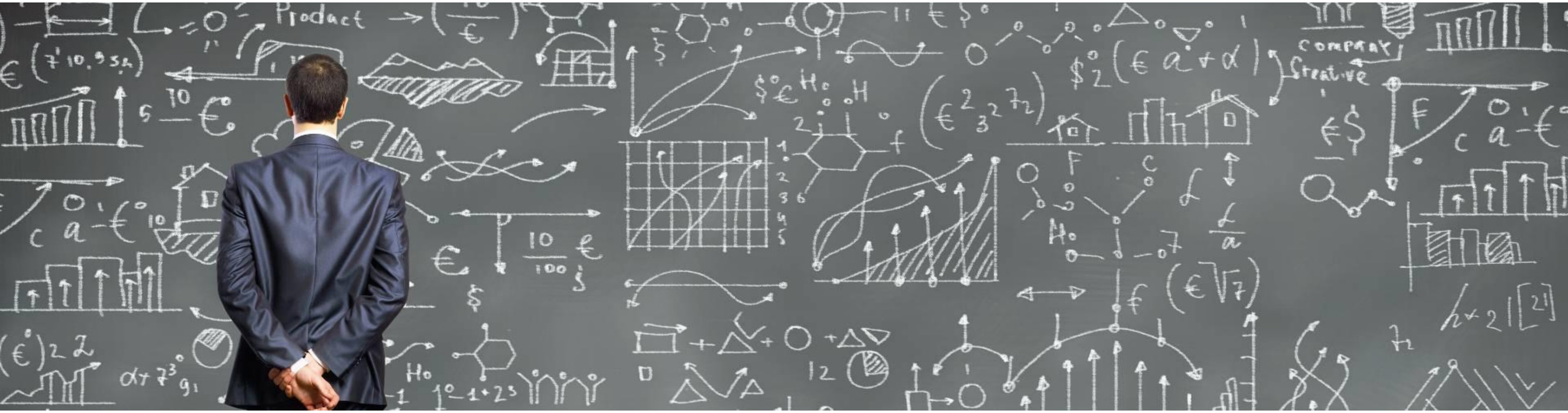
Ivana Hybenova



Salary Prediction

[12-04-2020]

Agenda

Salary Prediction

Background / Business Problem	3
Executive Summary / Key Takeaways	4
Analysis – Data Set Characteristics	5
Analysis – EDA	6
Analysis – Cleaning & Pre-processing	9
Analysis – Modelling, Tuning & Evaluation	10
Analysis – Key Results & Recommendation	11
Next Steps & Improvements	12
Appendix	13-16

Background / Business Problem

Situation

- As an HR company we want to provide a professional advice on salary for our customers. It has turned out to be a very valuable information for both the candidates, that are not sure how much money they should aim for in negotiation process and for companies that don't want to offer too low salary for an open position, which could discourage talented candidates, but on the other hand they don't want to overpay them, as unnecessarily high salaries mean smaller budget for other companies expenses.
- We estimate this salary as an average salary per given job type in given industry with given degree based on the data from the market we have.

Complication

- Besides the job type, degree and industry we have also information about how far the given job is from the metropolis and how many years of experience the cadindate has. We even have internal company id for each job. We hope that using a machine learning algorithm we could do the salary predictions based on these information, too.

Source: https://www.xyz.com
Reference: https://www.xyz.com

Executive Summary / Key Takeaways

Approach & Solution

- I examined the available data set and calculated the mean absolute error using the base model, which is almost 22 thousand dollars per year. I used the approach of calculating mean salary and other statistics per given job type, degree, industry and even major as a new predictor for the supervised machine learning model, which helped the model to see the patterns and relationships, and picked the models that can utilize these calculated statistics most and are robust to outliers.

- As everybody assumed distance form metropolis has negative impact on the salary and years of experience has positive impact. Combination of these numbers alongside with mean and median salary for each group was enough to build a model with mean absolute error only something over 15 thousands dollars a year. This model explains 75% of the salary and predictions are fully automated, so every night every new jobs in the database are scored.

- I did not use company id as a predictor, as we want a general model for any company and because the mean salaries of each company were not very different. But it is still possible that there are some differences from company to company, we just need to come up with a better approach how to differentiate the companies, to make the model general.

I believe that information like number of employees of the company and information whether it is a start-up, scale-up or an established company, and age would make the model even more powerful, as more stable and bigger companies are very likely to offer higher salaries.

