1. Objectives / Research Questions:

- Obtain groupings of different work environments or job titles.
- Check if there are significant differences between the company size and it's location.
- Create, train and test a model able to predict the salaries of determined data.
  - Regression models
  - Classification models
- Transform salaries based on the cost index of each country

2. Raw Data:

The Dataset consists of 11 columns and 1332 rows of data.

These variables (columns) are: work_year, experience_level, employment_type, job_title, salary, salary_currency, salary_in_usd, employee_residence, remote_ratio, company_location and company_size.

From all these attributes, only salary and salary_in_usd contain continuous values, the rest of columns have categorical values.

3. Data Analysis:

The dataset consists of 1332 samples. In order to have a first view of what the information we have is like, a series of graphs have been made to visualize the information graphically and thus be able to compare.

Firstly, we consider that the dataset information regarding the different values that each of the characteristics that form it can have is scarce, such as "employment_type" or "company_location", since most of the samples share the same values for those characteristics. In addition, we consider that "salary" is a feature that does not provide relevant information, since we already have "salary_in_usd", which provides the same information and also in the same currency, so it will be much easier to work with it.

Also, we have been able to observe that the "time" attribute does not provide any useful information to be able to carry out our objectives. As we see in the salary distribution, there is an outlier wich should be removed, as is an impresive high salary in Senegal.

4. Data Preparation:

Once the data set and its characteristics have been analyzed, we first eliminate the salary_in_usd column. Following are a number of transformations and feature additions:

- In order to carry out some of the supervised and unsupervised data mining methods, all the locations have been grouped by continents, since they found a wide variety of locations from different parts of the world, but more than 85% of the The samples are from the USA, so the rest have a very small number of samples and may hinder the accuracy of some prediction models.

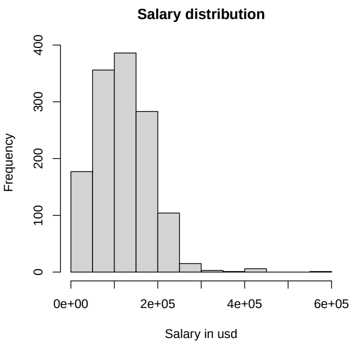
- The "salary_in_usd" attribute has been categorized into "high", "medium" and "low", adding it to the dataset as a new characteristic called "quartile". First, it was carried out using quartiles, but we saw that the best option was to do the limits manually.

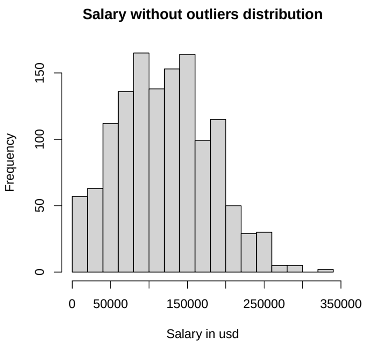
- The "work_year" and "remote_ratio" attributes have been categorized, since the number of possible values for each is discrete and small. "remote_ratio" can be worth 0 ("no remote work"), 50 ("partially remote") and 100 ("fully remote"), therefore, when categorizing "remote_ratio" a string has been assigned to each value for a better understanding of the values: 0 -> "NR", 50 -> "PR" and 100 -> "FR".

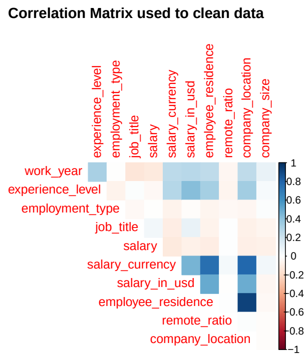
- Those samples where "employment_type" !-> "FT" have been eliminated, since more than 95% of the samples share this value and there is underfitting in the others.

- All the different values of "job_title_grouped" have been grouped into more general sets that encompass those jobs that are of the same scope.

- The cost living index is different for each country so we investigated about that and then, we got a csv with all the cost living indexes, so we created a new column called large_country with the complete name of the countries, then we associated each cost index to each sample of the dataset, and finally we made a new column transforming the salary by multsiplying it wsith the ratio of the cost index in USA and the cost index of the correspondent country.



