Salary prediction project

Project of predicting salaries based on job descriptions.

By: Juan Vicente Peluso

GitHub notebook

Table of contents

1. Introduction

The dataset, features and target value

The data is split into 2 CSV files to train the model, one with the features and one with the target variable. Other CSV file has the jobs for which we need to predict the salaries. The features are the descriptions of the jobs of a group of companies, being the columns the following:

- **jobId:** The ID of the job.
- **companyId:** The ID of the company.
- **jobType:** The description of the job.
- **degree:** The degree of the employee.
- **major:** The university or college field of specialization of the employee.
- **industry:** The field to which the company belongs.
- **yearsExperience:** The employee years of experience on the job.
- **milesFromMetropolis:** The distance in miles, the employee lives away from his/her place of work.

The target variable is the salary each employee receives, being the columns the following:

- **jobId:** The ID of the job.
- **salary:** Salary amount paid for that job.

Problem description

As we know, the job market is very complex and competitive. For those who want to be hired, and for the companies which want to have the best talent at their service. Knowing in advance, how much are you going to get paid, or will pay for a certain job or profile, could be helpful for both the employee and the employer.

With the data available, can we develop a model that predicts the salary for a specific job and profile? How accurate can we get? Are the features enough for a good predictive model?

We will work to develop a machine learning model, that will try to predict the salaries accurately as possible, for each job we have at our disposal.

2. Data quality check

Once defined the classes, the data files of the job description and corresponding salaries were uploaded into Pandas dataframes, each file has 1.000.000 records each.

We've merged both files for training tasks, then, during the data consistency checks, we've found no NaN or duplicated rows, but 5 rows had a **salary 0** (zero). Those rows were deleted

3. Descriptive statistics

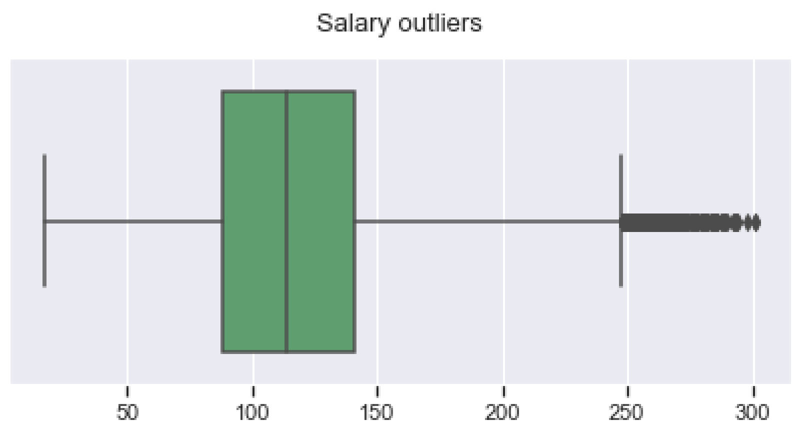
General statistics

At first sight, after computing the basic statistics of the dataset, we can make some initial conclusions:

- The numerical features (*yearsExperience* and *milesFromMetropolis*) both have a mean exactly in the middle of their ranges and a high STD value, which indicates that they aren't normally distributed. No outliers are detected, as we can see in the box plot.

- The target value *salary*, have a mean close to the 2nd quartile, and smaller, but still high STD. There are some high salaries, but salaries can be that high, considering the max value is 300, we consider those values are not outliers.

- Here, we can see the target value salary box plot:

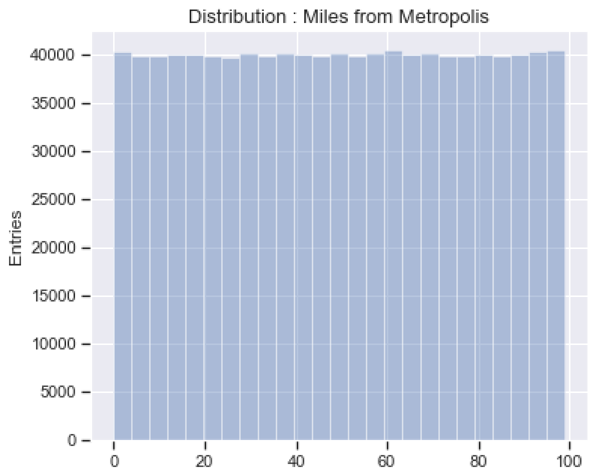


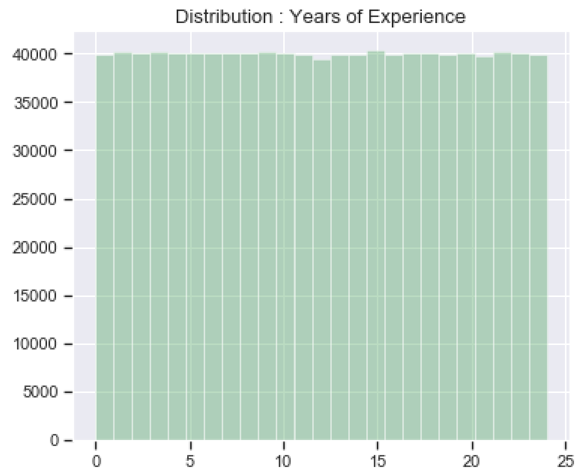
- The categorical features have the following unique values:

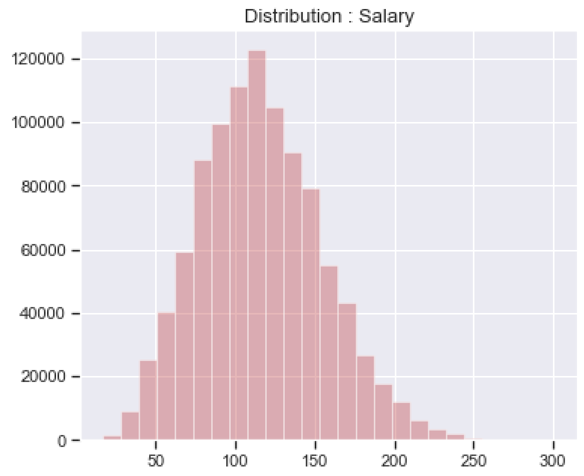
Feature	**Unique values**
jobType	CFO, CEO, VICE_PRESIDENT, MANAGER, JUNIOR, JANITOR, CTO, SENIOR
degree	MASTERS, HIGH_SCHOOL, DOCTORAL, BACHELORS, NONE
major	MATH, NONE, PHYSICS, CHEMISTRY, COMPSCI, BIOLOGY, LITERATURE, BUSINESS, ENGINEERING
industry	HEALTH, WEB, AUTO, FINANCE, EDUCATION, OIL, SERVICE

Feature distribution

- Numerical features (*yearsExperience* and *milesFromMetropolis*) are evenly distributed within their respective ranges, the salary feature has a normal distribution, slightly left squewed.