Data Set Characteristics

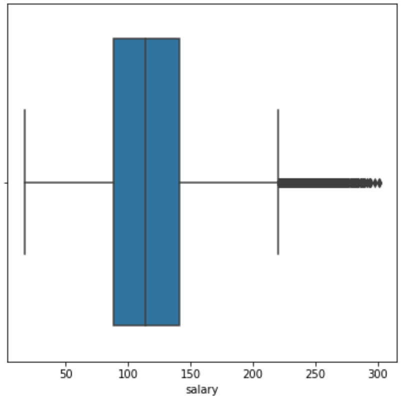
Dataset Information

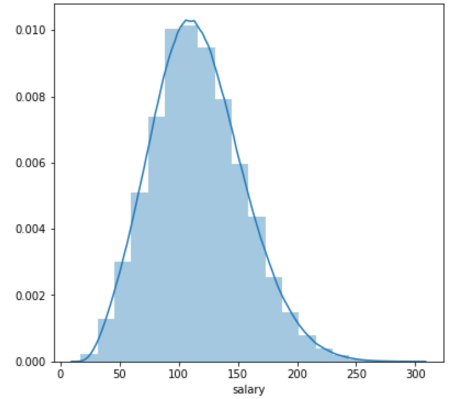
- The dataset has 1 000 000 unique jobs with 7 features (besides unique jobId) and the target variable:

  1. companyId – 36 unique combinations of word „JOB“ and numbers
  2. jobType – 8 categories, „JANITOR“, „JUNIOR“, „SENIOR“, „MANAGER“, „VICE_PRESIDENT“, „CFO“, „CTO“, „CEO“
  3. degree – 5 categories, „NONE“, „HIGH_SCHOOL“, „BACHELORS“, „MASTERS“ and „DOCTORAL“
  4. major – 9 categories, „NONE“, „LITERATURE“, „BIOLOGY“, „CHEMISTRY“, „PHYSICS“, „COMPSCI“, „MATH“, „BUSINESS“, „ENGINEERING“
  5. industry – 7 categories, „EDUCATION“, „SERVICE“, „AUTO“, „HEALTH“, „WEB“, „FINANCE“, „OIL“
  6. yearsExperience – with min value 0 and max value 24
  7. milesFromMetropolis – with min value 0 and max 99
  8. salary – after removing 5 zero values, min value is 17 and max is 301 thousands dollars per year

Dataset Visualizations

	jobId	companyId	jobType	degree	major	industry	yearsExperience	milesFromMetropolis	salary
0	JOB1362684407687	COMP37	CFO	MASTERS	MATH	HEALTH	10	83	130
1	JOB1362684407688	COMP19	CEO	HIGH_SCHOOL	NONE	WEB	3	73	101
2	JOB1362684407689	COMP52	VICE_PRESIDENT	DOCTORAL	PHYSICS	HEALTH	10	38	137
3	JOB1362684407690	COMP38	MANAGER	DOCTORAL	CHEMISTRY	AUTO	8	17	142
4	JOB1362684407691	COMP7	VICE_PRESIDENT	BACHELORS	PHYSICS	FINANCE	8	16	163

