CZECH TECHNICAL UNIVERSITY IN PRAGUE
FACULTY OF INFORMATION TECHNOLOGY

ASSIGNMENT OF BACHELOR'S THESIS

Title:	Job-board predictions
Student:	Martin Šmíd
Supervisor:	Ing. Pavel Kordík, Ph.D.
Study Programme:	Informatics
Study Branch:	Computer Science
Department:	Department of Theoretical Computer Science
Validity:	Until the end of summer semester 2017/18

Instructions

Explore the field of text mining methods and predictive modeling algorithms. Examine the dataset of job advertisements and the number of corresponding responses provided by the LMC company. Propose and verify diverse approaches to predict the number of job advertisement responses using text mining and predictive modeling methods. Evaluate the prediction success rate of algorithms and try to find a way to increase it utilizing the model certainty. Recommend and precisely document the best approach including the analysis of particular input attributes.

References

Will be provided by the supervisor.

L.S.

doc. Ing. Jan Janoušek, Ph.D.
Head of Department

prof. Ing. Pavel Tvrdík, CSc.
Dean

Prague November 12, 2016

Czech Technical University in Prague
Faculty of Information Technology
Department of Theoretical Computer Science

Bachelor's thesis

Job-board predictions

Martin Šmíd

Supervisor: Ing. Pavel Kordík, Ph.D.

16th May 2017

Acknowledgements

I would like to thank everyone who supported me during my studies, specifically my family and friends. It would be very difficult for me without any of them. I would also like to thank my supervisor for his guidance and patient help.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as school work under the provisions of Article 60(1) of the Act.

In Prague on 16th May 2017 .....

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis

Šmíd, Martin. *Job-board predictions.* Bachelor's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2017.

Abstrakt

Tato práce se zabývá metodami prediktivního modelování a vytěžování znalostí z textu. Na základě dat o pracovních inzerátech je cílem predikovat počet odpovědí na inzerát. Jednotlivé atributy jsou prozkoumány a zajímavé vztahy v datech prezentovány. Následuje návrh prediktivních modelů, jejich vytvoření a porovnání. Na základě jistoty modelu je vylepšena jejich úspěšnost. Nakonec je diskutována důležitost jednotlivých proměnných pro nejlepší model.

Klíčová slova prediktivní modelování, analýza dat, vytěžování znalostí z dat, vytěžování znalostí z textu, implementace, personalistika, H2O, R

Abstract

This thesis looks into the area of predictive modeling and text mining methods. Based on the job advertisements data, the goal is to predict number of responses to the job offer. Individual attributes are examined and interesting data relations are shown. Subsequently, predictive models are proposed, built, and compared. By using model confidence, the model performance is tuned. Lastly, variable importances for the best model are discussed.

Keywords predictive modeling, data analysis, data mining, text mining, implementation, HR management, H2O, R

Contents

List of Figures

List of Tables

Introduction

Evolution

Data mining and predicting future behavior of people or systems has been a raising trend recently that is spreading more and more throughout large companies, smaller bussiness, advisory, academic sphere or public service. It does not necessarily affect only IT domain because it is a topic widely popular amongst many diverse groups of people—both specialists and amateurs—business persons, futurists, psychologists, sociologists, philosophers to name but a few.

Specific results of machine learning and related disciplines suggest that these technologies will be dynamic, fast-developping, and impacting broad variety of fields in near future. Twenty years ago, IBM's supercomputer defeated contemporary World Chess Champion Garry Kasparov [1]. Last year, Google's artificial intelligence managed to succeed in much more complicated board game of go, defeating Lee Sedol who is considered one of the best present-day players [2]. Other significant advances in methods of gathering information from data can be found pretty much everywhere, notable mentions are automatic fraud detections in banking and insurance industry or refinement of natural language processing and speech recognition used by cell phone assistants Siri, Google Now, and Cortana.

Quality of advertising is a crucial factor for employers and their HR departments not only for economic reasons but also because of unemployment rate in the Czech Republic reaching only 5 % in October 2016, which was the lowest from the time of economic crisis [3]. Labour demand vastly surpasses the supply, meaning that HR officers must come up with new ways to attract attention of potential employees.

Motivation

I chose this topic beacuse many people may not be satisfied with their jobs. We usually invest majority of our time in the working life and some of us can be annoyed, nervous, even stressed by it for no reason. Improving the adverisement quality could thus help many to do what they wish for and what they enjoy.

Besides, I am generally interested in the topic of artificial intelligence and related fields as it is a relatively young, extensive, and perspective area with immense possibilities and, possibly, bright future. Moreover, it combines theoretical sphere of inventing new algorithms and their enhancement with practical impact that has real, measurable outcomes that could change our perception of the world and help us better understand the individual links and principles constructing the whole universe.

Goal of this work

The aim of this project is to help employers with advertisement creation, job portals with their advertisement, and thus future employees with finding the job they like. Based on the created models and their evaluation, the human resources officer should be able to consider whether it is probable that his advertisement will be answered by a suitable number of candidates, taking into consideration the importances of individual aspects of his job advertisement.

The goal of the research part of the thesis is to familiarize with the basic predictive models, with their evaluation, and with basic methods of text mining.

The objective of the practical part of this work is to propose and verify multiple methods of predicting the number of answers to the job advert. Another goal is to evaluate the success rate and try to ameliorate this prediction using the model certainty (model confidence). Last but not least, one of the main objectives is to evaluate the importance of particular input attributes and find out some interesting relations in the data.

Basic terminology

Here is a brief overview of the basic terminology.

Label Output variable, the variable we are trying to predict

Attribute Input or output variable

Instance, record Set of data corresponding to one occurence of the observed data; one row in a dataset

Dataset Assemblage of data, usually in a table form

Data mining Extracting knowledge from data using various processes and algorithms

Text mining Extracting knowledge from text using various processes and algorithms

Model confidence Number supplied by the model that expresses the extent of probability that the prediction is accurate

All important terms and expressions are explained in the Appendix A.

Chapter 1

State-of-the-art

The following sections briefly describe current progress in the field and mention the available tools for data mining and text mining.

1.1 Google's Cloud Jobs API

Cloud Jobs API was presented 15th Novermber, 2016. It is a platform that uses machine learning to recommend the suitable work position and job applicant, taking into consideration collected data (candidate's experience, his actions on the job portal etc.). However, its primary goal is not to evaluate the job ad quality.

The API incorporates Dynamic Recommendation Engine (API recommends based on live user rating), Real Time Query Broadening (low count of search results causes broader search range, e.g. increase in the location distance that it searches through) or Synonym and Acronym Expansion (searching company-specific or branch-specific jargon gives results for the similar or same expression) [4].

The Cloud Jobs API is built upon two ontologies: occupation ontology and skill ontology. Both are complicated multi-level structures that represent the individual relations in the area of work position or skills. The machine learning is used to detect skills and occupations in job seeker's queries or when the job titles are mapped to nodes in the skill ontology. This is done by using multiple techniques like denoising (removing unrelated words in the title), word to vector representations, and classification with Neural Based Ensemble Classifier.

The Cloud Jobs API is designated for the companies that participate in the HR management. Companies like Dice, Jibe or CareerBuilder use the API in its current alpha version [5].

1.2 Available tools for data mining and text mining

Tools for extracting knowledge can be divided in two parts: Proprietary and Open-source.

1.2.1 Proprietary tools

Most common proprietary software for data mining includes:

- *SQL Server Analysis Services*
- *SAS Enterprise Miner*
- *IBM DB2 Intelligent Miner*

Amongst the proprietary tools for text mining, the most popular are:

- *SAS Text Miner*
- *IBM SPSS Text Analytics for Surveys*
- *Cogito* by *Expert System*

These products are primarily focused on the corporate clients.

1.2.2 Open-source tools

The open-source tools for data and text mining include:

- *RapidMiner*
- *WEKA*
- *R*
- *NLTK*
- *H2O*

All of these tools are freely distributed. They are usually managed by companies that provide customer support and create charged customizations to enhance the performance.

R is a very powerful tool due to its relative simplicity to use and thanks to many libraries that extend its functionalities. Many algorithms are implemented—Decision Trees, Random Forest, k-means clustering to name but a few. Regarding text mining, R offers libraries such as ”tm” or ”tidytext”.