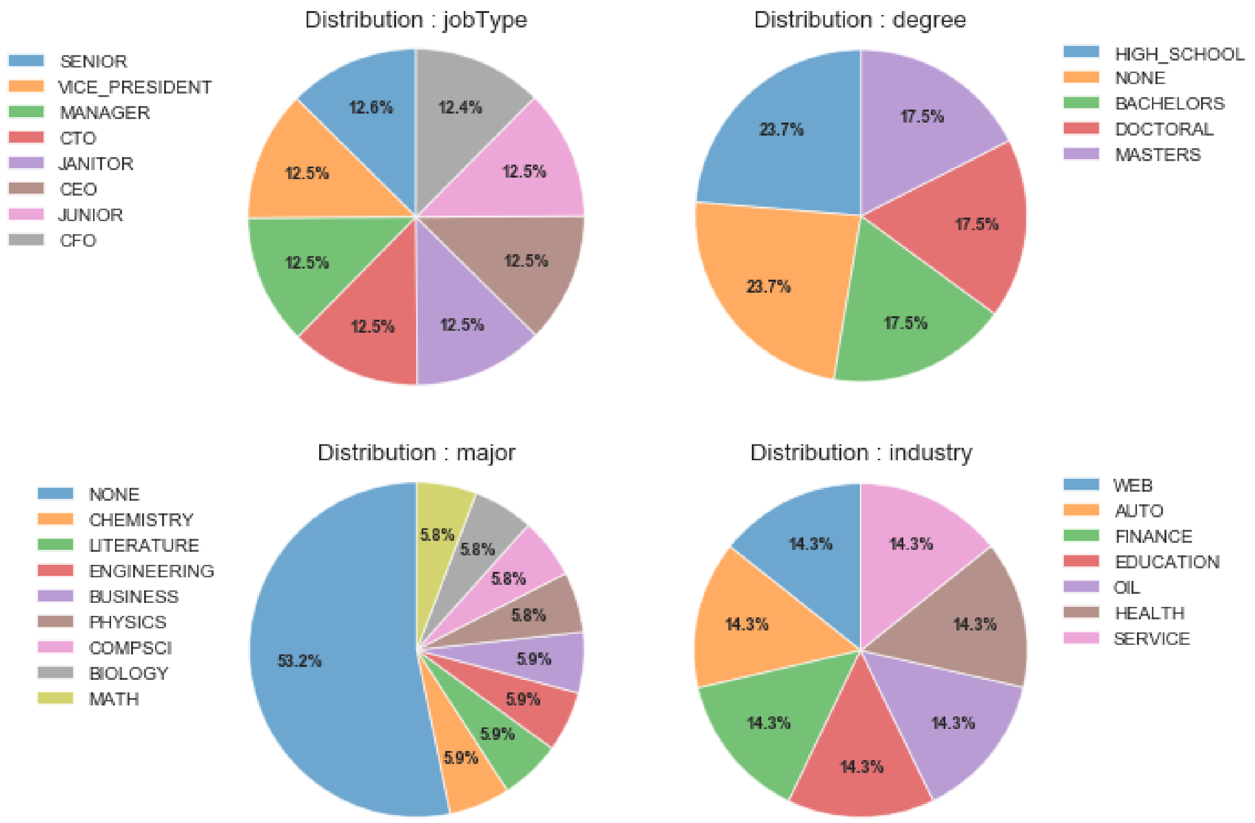




- 2 of the categorical features (*jobType* and *industry*) have a perfect evenly distribution, as each category has the same percentage of records.

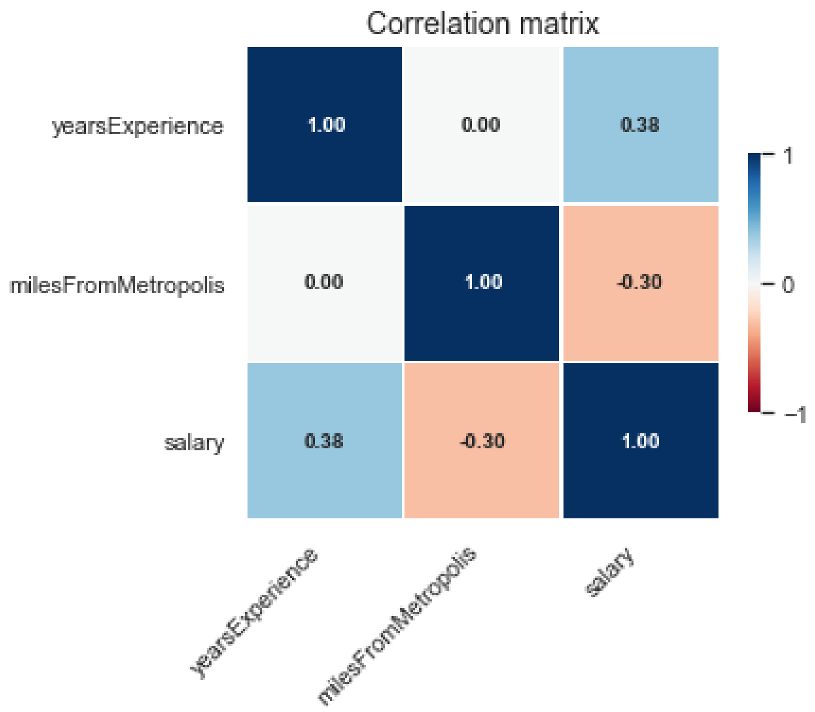
- Feature *degree* splits into 2 groups, one with **NONE** and **HIGH_SCHOOL** with 47.4% of the records; 23.7% each, and the remaining 52.6% is evenly distributed among the 3 remaining categories.