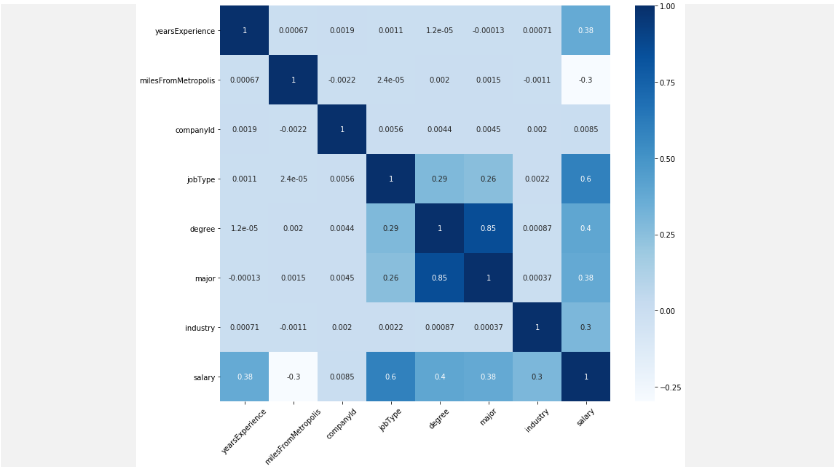




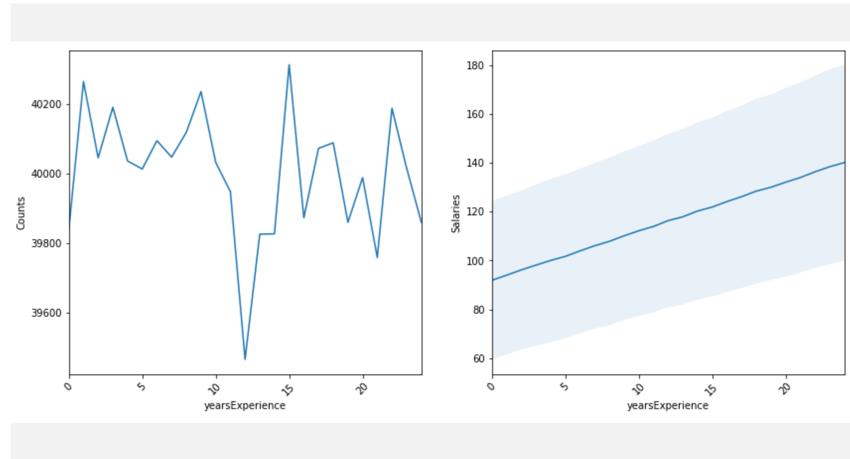
	yearsExperience	milesFromMetropolis	salary
count	999995.000000	999995.000000	999995.000000
mean	11.992407	49.529381	116.062398
std	7.212390	28.877721	38.717163
min	0.000000	0.000000	17.000000
25%	6.000000	25.000000	88.000000
50%	12.000000	50.000000	114.000000
75%	18.000000	75.000000	141.000000
max	24.000000	99.000000	301.000000