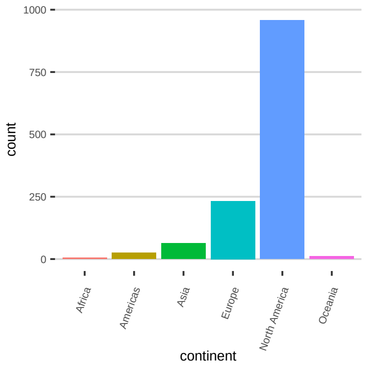


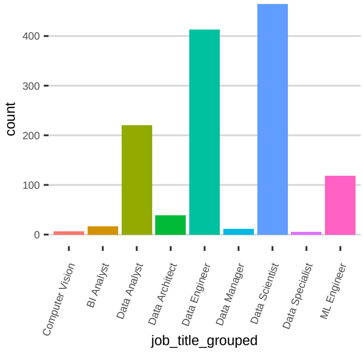


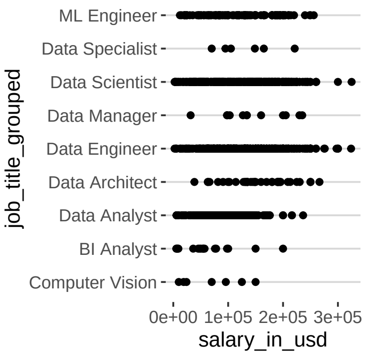
Distribution of the grouped data

As we can see in grouping the samples by continent, the difference between North America and the rest is still quite considerable, but the distribution of the data has improved. An alternative could have been to group the samples between “Americas”, “Europe” and “Rest of the world”, but in our case we are not interested, since we take into account the social and economic context of each continent.In addition, we can observe the new distribution of the data taking into account “job_title_grouped”. Due to the way in which we have paired the different titles, we see that those that predominate are those related to “Data analysis”, “Data engineer” and “Data Scientist”.

Finally as we see in the graph, all jobs are paid in a similar way, since we find people with different salaries in the same jobs. But, those in which there are people who earn a lot more than the rest is in the fields “ML Engineer”, “Data scientist”, “Data engineer” and “Data architect”. Naturally, the points on lower salaries are more dense. Finally, it should be noted that due to the scarcity of samples in certain works, it can cause us to have an erroneous perspective.







Salaries on company sizes and experience level

Also is important to check how salaries move on company sizes and the experience level of the workers. The best paid company size is the medium size with a mean of 131082 usd per year and the less paid are the small ones, with 80136 usd per year. The workers with entry level are the worst paid, wich was very expectable, and the best paid is the expert or director level wich is also very near from senior level.

5.1. DESCRIPTIVE MODELS: Clustering

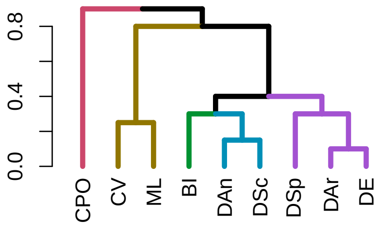
Clustering will be applied for us to be able to have a better understanding of our Dataset regarding the Job Titles and their differences. In other words, we want to obtain groupings of different work environments so we can draw conclusions from them. **Manual Distance matrix:** For us to be able to apply clustering techniques to our Job Titles, we need to create a Distance Matrix which will be used as an input in the clustering methods. To generate this Distance Matrix, we had to investigate about what these Job Titles consist of so we can actually know how similar they are to each other. Next we have created a distance scale in which 0 represents total similarity between 2 Job Titles and 1 represents that 2 Job Titles are completely opposite. So, with this distance scale and the research done previously, the Distance Matrix generated is the following:

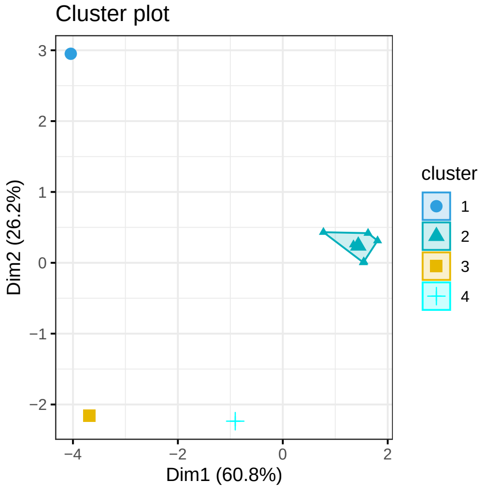
Note that: **CV** = “Computer Vision”, **BI** = “Business Intelligence”, **DAn** = “Data Analyst”, **DAr** = “Data Architect”, **DE** = “Data Engineer”, **CPO** = “Chief Product Officer”, **DSc** = “Data Scientist”, **DSp** = “Data Specialist”, **ML** = “Machine Learning”