Chapter 2

Theoretical background

This chapter summarizes the process of creating predictive models. Data preprocessing including analysis, model evaluation and improvements are mentioned alongside with some chosen classification and text mining methods and algorithms.

2.1 Data preprocessing

Before proceeding to suggesting how the models will be built, the initial data often needs to be filtered and cleansed.

2.1.1 Feature selection

Feature selection is a summarizing term for the choice of input variables for a model. Different methods can be used, 3 main categories are Filters (selecting variables independently of the chosen predictor, e.g. based on the correlations between them—the lower the better), Wrappers (selecting a subset of variables based on their predictive power), and Embedded (these methods are usually part of the training process of the model) [6].

Andrews plot is a way to visualize multidimensional data using a finite Fourier series. It gives insight whether or not there is any structure in the data, similarly as parallel coordinates graph. [7] This can be helpful when deciding whether or not the data is structured.

Other possibility to provide models with interesting data is trying to extract it from existing variable(s).

2.1.2 Binning

If it is needed to deal with categorical variables, they often need to be binned. The number of bins usually depends on the fact why we need to bin the variable and what values make logical sense. The bin boundaries can be

set either arbitrarily (supervised binning) or they can reflect equal-width or equal-height discretization. Equal-width discretization sets the bin boundaries in a way that created intervals are of the same length whereas equal-height discretization takes into consideration the number of instances in each bin. The latter method is usually better in most cases.

2.2 Text mining

Text mining spans a wide range of areas from predicting the next character or word while typing to the language style analysis. There are many different methods and algorithms, each of them suits another use.

2.2.1 Bag of words

Bag of words is a model that represents a text as a bag (or multiset) of individual words. This results in the loss of information that grammar encodes in the text.

Some more sophisticated methods come from this model, for example tf-idf with its use of term frequency.

2.2.2 Skip-gram

N-gram is a subsequence of the text that consists of N words.

K-skip-n-gram is an n-gram such that at most k words can be skipped in total in the original sequence. [8]

H2O uses this textual representation in its Word2Vec model. The neural network is trained to create the vector representation using various values of k and n.

2.2.3 Tf-idf

Tf-idf is a method that is able to quantify word importance in a document. It uses two measures—term frequency and inverse document frequency. Term frequency reflects the number of occurences of a word in a document. Inverse document frequency is an inverse function of the number of documents it occurs in. This means that the words that are too common or too rare get low rating and the words that are typical for given document rank higher. [9]

2.3 Classification models

Many algorithms were designed to predict a certain category. Some important and interesting ones are presented in the following part.

2.3.1 Deep Learning

Deep Learning is a multi-layer artificial neural network trained using back-propagation. It has a wide area of application. It can be used to predict a categorical or numerical label.

2.3.2 Ensembles

Ensemble methods were designed to attain a better prediction. They usually consist of more instances of simple models (such as decision trees) and the overall result is based on all of them.

2.3.2.1 Bagging

Bagging is an ensemble method that creates several simple models independently and the prediction is made as average of their votes.

Distributed Random Forest is a Bagging ensemble method. It independently builds a forest of classification or regression trees, based on type of the label.

2.3.2.2 Boosting

Boosting is an ensemble method that creates several simple models while each one new built tries to lower the overall error, meaning that newly created models are dependent on the ones built before.

Gradient Boosting Machine is a Boosting ensemble method. It sequentially builds classification or regression trees, based on type of the label. [10]

2.4 Model evaluation and improvement

The model quality needs to be evaluated and measured so that the models can be compared.

2.4.1 Confusion matrix

Confusion matrix is a neat representation of the model performance. It is a matrix composed of counts that denote what class was predicted and what class should have been predicted for all the possible combinations. Therefore, on the main diagonal there are the classification hits and all other spots represent classification misses. Some important values (e.g. error rate) can be calculated from this data.

2.4.2 AUC

AUC, area under curve, is the amount of surface that spans under the ROC curve. ROC (Receiver Operating Characteristic) curve is a graph depicting the

relation between sensitivity (true positive rate) and specificity (true negative rate) [11]. This value is between 0 and 1 and represents the model quality, the higher the better. The value itself basically respresents the probability of correctly predicting the label.

Chapter 3

Realisation

This chapter describes what I have done, from preprocessing the data and analysis to models suggestion, creation and evaluation.

3.1 Software

The essential used software is listed below.

3.1.1 H2O

H2O is an open-source software used to analyze data. Different REST API clients exist for H2O, the most notable ones are JavaScript, R, Python, and Flow. Flow is a notebook style web user interface. It is well arranged and intuitive so that even people who have no experience with programming can use it easily.

I downloaded, unzipped and used locally H2O version 3.10.4.3. When launching, 3 GB of memory were provided so that H2O can run smoothly.

H2O offers many models, for example Deep Learning, Word2Vec, GBM, and DRF.

To analyze the data and create models, I used the R API with H2O.

3.1.2 R

R is a free software environment for statistical computing and graphics.

I installed R in version 3.3.3.

RStudio, which is an integrated development environment for programming in R, was used in version 1.0.136.

R library called ”h2o” provides interface with the H2O. Other important libraries used in this work are ”tidytext”, ”dplyr” and ”wordcloud”.

3.2 Data preprocessing

The following paragraphs are devoted to the specific steps of data preprocessing.

3.2.1 Data characteristics

Details about the data are broken down in next lines.

3.2.1.1 Attributes

I was provided a dataset consisting of 14 variables, that is thirteen variables and the label (number of responses to the job ad). Namely, these are:

- Job ad description
- Company name
- Time when the job ad was published
- Time when the job ad was withdrawn
- Minimum proposed salary
- Maximum proposed salary
- Latitude of the job
- Longitude of the job
- Job ad title
- List of branches
- List of professions
- Amount of people that bookmarked the ad
- Amount of people that viewed the details about the ad

3.2.1.2 Dataset

Primarily, I used a manually created dataset sample consisting of 727 records (further on called 700 dataset). It was made as a subset of the original data where the job description is written in English. This is to ensure that the text mining algorithms—that do not have to be necessarily language independent regarding mixture of languages—do process data that is consistent.

Secondly, for some analyses, I used all the data consisting of 220 433 records.

If not stated otherwise, the dataset consisting of 700 records is primarily used to visualize data relations and to build models.

3.2.1.3 Attributes in detail

Detailed analysis of the attributes follows.

3.2.1.3.1 Responses—label variable This is the variable to be predicted. In the dataset and source code, it is usually called "cresponses".

Based on the histogram (700 dataset) 3.1, we can observe that the most common number of responses is zero and much less of them are connected with a too high amount of responses. The barplot 3.2 shows this specifically. Both these plots suggest geometric distribution.

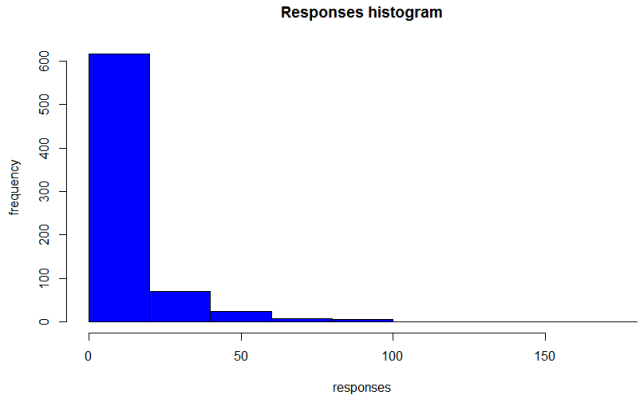

Figure 3.1: Responses histogram.

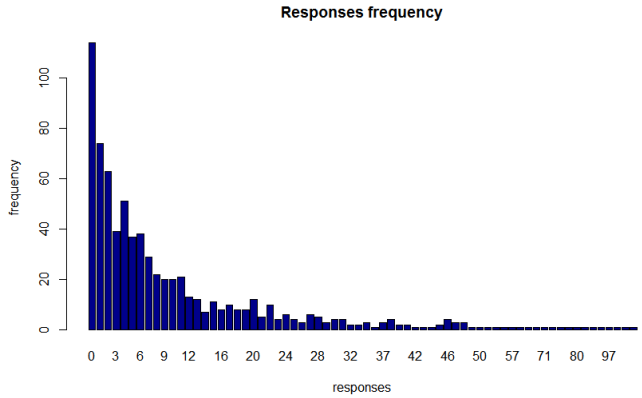

Figure 3.2: Responses barplot.