- Feature *major* has a predominant category, **NONE**, with 53.2% of the records, the remaining 46.8% is evenly distributed among the remaining categories.



Correlations

We conclude that there is no correlation in the numerical features, not with the target value (-0.38 / 0.30) and not between them (0.00).



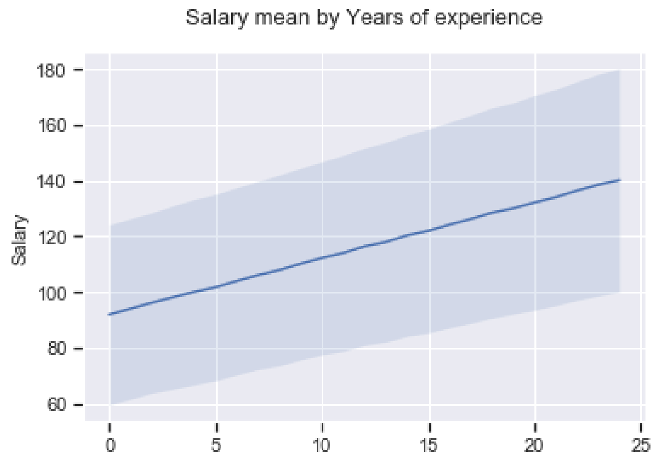
4. Exploratory data analysis

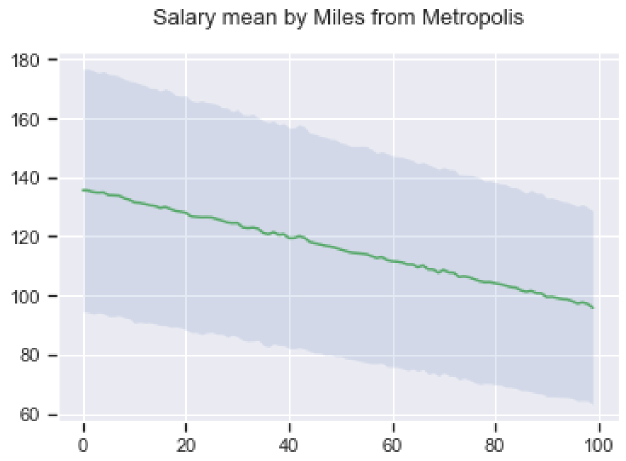
Features statistics analysis

The global mean salary is ***116.06***; if we calculate the mean per company, we see the range doesn't differ that much, as the max is ***116.79*** and the min is ***115.34***, with an inner range of only ***1.45***.

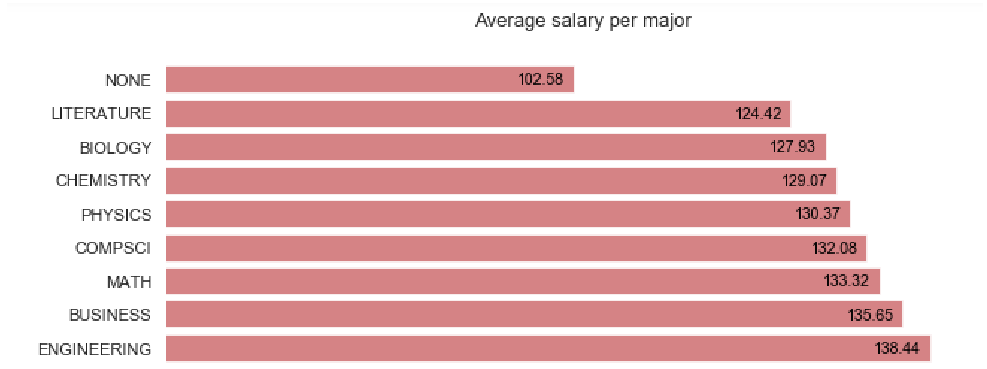
If we dig deep and calculate the mean salary per company and job position, the differences are minimal, the highest inner range is for position *CEO*(***4.38***) and the mean of the ranges is ***3.23***. These calculations indicate, that the average salary per company won't help with the prediction task.