	CV	BI	DAn	DAr	DE	CPO	DSc	DSp	ML
CV	0.00	0.75	0.75	0.75	0.80	0.90	0.75	0.75	0.25
BI	0.75	0.00	0.30	0.40	0.40	0.75	0.30	0.40	0.50
DAn	0.75	0.30	0.00	0.25	0.25	0.75	0.15	0.30	0.50
DAr	0.75	0.40	0.25	0.00	0.10	0.75	0.25	0.30	0.40
DE	0.80	0.40	0.25	0.10	0.00	0.80	0.30	0.30	0.40
CPO	0.90	0.75	0.75	0.75	0.80	0.00	0.80	0.80	0.85
DSc	0.75	0.30	0.15	0.25	0.30	0.80	0.00	0.15	0.50
DSp	0.75	0.40	0.30	0.30	0.30	0.80	0.15	0.00	0.50
ML	0.25	0.50	0.50	0.40	0.40	0.85	0.50	0.50	0.00

Hierarchical Clustering and K-Means: With the Hierarchical clustering method, we are able to visualize the clusters formed by the 2 closest Job Titles in each iteration.

Finally, to assure that the clustering results are correct, we use the K-Means method so we have 2 clustering techniques that most likely produce the same result.





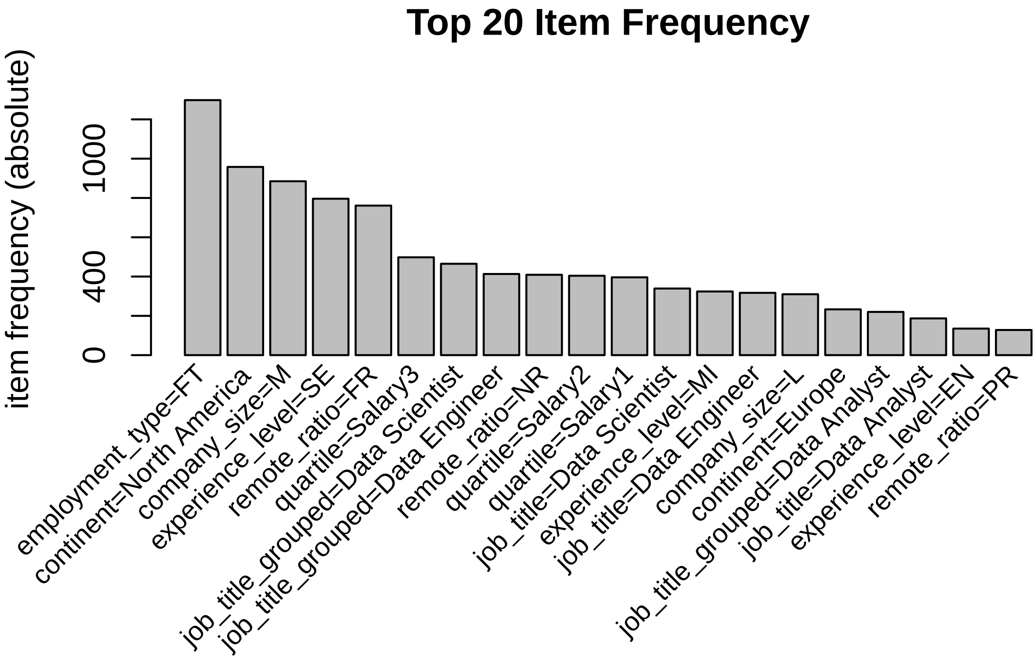
Once the K-Means algorithm has finished, this plot shows us the the results are: One cluster with “CPO”, another cluster with “CV”, one more with “ML” and finally one with “BI”, “DAn”, “DAr”, “DE”, “DSc” and “DSp”. In conclussion, we can observe that there are 4 main work environments, clearly distinct, divided by result clusters. One of them contains 6 job titles, and the rest are clusters of a single job title. By now, we have a better understanding of the different job titles included in this dataset.

5.2. DESCRIPTIVE MODELS: Association rules

The objective of applying AR to the dataset is to see which characteristics are most related to each other, taking into account each one's appearance frequency.

To do this, we first prepare the data that is necessary to be able to apply AR. Looking at the characteristics, we see that the data that we are not interested in are "work_year" and "salary_in_usd", since the first does not provide any information, and the second is a continuous value that we already have represented as a category in salary feature. In addition, the rows of the df are transformed into transactions to be able to apply the AR functions.