As one of the main goals of this work is to predict this variable, it can be deduced from the mentioned charts that predicting exact value of responses

would be quite hard. Thus, considering the purpose, the effort from further on is to predict the interval of responses. Therefore, the goal is to classify the advert into one of the suitable classes. This is discussed in the section 3.2.1.4 in detail.

3.2.1.3.2 Location Two variables in the dataset represent location: latitude ("clatitude") and longitude ("clongitude").

Graph 3.3 depicts density of location demands in the job advertisements where it is not stated as zero (of course these locations do exist, but based on the frequency of (0,0) coordinates in the dataset (9 042 instances when considering the whole dataset), it is safe to conclude that with high probability there is no job offer in the Gulf of Guinea). It is worth mentioning the plot silhouette roughly resembles the Czech Republic. The two visibly outstanding points are Prague and Brno. Plot 3.4 shows the locations with all the data. In comparison, the all data dataset contains more job offers from foreign countries, mainly Slovakia.

Figure 3.5 visualizes the job advertisements in Prague. The next image 3.6 shows the same information with all the data at disposal [12] [13].

Figure 3.7 shows the number of responses based on location, categorized in two bins.

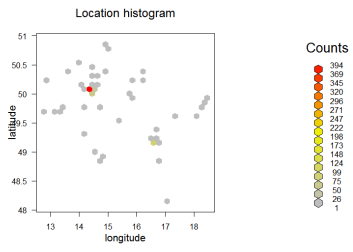

Figure 3.3: Location of advertisements and its density.

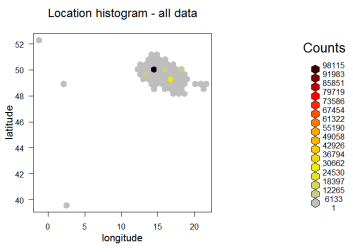

Figure 3.4: Location of advertisements and its density—all data.

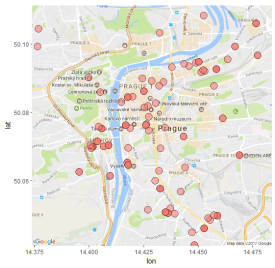

Figure 3.5: Job advertisements in Prague.

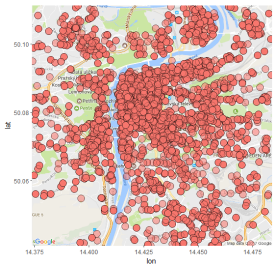

Figure 3.6: Job advertisements in Prague—all data.

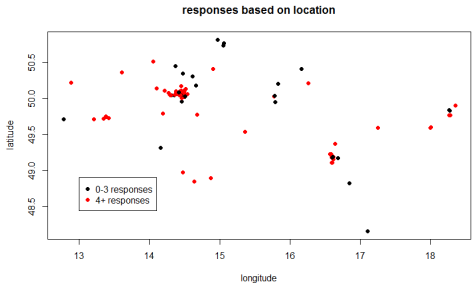

Figure 3.7: Number of responses based on location.

3.2.1.3.3 Salary Comparison of whether or not salary values ("csalarymin", "csalarymax") are stated (more precisely, whether or not they are stated as zero) in 700 dataset is displayed in figures 3.8 and 3.9. The plot also strongly indicates the linear dependence of the two variables. This is discussed in section 3.2.2.2.

Figure 3.8: Salary and zero responses.

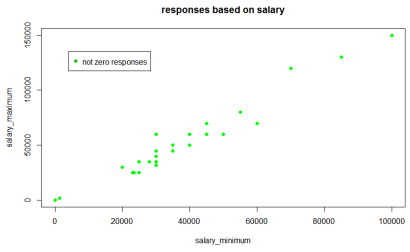

Figure 3.9: Salary and at least one response.

Figures 3.10 and 3.11 show the density of offered salary values and the connection between minimum and maximum salary values on the whole dataset, the first one with outlying values and the second one within a zoomed scale. Observable correlation is discussed in section 3.2.2.2.

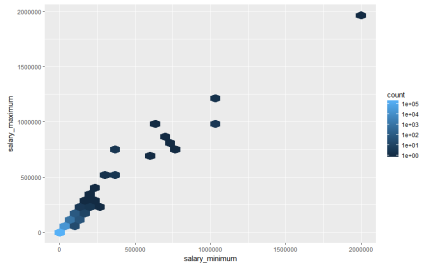

Figure 3.10: Salary histogram, all data.

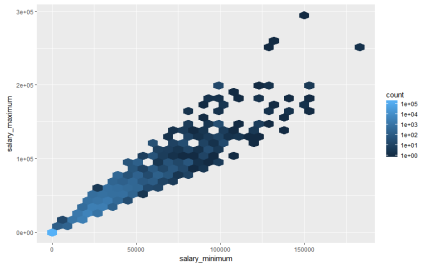

Figure 3.11: Salary histogram, without extreme values, all data.

3.2.1.3.4 Publication dates Publication dates ("cvalidfrom", "cvalidto") are represented as integer values that symbolize the unix time (in milliseconds). The validfrom attribute plotted in figure 3.12 is almost fully comprised of unique values. However, this could be helpful because this variable can provide the model with comparison which ad was submitted earlier and that could possibly have an influence on the label.

The validto attribute is, on the other hand, much less varied—see figure 3.13. The two values together imply for how long the advertisement was published and that could be a crucial information for the models, at least common sense tells so.

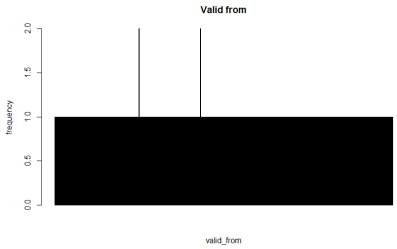

Figure 3.12: Valid from date barplot.

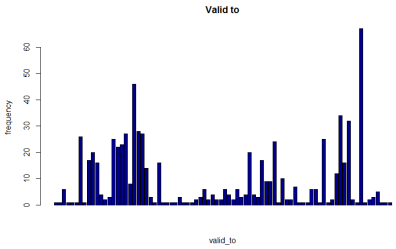

Figure 3.13: Valid to date barplot.

3.2.1.3.5 Description The job description ("cdescription") varies for each different job position. While it could be the same in some cases—e.g. if the same position is advertised again, but this would be obvious from the other

attributes—there is no need to treat this attribute differently than a text. In the all dataset, four languages seem to occur most frequently: Czech, English, Slovak and German.

The most used words in the ad description can be read from wordclouds 3.14 and 3.15. It is worth noting that the second wordcloud lacks the word "experience" because it could not fit in.

Figure 3.14: Description wordcloud.

Figure 3.15: Decription wordcloud, words with at least 4 characters.

3.2.1.3.6 Company The company name ("cname") is a variable consisting of characters, but according to the barplot 3.16, it is better to treat it as a category variable (enum).

The most common companies in the 700 dataset are: "Hays Czech Republic, s.r.o." (32 ads), "CPL Jobs s.r.o." (29), "Accenture" (28), "Honeywell, spol. s r.o." (22) and "ManpowerGroup s.r.o." (22).

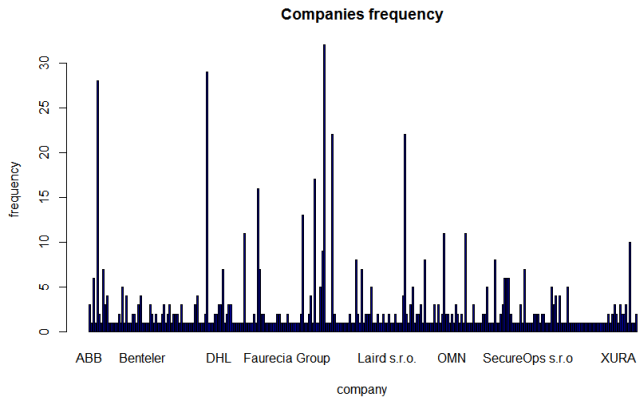

Figure 3.16: Companies barplot.