EDA – Exploratory Data Analysis

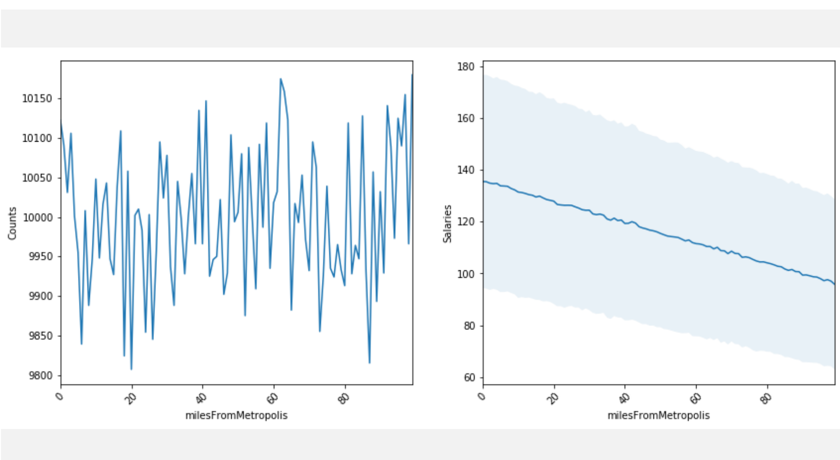
Correlation map



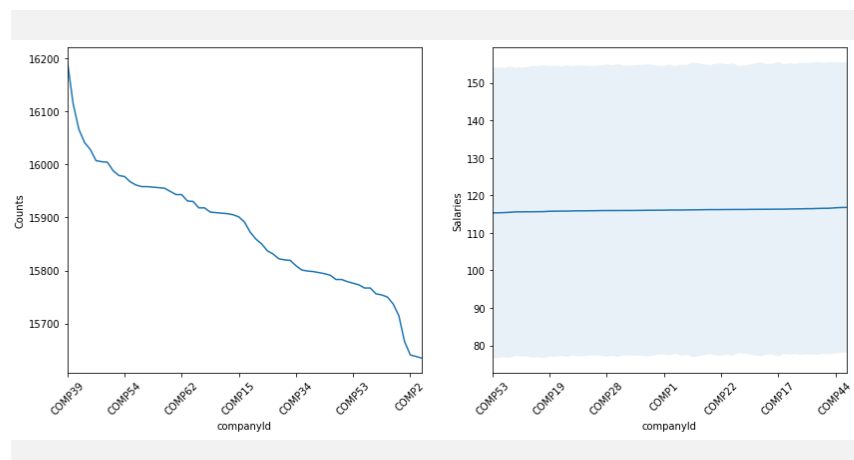
Years of Experience



Miles from Metropolis



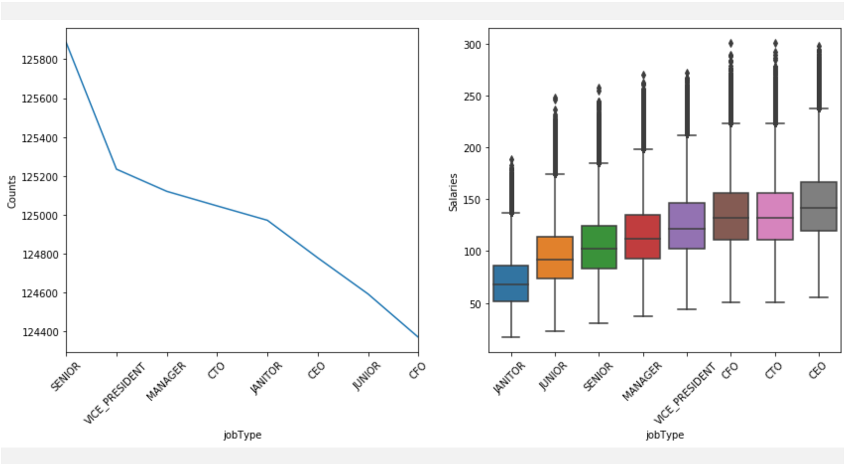
Company ID



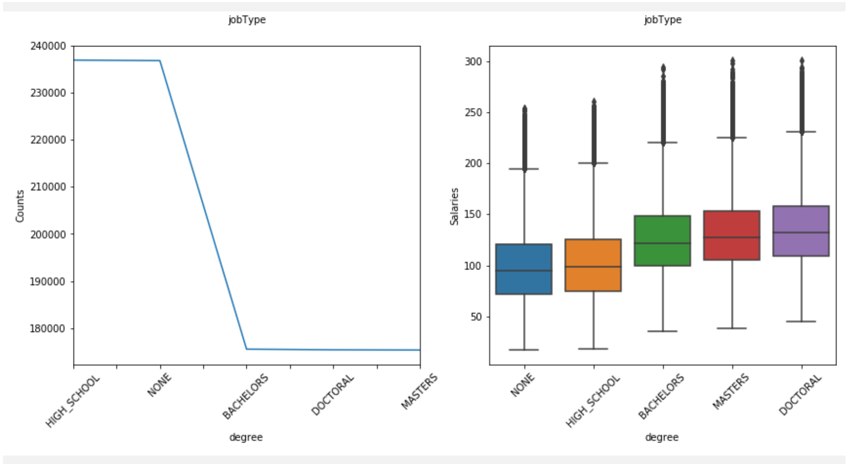
EDA – Exploratory Data Analysis

[Slide Tag line]

