportunity. It seems to be a very interesting and valuable problem. As a result, if companies use it, they can predict which people are going to leave and have the ability to decide if they want to take some valuable actions.

Some possible solutions would be increasing their salary, promote them or decrease their working hours in order to decrease the possibilities of a resignation. However, this is an HR department responsibility.

Potential extensions:

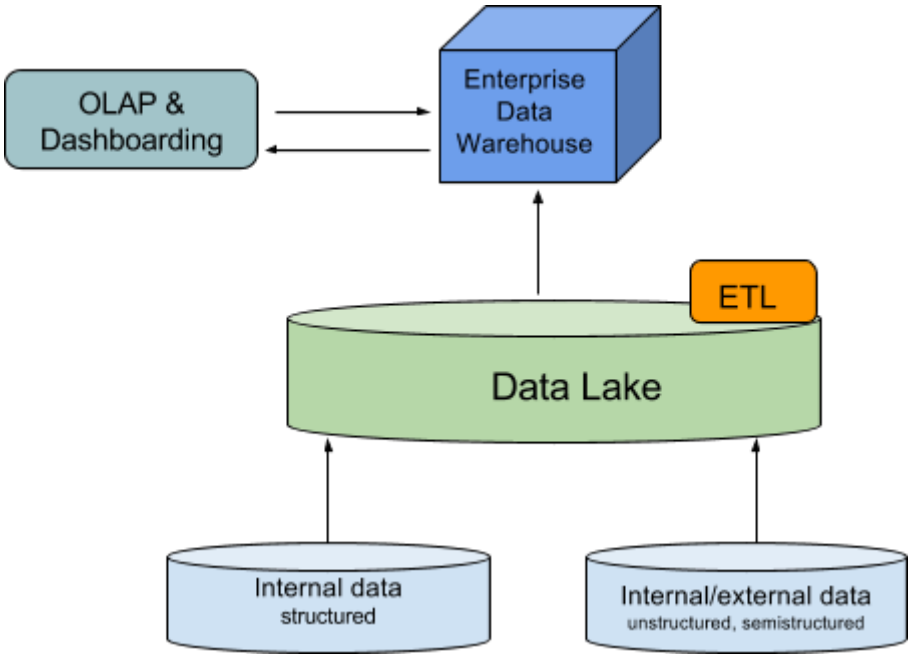
This data analysis can be extended to different cases. Initially, it can be supposed that datasets come from different organizations always handling structured and batch processed data:



We can make a standard schema for data and process every new data in order to integrate them to our global schema with statistical techniques that will help us have a more stable and accurate model when the volume will be huge and the data much different among companies.

As already mentioned, each company sends data of its employees. These data are firstly preprocessed through the ETL module and then integrate into the Data Warehouse. Handling data coming from different organization remarks the importance of the volume. In fact, a large amount of data, even if it requires more network availability and storage capacity, assists data scientists to apply machine learning and have more accurate predictions.

Later on, it is possible to design a system where not only batch data are analysed, but also emails, internal instant messaging services and other kind of data are considered. Introducing unstructured data, the scenario changes radically:



The most important element, that differs the previous design from this one, is the Data Lake. It is essentially a large store repository, where data coming from different sources and with different nature (structured, unstructured, semi structured) are collected. It does not really matter which kind of data is it, they all flow into the Data Lake. Its tasks are:

- holding data until is needed
- allowing rapid ingestion of datasets without extensive modeling
- scaling large datasets while delivering performance
- supporting advanced analytics

Data scientists work on the top of the data lake and they can do data blending, exploratory search, data mining, text mining, sentiment analysis, statistical analysis etc.

The role of the ETL module is the same as the previous traditional design: preprocessing, normalization and cleaning data coming from different sources.

Later on, all these data are join into the Enterprise Data Warehouse. This module interfaces directly with the OLAP and Dashboarding system. Here, HR specialist can visualize the final report and find out who is going to likely leave the company.

The analytic process could be extended considering also data coming from outside the company. It is known that today people tend to complain and discuss their issues using social networks. Thus, monitoring social media posts concerning our company can be another way to identify potential discontents at work. The activity of listening

social conversations (post, forum, hashtags) for extracting what people think about your business or your company is called **Social Media Monitoring** (SMM).
SMM is strongly related to the idea of brand loyalty and it provides useful insights in order to answers questions like:

- what do people think that working in my company would be?
- why newest employees have chosen my company?
- how my company is perceived from the outside?
- why people looking for job should choose me rather than competitors?

Hence, the key to avoid employees resignation relies in a strong analysis of data we already have. This analysis can be done over a single datasets or multiple datasets coming from different companies. Later on, data are integrated into the data warehouse. To improve accuracy results, it is a good idea to take into account also unstructured data such as emails, instant messaging conversations. For sure, handling structured and unstructured data requires a shift in the system architecture. A data lake is needed to collect all these kinds of data. Eventually, SMM tools can be used to mine what people think about our company, including and collecting data coming from outside the organization.