3.2.1.3.7 Title The job ad title (”ctitle”) is a variable consisting of characters. Because there are 719 different titles (one title is used 3 times and five are used twice), the variable should be treated as a text. However, because the individual languages are mixed (mainly Czech and English) even in the 700 dataset that has its description only in English, this variable is not used for text mining. The top words are in the wordcloud 3.17.

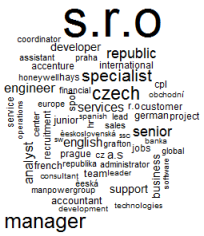

Figure 3.17: Advertisement title wordcloud.

3.2.1.3.8 Branches The branches assigned to the job ad (”cbranches”) are also represented as a text. In the 700 dataset, there are 249 distinct branches (or their mixtures). From the top five that occur the most frequently, four are a mixture of IS/IT (application and system development, consulting, system and HW, and so on) and one is Podnikové finance (Czech word for Corporate Finance). These categories occur sometimes in English and sometimes in Czech. However, because all branches from the top five occur more than 20 times, this variable will be treated as a category variable (enum). The top words are captured in the wordcloud 3.18. The number of appeareances of 10 most common branches is plotted in figure 3.19 without the names (they are summarized in the text above) because the names would not fit in the image.

Figure 3.18: Branches wordcloud.

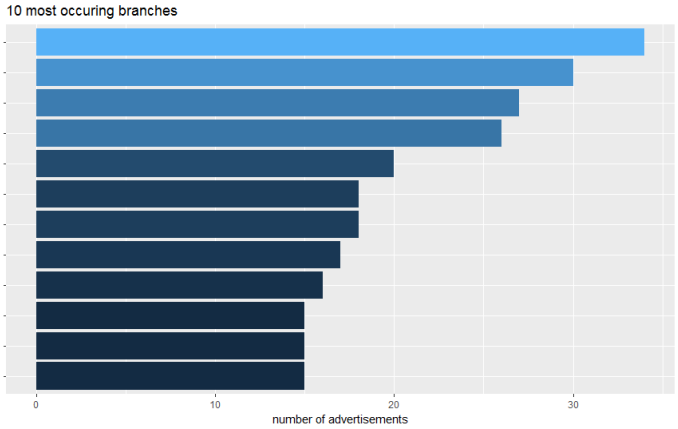

Figure 3.19: Top branches.

3.2.1.3.9 Professions The professions assigned to the job ad ("cprofessions") are also represented as a text. In the 700 dataset, there are 336 distinct professions (mixed in Czech and English). Most frequently occuring are Project Manager (15), Programmer (14), Administration (11), Tester (9) and Programátor (8, Czech word for Programmer). This input variable seems to represent few information, but in combination with other attributes it could bring some additional value to the models because the diversity may be the key to predict the label. The top words mentioned in professions are captured in the wordcloud 3.20. The top five professions are visualized in 3.21.

Figure 3.20: Professions wordcloud.

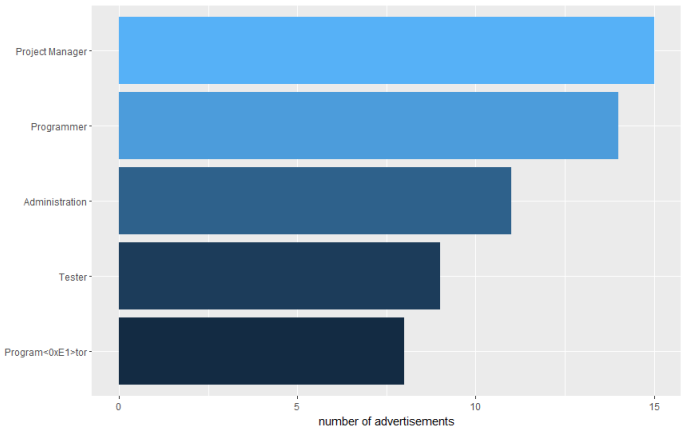

Figure 3.21: Top professions.

3.2.1.3.10 Bookmarks This value is gathered after advertisement publication, so even though it is correlated with the label (see section 3.2.2.2), it is not really helpful in our primary goal. However, it is of course possible to create models with it to see if it is a good predictor at least. The barplot 3.22 shows the data structure.

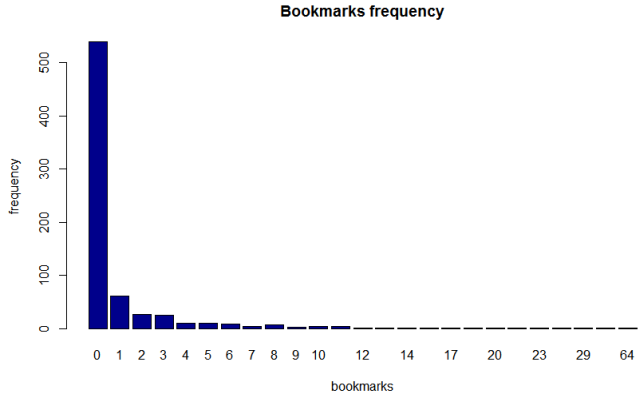

Figure 3.22: Bookmarks barplot.

3.2.1.3.11 Detail views This value is gathered after advertisement publication similarly as the bookmarks variable, so even though it is correlated with the label (see section 3.2.2.2), it is not really helpful in our primary goal. However, it is of course possible to create models with it to see if it is a good predictor at least.

The range of views is very wide and the most common number of views is zero, so one of the best ways to visualize this variable is histogram 3.23.

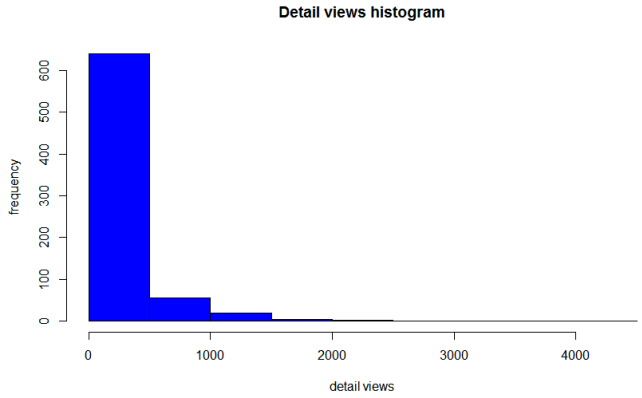

Figure 3.23: Detail views histogram.

3.2.1.3.12 Data boxplots To get a better idea about the all data dataset values range and distribution, the boxplot graph 3.24 is a perfect tool. It makes clear that some variables follow geometric distribution and that most of them have a wide range of values even in the whole data dataset. Compared to selected variables in boxplots 3.25, they have a bit greater variance.

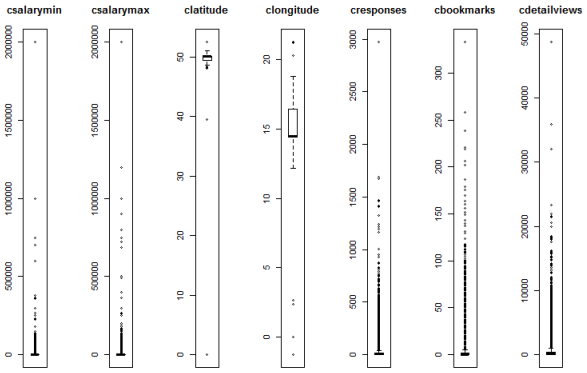

Figure 3.24: All data dataset boxplots.

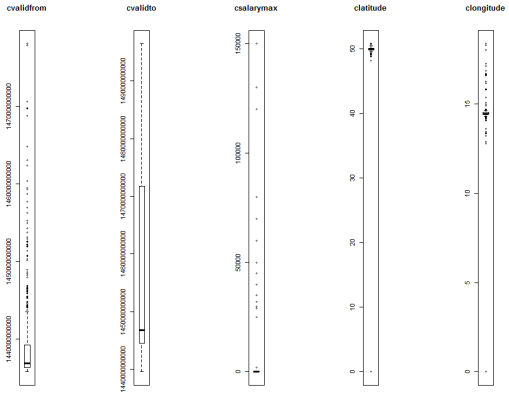

Figure 3.25: 700 dataset boxplots.