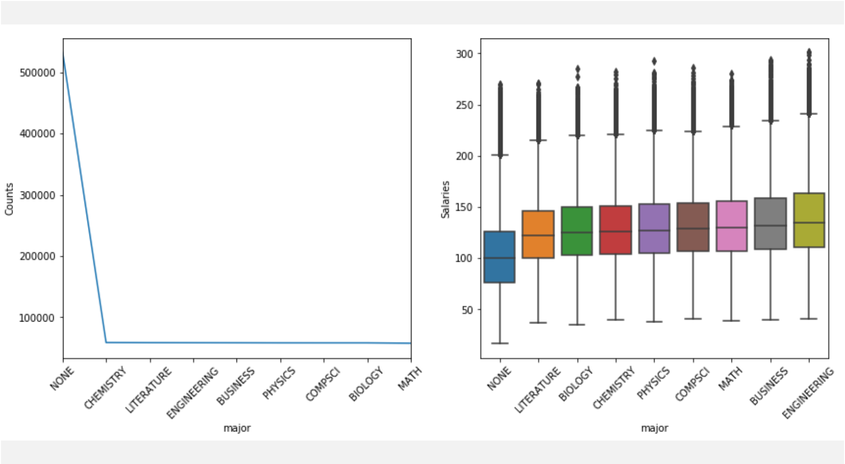
Job Type



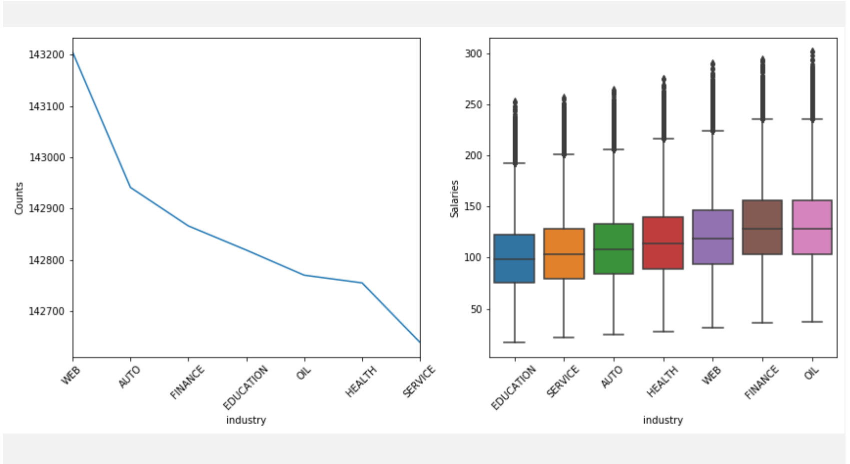
Degree



Major

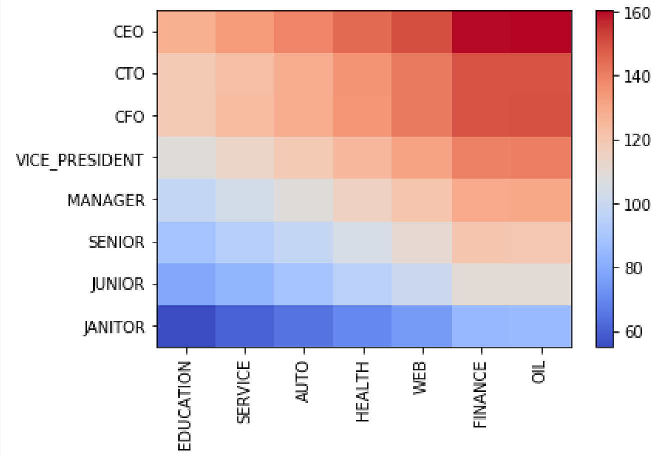


Industry

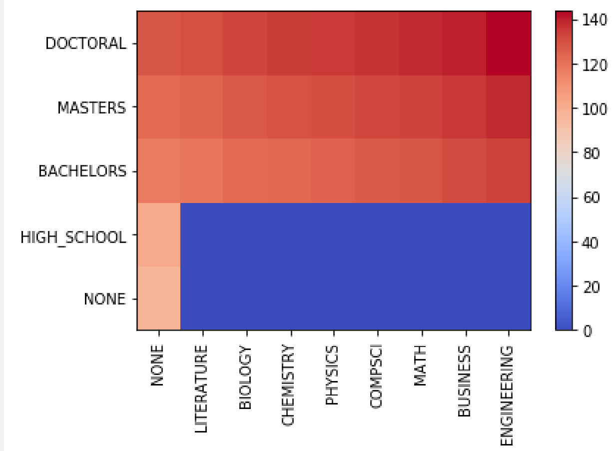


EDA – Exploratory Data Analysis

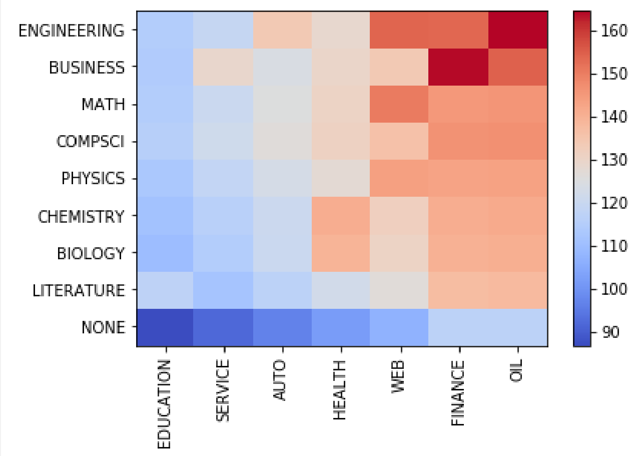
Job Type vs Industry



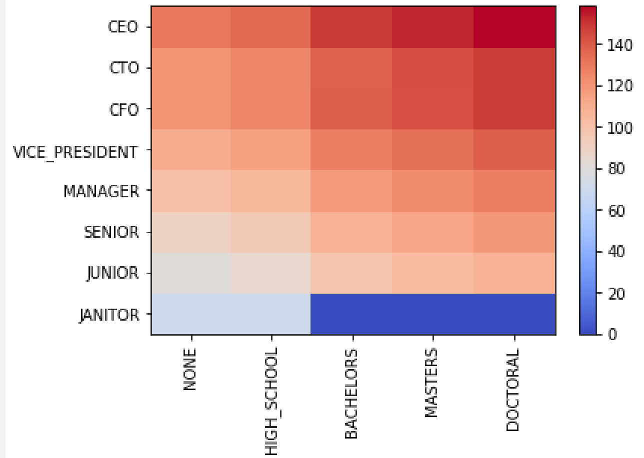
Degree vs Major



Major vs Industry



Job Type vs Degree



Data Cleansing & Pre-processing

Categorical Features

- Ordinal features: JobType and degree
  To prepare them for tree-based algorithms, the OrdinalLabelEncoder is used

- Nominal features: major and industry
  The Mean Encoding technique is used, so they are replaced with the average salary for given category

jobType	degree	major	industry
5.0	1.0	102.587732	121.719715
5.0	4.0	102.587732	130.629367
3.0	1.0	102.587732	130.629367
7.0	3.0	102.587732	104.417309
1.0	4.0	128.974052	99.431658

Numerical Features

- yearsExperience, milesFromMetropolis and salary

- 5 jobs with zero salary where removed from the dataset, the features where not standardized, as for tree-based algorithms it is not necessary

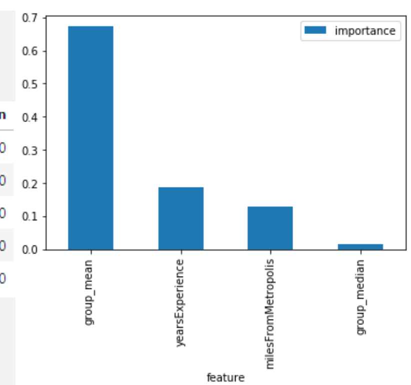
yearsExperience	milesFromMetropolis	salary
15	60	129
17	37	141
3	36	121
6	67	130
15	90	94

Feature Engineering / Dimension Reduction

- Tree based methods utilize features like statistics for each group. Company Id was dropped to make the model more general and statistics (min, max, mean, median, standard deviation) for given major, degree and industry where calculated.

- After training the algorithms, features to proceed with where selected based on feature importance of the best model, which was Gradient Boosted Trees.