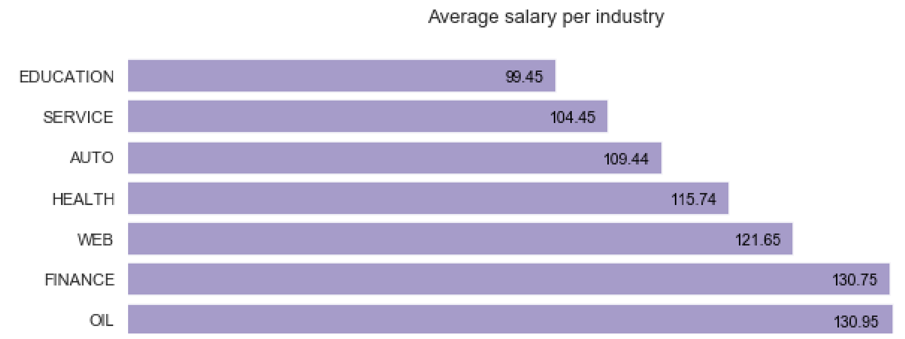
As expected, the mean salary increases together with the years of experience of the employee, but interestingly, decreases the far the employee lives from the metropolis.

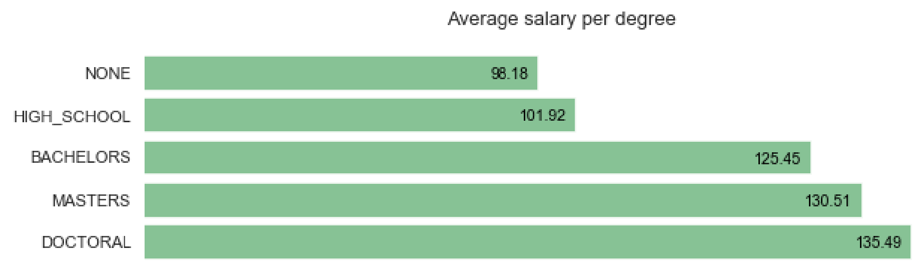


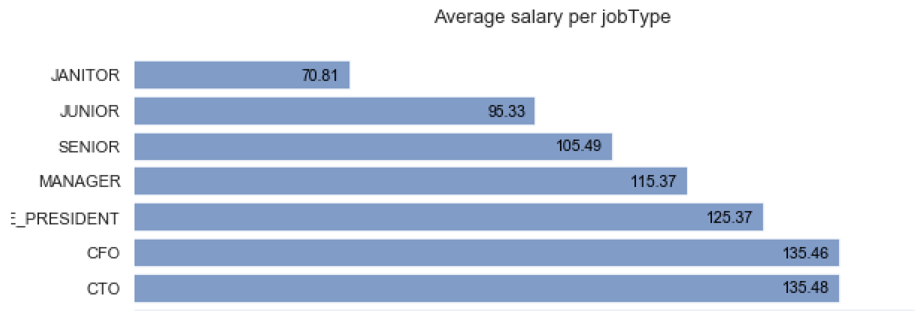


When we grouped the salary means by the unique values of the categorical features, we got the following insights:









•**jobType:** As expected, the lowest value is for *JANITOR* and the highest for *CEO*. Overall, the values gradually increase according to the level of the job, but interestingly enough, *CTO* and *CFO* have the same value.