3.2.1.4 Binning the label variable

The label variable is binned into two classes, based on two facts: firstly, its distribution 3.1, indicating rather equal-height discretization, and secondly (and more importantly) on the fact that we are generally interested in applicable output, therefore we want to select out the low and the high values of responses primarily. The two classes are as follows:

- Class 0: 0—3 responses
- Class 1: 4 and more responses

The distribution of the label variable responses to classes is depicted in figure 3.26.

Figure 3.26: Label binning.

3.2.1.5 Specifying the variables

Other interesting data can be gained when the responses are investigated considering whether other variables are stated or not. Generally, the probability of an ad to be classified as 4+ responses class is approx. 60.06 % (290 records in class 0—3 and 436 in the second class).

If the salary information is provided (both minimum and maximum), the probability slightly rises to 63.64 % (12 records in class 0—3 and 21 in the second one).

If the profession is specified, the probability slightly rises to 62.72 % (170 records in class 0—3 and 286 in the second one).

If the location information is provided (both latitude and longitude), the probability is comparable, 60.88 % (277 records in class 0—3 and 431 in the second one).

3.2.2 Feature selection

The next lines discuss which attributes are useful and which not to the label prediction based on the statistic measures.

3.2.2.1 Andrews plot

In figure 3.27, depicting all the numerical variables from the 700 dataset as an Andrews plot, it can be observed that there is probably some kind of structure in the data, but the course of the curves is similar and the individual categories (black for 0—3 responses and red for 4+ responses) permeate each other. This tells that it might be possible to try and find a relationship between solely numerical input variables and the number of responses, but according to this visualization it is certainly a good idea to add some more information to the

models to be able to predict the label. The additional information can be gathered from nonnumerical variables—categorized or textual.

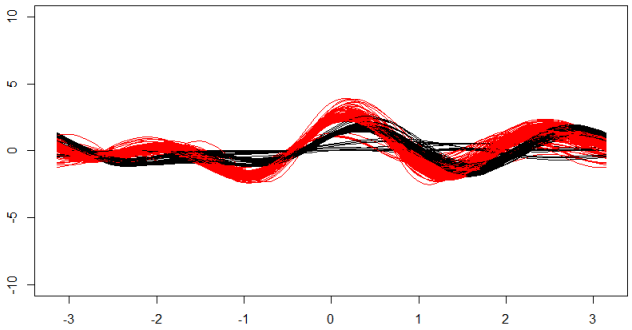

Figure 3.27: Andrews plot.

3.2.2.2 Correlations

The most strong correlations are between minimum and maximum salary, between latitude and longitude, and between the three of responses, bookmarks and detailviews, as is recognizable from the darker fields in the Correlations matrix 3.28, blue represents positive correlation and red negative. The individual relations between variables are even more visible in figures 3.29 and 3.30.

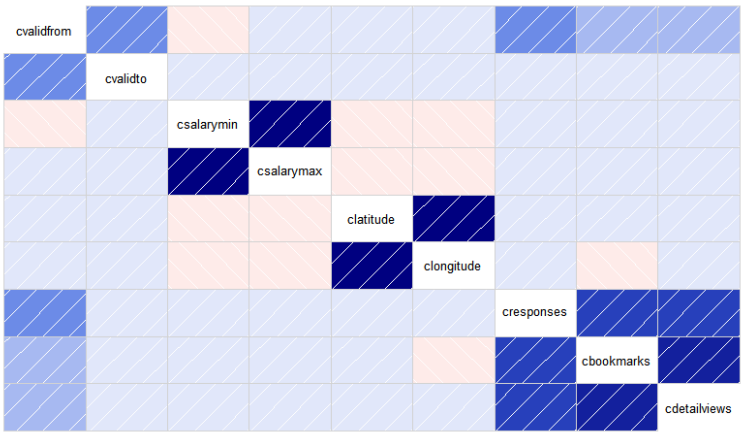

Figure 3.28: Matrix representation of individual correlations.

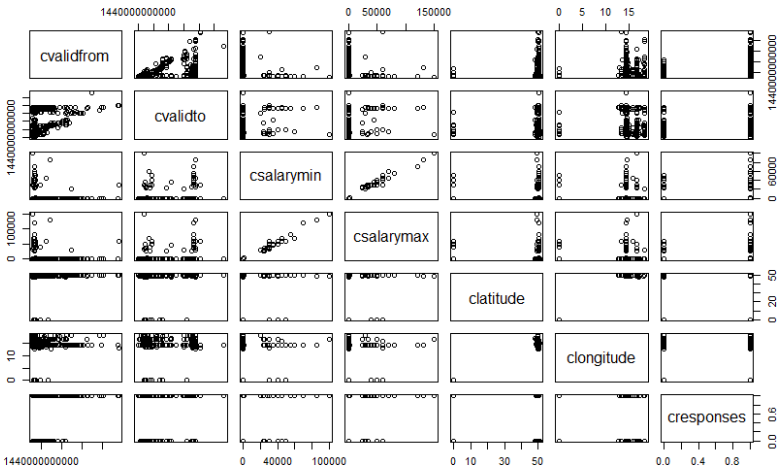

Figure 3.29: Visual matrix representation of majority of individual correlations.

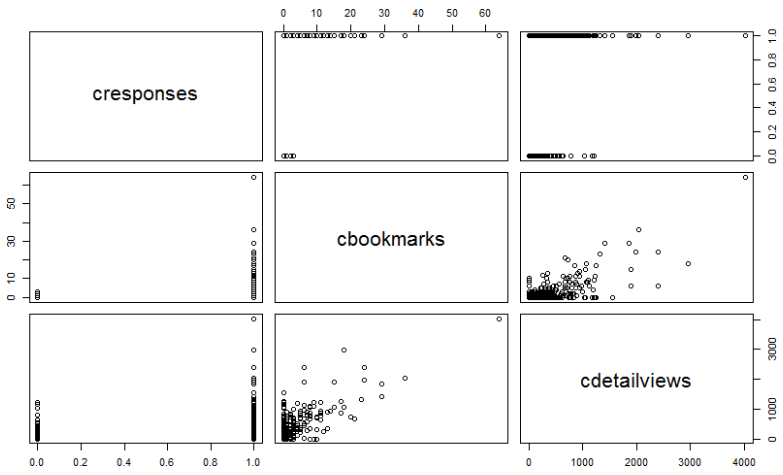

Figure 3.30: Visual matrix representation of the rest individual correlations.

Another possibility to visualize multi-dimensional data is to create a parallel coordinates plot. Figure 3.31 shows that there should be a relation between the label category and the two attributes that were shown to be in correlation with the label, i.e. bookmarks and detail views.

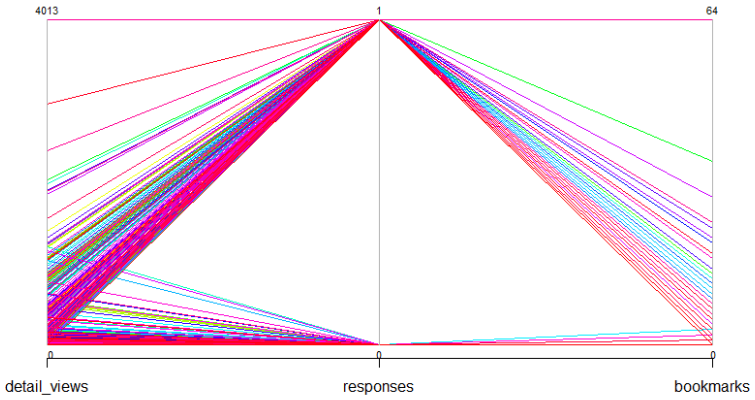

Figure 3.31: Parallel coordinates—bookmarks, detail views, and responses.

Based on the correlations between variables, it is clear that minimum salary attribute is useless because it pretty much copies the maximum salary attribute, so it will not be included in any model. Similar is the case with latitude and longitude, but the full information about location could be useful so it could be risky to drop one of these variables just because there is a correlation between them.

3.2.2.3 Description of the ad

A feature that could be extracted from the job advertisement is the extent of requirements on the applicants. The ”juniority” or ”seniority” required for the position is somehow encoded in the natural language using certain words in certain context (for example ”skilled, senior, experienced” or ”graduate, junior, fresh start”). It should be possible to extract it from the job description using a context-wise text mining algorithm, such as skip-gram. This feature could provide the models with more insight whether or not the people will answer the job advertisement.