Firstly, we want to see which items are the most frequent in the entire Dataset, taking into account a support of 0.2. Later we will see how we will not be able to work with a support greater than 0.3, since the amount of information in the Dataset is quite scarce and the number of rules decreases considerably.



Using the eclat function from the "arules" library, we calculate the most frequent items that have a support greater than 0.2. The set of items that it gives us is 101, that is a considerable number to be able to obtain the necessary rules.

	items	support	count
[1]	{employment_type=FT}	1.0000000	1298
[2]	{employment_type=FT, continent=North America}	0.7380586	958
[3]	{continent=North America}	0.7380586	958

At first glance, looking at the summary we can see that the items with the highest frequency are "full time","North America","Senior","Full remote" and "Medium".

Making the Rules

Next we proceed to create the rules. To do it, we have created all the possible rules from our most frequent items, with a confidence greater than 0.45, since we consider that it is the correct measure for this case since we do not have much input information. As we can see, once the redundant rules are eliminated, the result is 77 rules in total.

	lhs		rhs	support	confidence	lift
[1]	{continent=North America}	=>	{employment_type=FT}	0.7380586	1.0000000	1
[2]	{employment_type=FT}	=>	{continent=North America}	0.7380586	0.7380586	1
[3]	{company_size=M}	=>	{employment_type=FT}	0.6818182	1.0000000	1

Once all of them have been inspected, those that have called our attention are all those that involve the appearance of "salary", since they are the ones that will best help us understand those characteristics that are more related to "salary", whether good or bad.

To have those in which "salary" appears, we group all of them in which lhs = quartile. Next, We do the same with those where rhs=quartile. To get the most rules where "salary" appeared, support and trust have been lowered.

To check if there are redundant rules, all are ordered based on the elevator value, both AR1 and AR2.

Once we have them ordered, we have chosen to check the redundancy automatically using "is.redundant". We have verified that in AR1 there is no redundancy and therefore we are left with the same number of rules.

Instead, in AR2 we find 16 redundant rules out of 31, therefore, we keep the non-redundant ones.

	lhs	rhs	support	confidence	coverage	lift	count
[1]	{quartile=Salary1} =>	{continent=Europe}	0.1332820	0.4368687	0.3050847	2.433715	173
[2]	{quartile=Salary1} =>	{experience_level=MI}	0.1363636	0.4469697	0.3050847	1.790638	177

|| || || ||

Finally, it only remains to analyze the rules and draw conclusions: Looking at the rules of AR1 and AR2, we can conclude the following:

- People who have a high "salary" are usually seniors, who work "Full remote", in a medium-sized company in North America, full time and who work as Data scientists or Data engineers.
- People who have an average "salary" are usually seniors, who work full time in a medium-sized North American company "full remote".
- People who have a low "salary" are usually Mid-level, who work full time in a large company or median North America or Europe "full remote". Note that the probability of it being in a medium-sized company is greater than in a large one

Association Rules conclusions: In conclusion, taking into account both these rules and those generated at the beginning of the analysis, the following can be said: 1. Most of the workers in the Dataset work in medium-sized companies in North America, regardless of whether they have a high, medium or low salary, 2. Most work "Full time", either "Full remote" or "No remote", 3. The companies that pay less are the European companies, 4. Seniors usually have a medium-high salary, 5. More than 50% are medium-sized companies from North America, 6. In medium-sized companies they usually work "Full time", 7. Most of those who work "Full remote" work in American companies.

6. PREDICTIVE MODELS: Machine Learning Models

Linear model

At first we will try using a linear model to predict the salary in usd, and as we observed, it fits better by transforming this variable with log function. We are using work year, experience level, job title, company size and company location to fit the model and predict the results, because as we saw previously, these are the most correlated columns with the salary_in_usd variable.

Multiple R Squared: 0.731417

Adjusted R Squared: 0.7035301

This Adjusted R Squared coefficient tell us that the linear model is capable of explaining the 70% of the data. To state that a model is reliable could be better get a 85% or more of explanation, but the distribution of the dataset doesn't help to much and we lost many columns with problems of multicollinearity so is not much bad. Also the log transformation improved a 10% the explanation of the model because helps to flattern the distribution and is a technique very used in economy for manipulating currency measures.