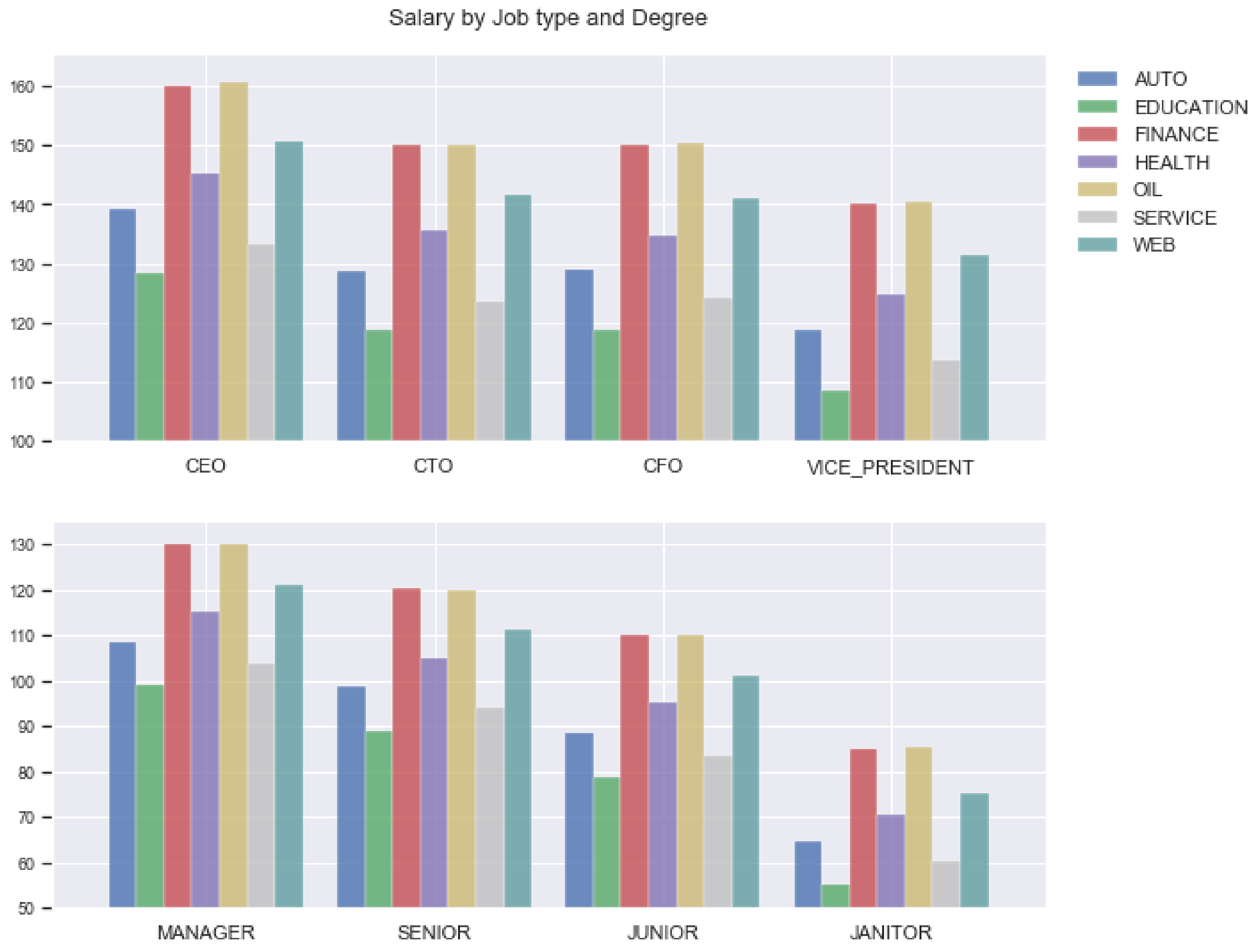
- **degree:** The values are split into 2 groups, *NONE* and *HIGH_SCHOOL* in the lower range, and *BACHELORS*, *MASTERS*, and *DOCTORAL* at the top. The gap is significant between *HIGH_SCHOOL* and *BACHELORS*.

- **major:** Having a major pays best than not having one; *NONE* has the lowest salary and is widely separated from the rest of the categories, the top major is *ENGINEERING*.

- **industry:** Perhaps the feature that shows a more gradual rise between categories. *EDUCATION* pays the least, *FINANCE* and *OIL* the most, and both have essentially the same salary mean.

As the position is a key feature, we've analyzed the average salary per *jobType* divided by the other 3 features. Interestingly, the differences remain stable in all cases, that is, they follow the same behavior when we divide them by *degree*, *industry* or *major*. They all descend in the same way, through the different job types, and only becoming pronounced when we reach *JANITOR*.