3.2.3 Variable transformations

The data scaling process is done while building each individual model, if necessary. For example, Deep Learning model works properly only when provided with standardized data. Otherwise, its output would be biased.

Algorithms based on decision trees (Distributed Random Forest or Gradient Boosting Machine) are immune to variable scaling because they make decisions by comparing the values.

3.3 Proposed models

Proceeding from the feature selection and analysis, I define several more or less distinctive models that will be created and compared.

3.3.1 Model_1

Model_1 has latitude, longitude and maximum salary as its input attributes. The relations of the variables look messy at first 3.32, but the second look tells that there could be some kind of regular pattern. Based on the distributions of all three input variables (a good variance especially in case of longitude), this model could perform reasonably good, so it is a reasonable starting point to reference.

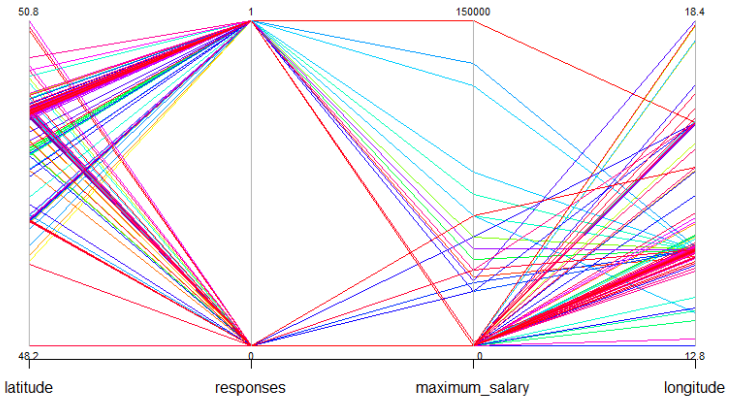

Figure 3.32: Parallel coordinates of the Model_1.

3.3.2 Model_2

Model_2 is composed of categorical variables—branches, professions, and company. This gives it an opposite approach to predict the label. The visualization using parallel coordinates 3.33 is a bit tricky because the y axis does not represent a comparable value, it represents a (random) bin. Although, it gives a rough image about the density of individual variables. Let's give this model a chance because in the individual categories could be stored some information based on the distributions and top representatives depicted in figures 3.16, 3.19, and 3.21.

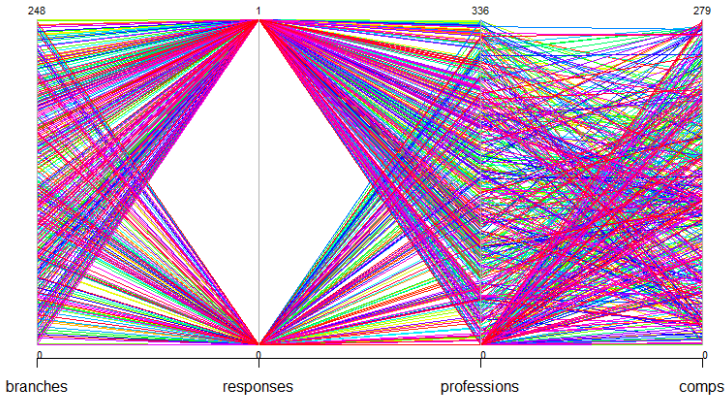

Figure 3.33: Parallel coordinates of the Model_2.

3.3.3 Model_3

The third model combines two categorical variables (branches and professions) with the extracted information from the job description, using Word2Vec algorithm. Two variations are created, one based on the sample of ad descriptions that have fewer requirements (junior) and one using a set of more demanding requirements (senior).

3.3.4 Model_4

Model_4 consists of publication dates, location attributes, maximum salary, professions, and branches. This model probably has the maximum information and has a decent chance to perform the best, also because the correlation between the used attributes is quite low as was shown before.

3.3.5 Model_5

Model_5 has input variables that are a subset of the input variables for Model_4, consisting of location (latitude and longitude), professions, branches, and maximum salary. This model could show whether the publication date variables are important to predict the label.

3.4 Building the models

3.4.1 Tools

All mentioned models were built using H2O REST API within the R.

The R source codes to process the data and build models are in the enclosed DVD.

Primary sources to create the R codes were H2O documentation and examples and R documentation and examples [9] [14] [15] [16] [17].

I tried to improve the models that had promissing AUC and/or low error rate. To do this, the model predictions were filtered in the manner that only the job advertisement instances the model was certain about were classified. The cases when the model was not confident on a level surpassing 80 % were eliminated from the validation frame.

3.4.2 Model_3

3.4.2.1 Text processing

The aim is to extract the information about amount of requirements on the position, supposing that it will be reflected in the applicants' demand, maybe in combination with other variables.

At first, two category samples need to be created based on the job description: the one representing few requirements and targeting more junior applicants, and the one with more demanding specifications and aiming on more experienced candidates. The two sets of job descriptions that were carefully chosen are not from the 700 dataset. They are stored in separate files.

The description of ads is in plain text, so it has to be processed firstly. That was done by tokenizing the description text into single words, then filtering out the ones that bear little or no useful meaning (so called stopwords). Words shorter than 4 characters were dropped.

The next step was creating two Word2Vec models (one considering the junior positions and one considering the senior ones) from the words present in the junior/senior category samples files. They were built with minimum word frequency 4 to ensure that less common and too specific words are not used in the model using the function

```
w2v.model.junior <- h2o.word2vec(
                    words.junior,
                    sent_sample_rate = 0,
                    epochs = 5,
                    min_word_freq = 4)
```

, where *words.junior* is tokenized, filtered and processed list of words made of job descriptions.

The built Word2Vec models were each transformed to vectors, using description of the ads in 700 dataset as the input. That means two sets of vectors were built: one representing the 700 dataset job description with bias towards junior positions and one set of vector that incorporates the bias in favour of senior positions. These columns can be passed as input variables to a model that will predict the label. In R, the method

```
h2o.transform(w2v.model.junior,
        words,
        aggregate_method = "AVERAGE")
```

returns the vector representation of the *words* (tokenized job descriptions from the 700 dataset) based on the *w2v.model.junior*.

3.4.2.2 Individual models

3.4.2.2.1 GBM The model was built using the function

```
h2o.gbm(x = x,
        y = "cresponses",
        training_frame = data.split[[1]],
        validation_frame = data.split[[2]],
        seed = 333)
```

, where x is the vector with names of input variables and *data.split[[1,2]]* are prepared training and validation frames. Seed is used to ensure reproducibility of the results.

3.4.2.2.2 DL The model was built using the function

```
h2o.deeplearning(x = x,
                 y = "cresponses",
                 training_frame = data.split[[1]],
                 validation_frame = data.split[[2]],
                 standardize = TRUE,
                 activation = "Rectifier",
                 epochs = 100,
                 seed = 333,
                 reproducible = TRUE,
                 hidden = hidden,
                 variable_importances = TRUE)
```

, where x is the vector with names of input variables and *data.split[[1,2]]* are prepared training and validation frames. The model is built on standardized input data. One thread computing and seed are used to ensure reproducibility. Two hidden layers were used, each composed of 10 neurons.

3.4.2.2.3 DRF The model was built using the function

```
h2o.randomForest(x = x,
                 y = "cresponses",
                 training_frame = data.split[[1]],
                 validation_frame = data.split[[2]],
                 seed = 333)
```

, where x is the vector with names of input variables and *data.split[[1,2]]* are prepared training and validation frames. Seed is used to ensure reproducibility of the results.

3.4.3 Other models

Other four models were made using other input data based on the specifications in section 3.3. They were built the same way the Model_3 was, without need to process the text variables.

3.5 Results interpretation and comparison

The next part evaluates the performance of built models.

3.5.1 Models performance

The individual results (confusion matrices and AUC) are usually mentioned in the source code after the corresponding model.

The first model performed the best with GBM, it exhibited a high AUC of 0.707 after applying the model confidence constraint (raise from 0.654). However, the validation sample lowered to only 31 records. DRF performed even better with AUC 0.75 after applying model confidence constraint, but the validation sample lowered to only 12 samples and that gives no evidence about the model besides the fact that it is really uncertain about itself. Error rate of DL model raised after not classifying the uncertain instances, meaning that in this case the model was completely wrong.