Naive Bayes & Random Forest

We will now try to get this problem to a classification one by trying to predict in which quartile a worker should be. At first we are creating a Naive Bayes and a Random Forest classifiers. We will use Naive Bayes because it is a really confident classification model and we think it should perform pretty well, and Random Forest because it is one of the best models in terms of accuracy.

```
Models:          Naive Bayes                          Random Forest
```

	Salary1	Salary2	Salary3	Salary1	Salary2	Salary3
Salary1	83.0000000	27.0000000	8.0000000	83.0000000	26.0000000	9.0000000
Salary2	20.0000000	49.0000000	15.0000000	22.0000000	49.0000000	16.0000000
Salary3	15.0000000	45.0000000	126.0000000	13.0000000	46.0000000	124.0000000
Precision	0.7033898	0.5833333	0.6774194	0.7033898	0.5632184	0.6775956
Recall	0.7033898	0.4049587	0.8456376	0.7033898	0.4049587	0.8322148
f1_Score	0.7033898	0.4780488	0.7522388	0.7033898	0.4711538	0.7469880

	Train Accuracy	Test Accuracy
Naive Bayes	0.6670	0.6649
Random Forest	0.7297	0.6598

The first three rows of the table are used as a confusion matrix and the others are the metrics for each class for the two models. As we can see, we have obtained good results on our testing predictions, not as good as the linear model, but still higher than 65%. It is visible that the medium salary have the worst metrics for accuracy and recall because the model is confusing to many samples from the low and high salary into the medium one, and this is probably because we have to define better the separation by groups of the salary, but we tried to separate salary into 4 and 5 groups and we got worse results, so even if three groups is a bit general, it gaves us the best performance.

Logistic regression

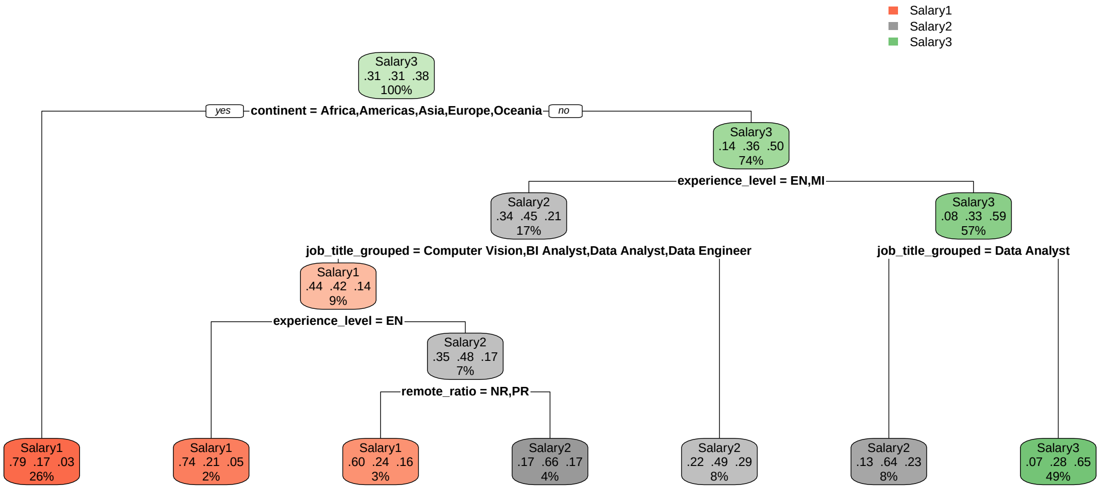
First, we get the dataset with "quartile" column as a binary class column, by creating different datasets with the same data. The dataset will be splitted into the 3 quartiles, and a linear regression will be obtained for each one. Once we have all three models built, we can predict the quartile where an input belongs to comparing all probabilities.

Then we use a logistic regression for each of the dataset generated, so we can iterate over the test set and get the model it fits best. An algorithm to calculate the accuracy of the models has been designed as follows: Each row of the test set is used to predict the quartile of every model, and the one which gives the highest probability of a true prediction is considered from its salary category. Then we check if actually this last prediction belongs to its quartile and we add it to a true predictions counter. After all test rows have been filtered, we can calculate the accuracy of the models dividing the true predictions by the number of test rows. The accuracy obtained is: 0.6623711

From this index we can conclude that with a certainty of a 66% we will get a true prediction of salary category. We know that the algorithm is very reliable because we tried out building the models with an included r package and it gave us a 5% lower accuracy. We also tried to rising the quartiles to 4 for more precision, but its loss in accuracy was too big. The following logistic regression builds a tiny neuronal network with the nnet package. This package gives a true output for the best option of salary like our models. The NNet Accuracy obtained is: 0.6185567