Here you can see the bar plot of the salary average for *degree* feature, where the trend is easily perceived. Other plots can be seen in the notebook.



5. Data pre-processing

Feature engineering

To enhance the performance of the models, 6 new features were created:

- **hasMajor:** As the distribution of this feature is 53% of NONE value, and the rest is equally distributed among the remaining values, a dummy fueature was created.
- **compID:** To be able to use the company ID in the machine learning algorithm, a new column was created with only the numerical code of the companies.
- **jobIndDegMajYex:** To take advantage of the information we already have, we calculated the mean salary values per ***job****Type*, ***ind****ustry*, ***deg****ree*, ***maj****or* and ***years*****Ex***perience* categories. 4 features based on that grouping were created:

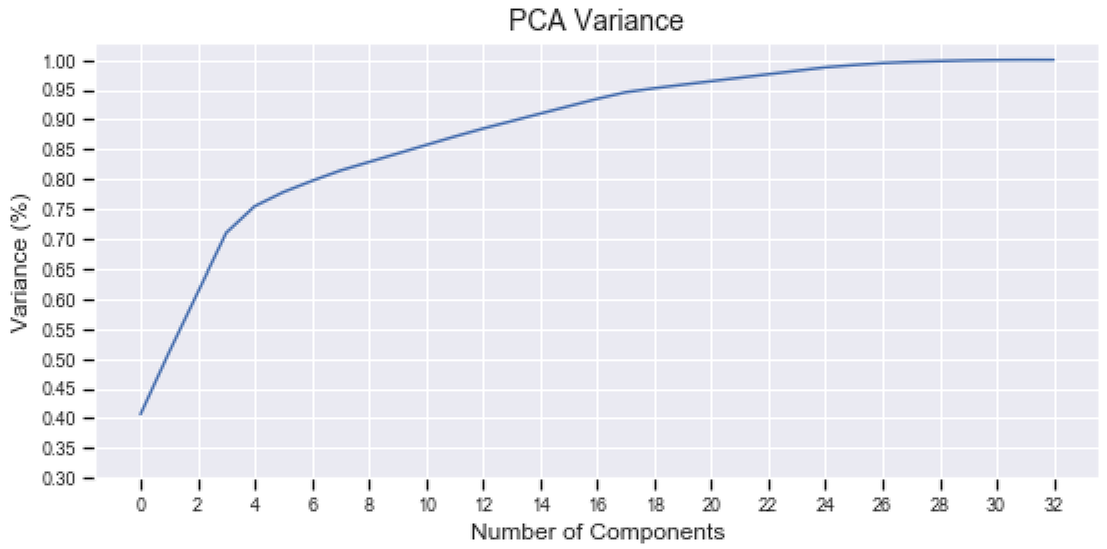
    - *jobIndDegMajYex_mean:* Salary mean of the group.
    - *jobIndDegMajYex_std:* Salary STD of the group.
    - *jobIndDegMajYex_min:* Min value of the Salary mean in the group.
    - *jobIndDegMajYex_max:* Max value of the Salary mean in the group.

We've deleted the column *companyId*, and by applying the Pandas function ***get_dummies***, we've converted the categorical features into numerical dummy variables.

Data normalization and PCA analysis

To most machine learning algorithms, the more uniform the data is, the better the results are. In this case, as we have different value ranges across the different features of the dataset, we've normalized the data with the ***Z Normalisation*** technique.

We've analyzed the train dataset, to see if we could reduce its dimensionality. Applying PCA, we've concluded that there's no gain creating a new PCA dataset. You can see the graph in the notebook, to capture the *95%* we need more than **26 features**.



6. Model development

Baseline

To begin with, we must define how to measure the effectiveness of the models we are going to develop. In this case, our evaluation measure will be the ***mean squared error (MSE)***.

Our baseline is the mean of the salary grouped by *jobType* and *companyId*. The mean *MSE* error obtained is ***963.4481***.