The second model had all the individual models' AUC between 0.68 and 0.702 with the limited validation data and the number of instances the models were certain of stayed quite high (115, 124 and 171 out of 191 instances), meaning that the models are confident.

The best combination in the third model was senior model and GBM, it performed with AUC equal to 0.666 after reducing the number of instances based on model confidence. However, that is still not as good as other models.

The fourth model performed best with DRF (AUC 0.73) and DL (AUC reaching 0.719), while still considering more than a hundred instances in the trimmed validation dataset. The DL model got better by more than 7 % from AUC 0.646 using the model confidence constraint.

The fifth model performed best with DRF (AUC 0.715) and DL (AUC reaching 0.694). So it seems that the publication dates bear some useful information because without them, the prediction got a bit worse.

The overall best AUC, taking into account only representative validation samples, was achieved with combination of Model_4 input variables (publication dates, location attributes, maximum salary, professions and branches) and DRF method, reaching to 0.73. Before applying the model confidence

criterion, the AUC was 0.684. The confusion matrix of the best performing model is in table 3.1. Vertical numbers denote actual values, horizontal denote the predicted values.

Table 3.1: Confusion table for Model_4 using DRF.

	Class 0	Class 1	Error	Rate
Class 0	19	27	0.586957	=27/46
Class 1	3	63	0.045455	=3/66
Totals	22	90	0.267857	=30/112

Generally, the results are similar and moderate. In most cases the DRF performed best, but this could be a coincidence. DL models were usually also good. With Model_3, which used Word2Vec, GBM had best results compared to DRF and DL.

3.5.2 Variable importances

The comparison of variable importances of GBM in Model_3, senior model is shown in figure 3.34. We can see that the Word2Vec variables (beginning with "C") are less important than the other two, branches and professions. On the other hand, there are one hundred of Word2Vec variables in the model, so they are not completely meaningless.

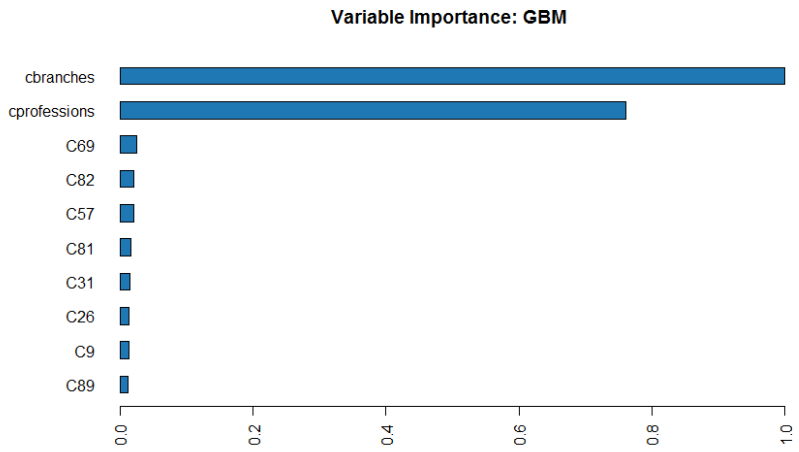

Figure 3.34: Variable importances of the Model_3 with GBM.

The variable importances of the best performing model are depicted in figure 3.35.

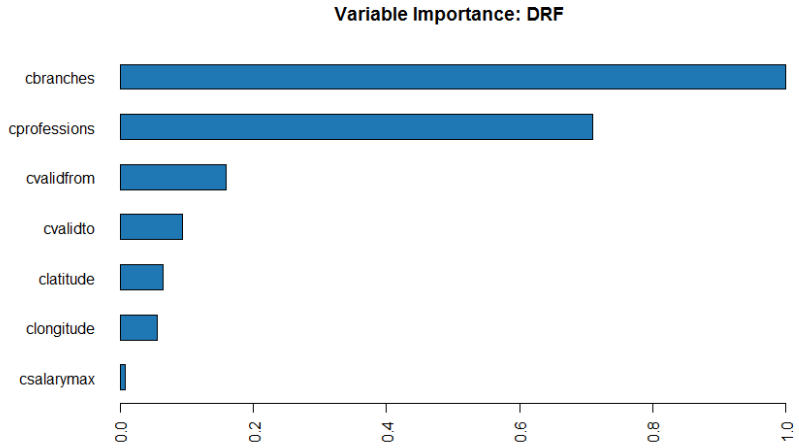

Figure 3.35: Variable importances of the Model_4 with DRF.

In these two plots, both branches and professions hold first positions. Both being category variables, this means that the numerical attributes are not sufficient enough to predict the label (specifically, they do but the created models are not as confindent as the models that also use categorical variables).

The most common branch, "IS/IT: Application and system development", has only 0.24 (=8/34) probability to be in the class with more responses. The second most common branch, "Ekonomika a podnikové finance" (Economics and Corporate Finance) has, on the contrary, 0.67 (=20/30) probability to be answered by more candidates. And the "Administration" branch has this probability as high as 0.72 (=13/18).

As for professions, "Project Manager" has probability of 0.47 to be answered by decent number of applicants, "Programmer" only 0.36 ("Programátor" has almost identical 0.37). All five "Brand manager" advertisements in the 700 dataset were answered by more than 3 people.

3.6 Other approaches

While attempting to build reliable predictive models, I also tried some approaches that are a bit different.

3.6.1 Using bookmarks and detailviews

As a matter of interest, I built Model_6 that has detail views and bookmarks as input variables. This is a bit misleading, because these values are gathered after job ad publication and are highly correlated with the label—see section 3.2.2.2. Nevertheless, DL has the best results. After applying the model

confidence criterion, the AUC skyrockets from 0.662 to 0.972. Table 3.2 shows what was expected: error rate is very low. The ameliorated DL model classifies only in the class with more responses, but it hits almost everytime.

Table 3.2: Confusion table for Model_6, DL.

	Class 0	Class 1	Error	Rate
Class 0	0	1	1.000000	=1/1
Class 1	0	36	0.000000	=0/36
Totals	0	37	0.027027	=1/37

The GBM model scored 0.934 AUC (63 records after validation data reduction based on model confidence) and DRF 0.875 AUC (72 records left).

3.6.2 Tf-idf

Aiming on useful output, there is a possibility to analyse the job description specifics. Tf-idf algorithm was applied in a way that sets of descriptions are categorized as documents regarding the number of answers and the knowledge what are the (un)successful ads about can be attained.

For this purpose I singled out zero responses as a stand-alone class. The typical words for each category are depicted in figure 3.36. The most-answered category 2 (4+ responses) descriptions present topics like *resale* or *trade* and use other words like "study, typical, helpdesk". At the other end, the unsuccessful ads are typical in using words with little meaning like "appear, redirected, signed, subscribe" that do not describe the advertised position.

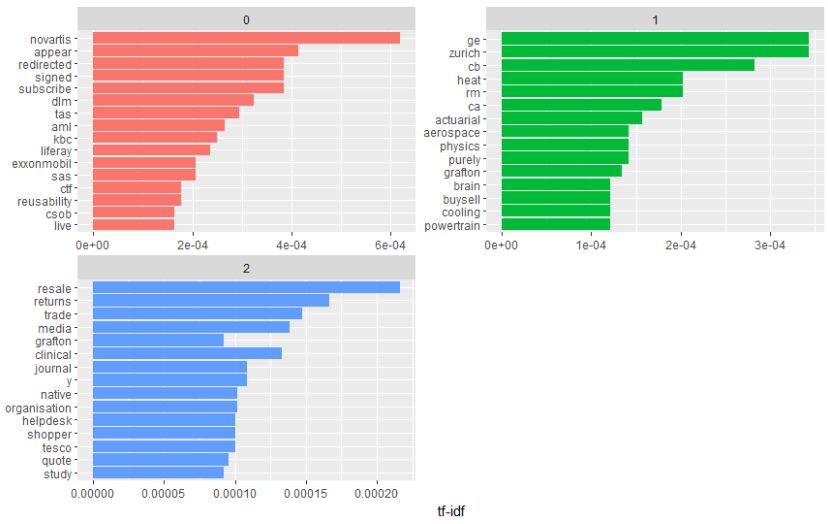

Figure 3.36: Most significant words according to tf-idf for three categories based on number of responses.