group_mean	group_max	group_min	group_std	group_median
129.822748	223	70	27.529276	127.0
156.910072	244	95	29.972235	156.0
119.141943	213	62	26.990595	116.0
134.245614	202	77	26.309025	130.0
89.454874	152	48	22.955746	86.0



Modelling, Tuning & Evaluation

Model Selection

- Since I wanted to predict salary, which is a continuous variable, group of regression models were considered
- Based on EDA I could see that different combinations of categorical variables can leads to different mean salary, that is why I considered algorithms that can catch non-linear relationships, namely tree-based algorithms
- I evaluated Decision Trees, Random Forest and Gradient Boosted Trees (XGBoost)
- Decision Tree is simply put a group of rules that leads in regression case to mean salary, for example: If yearsExperience > 5, the next based on answer condition is evaluated etc.
- XGBoost builds a lot of such trees, where the next one tries to fix the error of those built before

Model Evaluation

- I first split the dataset to train and validation set, where validation set is represented by 30 % of the data.
- Each algorithm was evaluated based on average of mean squared errors. The cross validated score with 5 folds was used, which means that each model was trained 5 times on 4 different sub sets of train data and evaluated on the 5th one, that was not used to train the model.
- After selecting the best algorithm and the most important features, I tuned and validate it on the validation set to make sure, that it is not overfitting, e.g. that it is general enough, to be equally effective on data, that were not used to engineer the features and build the model.