Best model selection

For this regression task, 3 different machine learning algorithms were selected, one linear and two ensembles, to see which performs better for the problem:

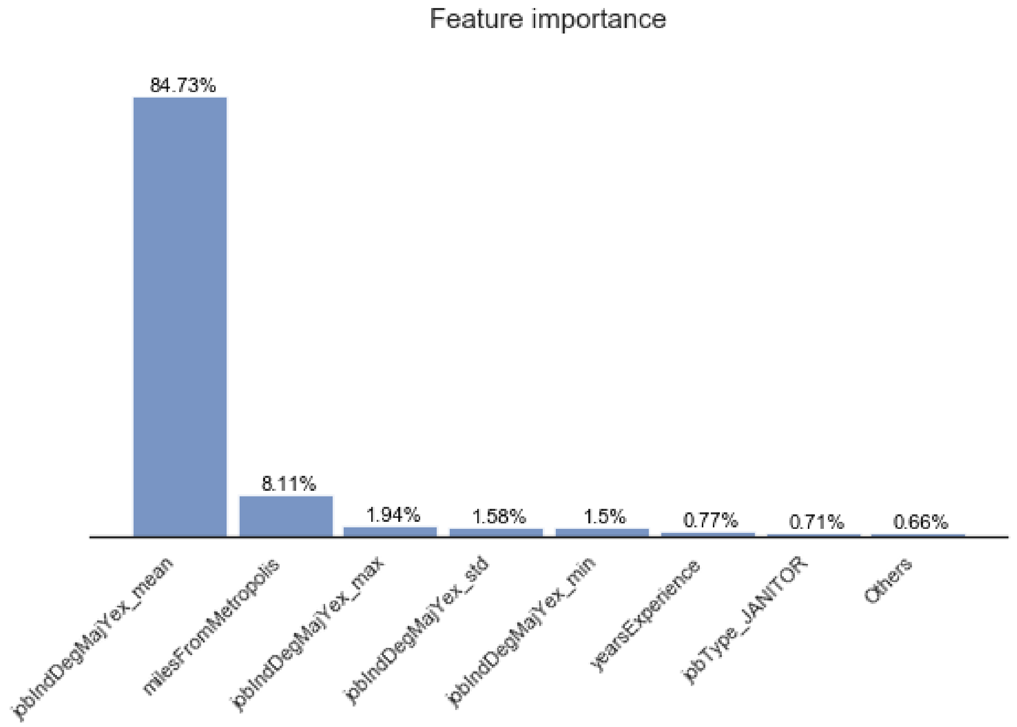
- **Linear regressor.**
- **Random forest regressor.**
- **XGboost regressor.**

After being carefully tuned, we've tested each of them with *cross_val_score*, having the following results:

Classifier	MSE	RMSE
Linear regressor	359,4400	18,9589
Random forest regressor	340,5944	18,4552
XGBoost regressor	339,3365	18,4211

Feature importance and predictions

The lowest *MSE* error was returned by the ***XGBoost regressor***, although the score of the *Random forest regressor* was very close. Once the best model was fitted with the train data, the feature importance generated by the model are:



Then, the prediction for the unseen jobs position was made and exported into a CSV file.

7. Conclusions

Conclusion

We can confirm that we've developed an accurate model to predict the salary based on the features given, an *MSE* error of **339.36** (*RMSE 18.42*) is good, but not enough accurate to be taken as a strong baseline, we consider that it needs further development, especially on the categorical features.

As a future development, we will handle the categorical features with *label encoder* instead of *Pandas get_dummy*, as the ensemble models aren't affected by this kind of encoding, and check its results to see if there's room for improvement.

Beyond that, the decision of calculating a narrower salary average, by grouping it by almost all categorical features has been a success. According to the feature importance of the model, the mean salary is ***THE*** feature that performs the prediction most by itself.

Final thoughts and recommendations

- As explained before, we consider that the model performs good, but can be improved. But is true that, perhaps with more information (features like projects, extra hours, people in charge, acknowledgments, etc.), the model would be more accurate.

- Bear in mind that, gather all that new data has a cost, the stakeholders should analyze if the investment is profitable if a more accurate model is developed. If the salary predictions will save money, time and enhance HR processes, is an option to take in deep consideration.