3.6.3 Binning into more classes

I tried other binning possibilities, for example into 4 classes. The distribution resembles equal-height distibution and is drawn in figure 3.37. However, the error rate of built models was regularly around 50 %, so this was not a good approach.

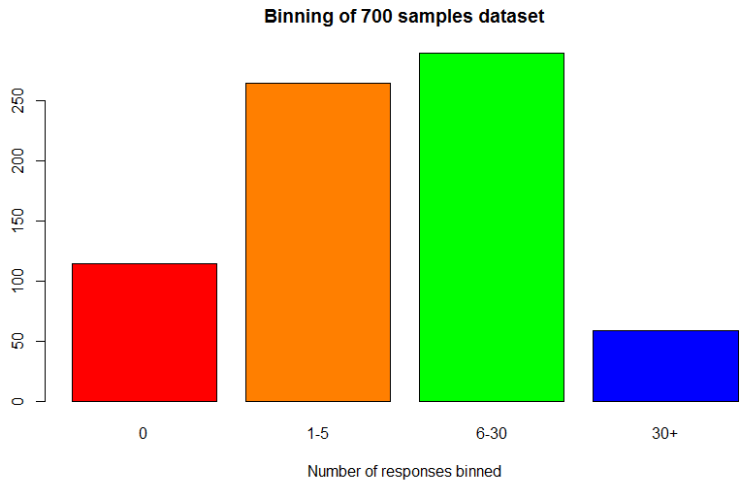

Figure 3.37: Binning into 4 variables.

3.6.4 Predicting exact number of responses

I also tried to build models predicting the exact number of responses. The mean value of responses is 10.62, maximum value is 170 and the mean average error of the best model was around 7. This does not tell much but based on common sense, the built models were not of a high quality.

Conclusion

Summary

I learnt that the best way to predict number of responses to the job ad is using data collected while the ad is published, but this has little or no value in the moment of ad creation. Some interesting relations in the data were presented, for example that not stating salary or location information can lead to less responses. Other than that, predictive models were built based on the input data and some of them had their AUC greater than 0.70, which is not much, but it is surpassing a level of pure coincidence and guessing or trying to constantly predict one of the classes no matter the input variables. Theoretically, the built models can be used to predict whether more than 3 applicants will reply to the job advertisement even though the success rate is not as high as desired. The way of building this and other models was described and it is reproducible using the source code found in the enclosed DVD.

Prospects

All presented findings can be used mainly in potential further attempts to predict the number of advertisement respondents. Some other text mining methods could be applied to extract the information from text attributes. Larger dataset of English job advertisements could be used, too. Another possibility is to use some other attributes that I do not currently have at disposal.

One possible future progress could be combining these predictive models with recommendation systems and acting in the live, production environment using the real-time data.

Bibliography

[1] IBM100 — Deep Blue. [online], March 2012, [cit. 2016-12-1]. Available from: http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/deepblue/

[2] METZ, C. Google's AI Wins Fifth And Final Game Against Go Genius Lee Sedol. *Wired*, March 2016, [cit. 2016-12-11]. Available from: https://www.wired.com/2016/03/googles-ai-wins-fifth-final-game-go-genius-leesedol/

[3] finance.cz. 5% nezaměstnanost v říjnu: Nejnižší míra od konce roku 2008. [online], 2016, cit. 11.12.2016. Available from: https://www.finance.cz/479189-nejnizsi-nezamestnanost-od-2008/

[4] Google Cloud Platform. Cloud Jobs API. [online]. Available from: https://cloud.google.com/jobs-api/

[5] Google Cloud Platform. Cloud Jobs API: machine learning goes to work on job search and discovery. [online]. Available from: https://cloud.google.com/blog/big-data/2016/11/cloud-jobs-api-machine-learning-goes-to-work-on-job-search-and-discovery

[6] GUYON, I.; ELISSEEFF, A. An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research 3*, 2003, cit. 11.5.2017. Available from: http://www.jmlr.org/papers/volume3/guyon03a/guyon03a.pdf

[7] KHATTREE, R.; NAIK, D. N. Andrews plots for multivariate data: some new suggestions and applications. *Journal of Statistical Planning and Inference*, 2002, excerpt available from: http://www.sciencedirect.com/science/article/pii/S0378375801001501.

[8] GUTHRIE, D.; ALLISON, B.; et al. A Closer Look at Skip-gram Modelling. Cit. 11.5.2017. Available from: http://homepages.inf.ed.ac.uk/ballison/pdf/lrec_skipgrams.pdf

[9] SILGE, J.; ROBINSON, D. tidytext: Text Mining and Analysis Using Tidy Data Principles in R. *JOSS*, volume 1, no. 3, 2016, doi:10.21105/joss.00037, more info svsilsble from: http://tidytextmining.com/tfidf.html. Available from: http://dx.doi.org/10.21105/joss.00037

[10] The H2O.ai team. Data Science Algorithms. [online], 2017, cit. 3.5.2017. Available from: http://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science.html

[11] FAN, J.; UPADHYE, S.; et al. Understanding receiver operating characteristic (ROC) curves. *Canadian Journal of Emergency Medicine*, volume 8, no. 1, 2006: p. 19, doi:10.1017/S1481803500013336. Available from: https://www.cambridge.org/core/services/aop-cambridge-core/content/view/S1481803500013336

[12] Google Maps. Map of Prague and surroundings. May 2017, acquired via Google Maps API.

[13] KAHLE, D.; WICKHAM, H. ggmap: Spatial Visualization with ggplot2. *The R Journal Vol. 5/1*, 2013, cit. 11.5.2017. Available from: https://journal.r-project.org/archive/2013-1/kahle-wickham.pdf

[14] The H2O.ai team. *h2o: R Interface for H2O*. 2017, r package version 3.10.4.3. Available from: https://github.com/h2oai/h2o-3

[15] WICKHAM, H.; FRANCOIS, R. *dplyr: A Grammar of Data Manipulation*. 2016, r package version 0.5.0. Available from: https://CRAN.R-project.org/package=dplyr

[16] FELLOWS, I. *wordcloud: Word Clouds*. 2014, r package version 2.5. Available from: https://CRAN.R-project.org/package=wordcloud

[17] michalkurka. word2vec, transform sentences to vectors by averaging the word-vectors. [online], 2017, cit. 3.5.2017. Available from: https://github.com/h2oai/h2o-3/blob/master/h2o-r/demos/rdemo.word2vec.craigslistjobtitles.R

Appendix A

Glossary

AI Artificial intelligence, i.e. intelligence manifested by machines

AUC Area under the curve, a model quality measure

API Application programming interface, tools and definitions to build specific software

Data mining Extracting knowledge from data using various processes and algorithms

Dataset Assemblage of data, usually in a table form

DL Deep Learning, a multi-layer artificial neural network, a classification and regression method

DRF Distributed Random Forest, a classification and regression method

Feature Model input; usually one measurable variable

GBM Gradient Boosting Machine, an ensemble method

H2O Open-source platform for analyzing data, developed by the *H2O.ai*

HR Human resources, usually a company department

Label Output variable that is to be predicted

Model certainty, model confidence Probability computed by the model that indicates how certain the model is about the individual prediction

Ontology Formal definition of types and relationships in a domain

Open-source software Software that has its source code available, usually under a less or more restrictive licence describing terms of use

Predictive modeling Using statistical methods to reckon the desired output

R A language and environment for statistical computing, developed by the *R Development Core Team*

Text mining Extracting knowledge from text using various processes and algorithms

Tf-idf Term frequency-inverse document frequency

Word2Vec Model that creates vector from a word

Wordcloud Visual respresentation of most used words

Appendix B

Contents of enclosed DVD

- `readme.txt` ..... the file with DVD contents description
- `data` ..... the directory with used data
- `model` ..... the directory with model
- `src` ..... the directory of source codes
  - `impl` ..... implementation sources
  - `thesis` ..... the directory of LaTeX source codes of the thesis
- `text` ..... the thesis text directory
  - `BP_Smid_Martin_2017.pdf` ..... the thesis text in PDF format