Model Performance Results

MSE : 356

MAE : 15

R squared : 76 %

- Mean squared error was improved compared to the base model by 52 %.
- Mean absolute error was improved by 32 %.
- The delivered model can explain 76 % of variance in the salary, which is improvement by 25 % compared to the base model.

Analysis Results & Recommendations

Result #1

- The key predictor is mean salary for given degree, major, type of job and industry, the next one is number of years of experience, which is positively correlated with the target, e. g. the higher the number the higher the salary. With growing distance from metropolis on the other hand the salary is lower. The last predictor is median for given group.

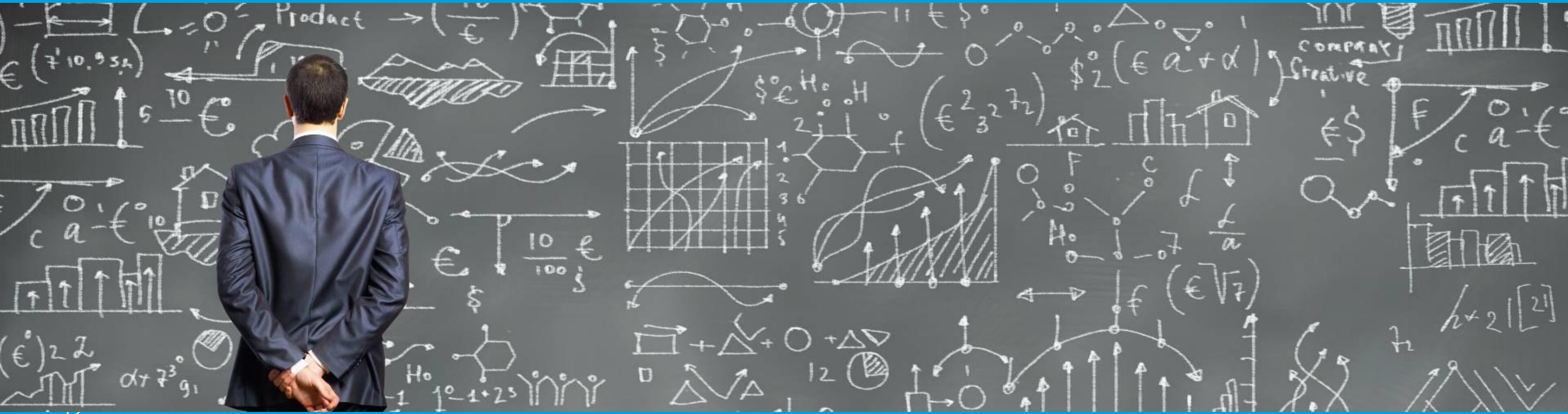
Result #2

- The built model has per job on average error of 15 000 dollars a year, and explains 75 % of variance of the salary. The model is in the deployment and automatically predicts salaries for new jobs in the company database.

Result #3

- Even the company id was not use as a predictor I highly recommend to try to repeat the modelling again with 3 extra features: number of employees, company type (start-up, scale-up or established) and number of years the company was on the market at the time the job was posted.

Appendix



Data Science Approach

1. Understand the problem	▪ Never forget which business problem you are trying to solve and the business objectives.
2. Explore the data	▪ Exploratory data analysis to understand the quality of the data (i.e. missing fields), the shape of the data (size, number of features, type of features), the statistic profile of the data (i.e. outliers, distribution etc.)
3. Cleanse the data	▪ Clean any data quality issues: garbage in, garbage out
4. Preprocess the data	▪ Transform the data or engineer new features if necessary to gain more insights
5. Metrics and Modeling	▪ Model creation, evaluation and selection
6. Evaluate findings	▪ Are they logical and do they make sense? Is the modeling approach used appropriate?
7. Iterate and Refine	▪ Refine analysis and fine tune models and findings
8. Communicate clearly	▪ Simple and straightforward messaging linking the results to the business outcome. ▪ Assumptions stated.

Code is clean, easy to read and the analysis is repeatable

Development Environment

- Python, R
- Libraries: Scikit-Learn, Pandas, SciPy

Code

python™ scikit learn SciPy

- Windows
- JupyterLab
- Spyder

Environment

Windows 10 lab JupyterLab Spyder

- Seaborn,
- Matplotlib

Visualization

Seaborn matplotlib

- CSV
- PostgreSQL Database

Data

PostgreSQL