Decision tree

A decision tree has been built to check how it performs by classifying our data in the 3 different salary__in__usd quartiles. It has been first done excluding the salary__in__usd in the train dataset, company__location and job__title because they have too many categories, which leads to an unending execution of the rpart library. We still have the columns continent and job__title__grouped, which are more useful to classify with more accuracy without overfitting, and gaining simplicity in the tree. This also explains that the tree is binary, as it is the most common solution given in actual algorithms. 'rpart' uses CART which applies Gini Index to order the partitions. This type of algorithm is very useful in our dataset because our partitions in salaries are equally sized and we don't provide a lot of classes to the tree, as they have been grouped.



We can now conclude that the continent is the most important predictor to split the data. The first split separates North America from the rest of the continents. That may happen because of the dominance of data provided from that "continent". This is supported by a probability of 79%, and in cases like Europe which is dragged to the left node, we maybe won't find that many cases of a low salary. Overall, the accuracy is at 0.67, which means that the tree is not randomly generated, but it fails pretty often and explains some of these biased categories. As we go down though the nodes, we can see that when a worker has an entry-level experience or mid-level, they will be only classified into low or mid salary, which makes a lot of sense. From the higher levels of experience, we can distinguish one job that never gets a high salary: Data Analyst. The levels of job__title__grouped not showed in the tree are: Data Architect, Data Manager, Data Scientist, Data Specialist and ML Engineer. The attributes excluded from the tree may be because they don't have enough impact on the outcome, or they have a high correlation with other predictors.

7. Applying the transformed salaries by cost index:

At the begining of the document we stated a transfomation based on the cost living index. This help us to see how really “rich” is a person and to state wich is in really the best paid jobs and countries. So we think that probaly if we try the regression model with the transformed salary as a target, the slope coefficients of the different company locations and job titles could change and become into new ones.

Multiple R Squared: 0.5898409

Adjusted R Squared: 0.5471086

And after the execution of the linear model we get worse accuracy and the coefficients are intact, so the attributes are contribuying the same, but let’s try to understand why is this happening. Could be than even if we adjusted the salaries by the living cost index, the workers from other countries are still underpaid or overpaid. To state this lets make the mean for the most common job Data Scientist.

Country	Salary	Salary_transformed
United States	149171.11	149171.11
Australia	83793.00	80565.91
Belgium	68173.00	75239.71
Brazil	38332.25	79978.53
Switzerland	120221.00	76217.17

So to conclude this point, we see in this table, that US is still pretty dominant in terms of salary. Is very interesting that in Switzerland having a salary of 120.000 usd per year is less than getting 38.000 usd per year in Brazil wich at least make significance this approach of transforming the salaries, because is really important where are you going to live while you are working. It is possible that the other countries doesn’t pay more because the most important companies are on Silycon Valley or because in USA there is more investment on IT jobs.

8. CONCLUSIONS:

To conclude this assessment, there are some things we wanted to note about the different methods we used to analyze, transform and use the data. About the dataset itself we must say that it is quite difficult to work with, because it has too much samples with the same value in different columns, such as “work_time” and “company_location”. We also found a high correlation between more than 2 pairs of columns, which made us reduce even more the columns we would use. After that, transforming the data has also been pretty difficult, as we previously said, we had to transform it to ease the use of the dataset in order to get our models to perform better. Finally, when using the data, even though we tried to simplify and complete the data to get a better dataset, we weren’t able to produce a subjectively well-performing model (>75% accuracy), but we got close to it and learned a lot about data treating and machine learning while trying.

We found these difficulties because we are used to see “toy dataframes”, where all the variables are perfectly distributed and there are many columns highly correlated with the target, in other subjects from last years, so we also appreciate the opportunity to manage a real dataset where there is some useless data and we have to work rough to obtain important resources.

About the results obtained, we saw in the Association Rules that one of the most important factors that determine a worker salary is the experience, as it is very frequent to get high salaries for senior level workers. Also the mid size companies in North America pay very well so working on Canada and USA is a pretty good option. On the decision tree plot it is also stated that the medium and hight salaries from this dataframe are directly asociated to work on North America and this is also remarked when we transformed the salaries by ratio of cost living index from USA with the other countries and the USA is still dominant in terms of salary, so there is no clue that in terms of IT works, USA is the best option.