Predicting Salaries with Random-Forest Regression

Frank Eichinger and Moritz Mayer

Abstract For companies it is essential to know the market price of the salaries of their current and prospective employees. Predicting such salaries is challenging, as many factors need to be considered, and large real datasets for learning are scarce. For this reason, research on salary predictions is comparably rare and limited. In this study, we investigate whether and how an advanced machine-learning approach, namely ensembles of random-forest regression, can achieve high-quality salary predictions. We use a large real dataset of more than three million employees and more than 300 professions. Our approach learns – for each profession – a random-forest regression model to predict salaries. In our evaluation, we show that this approach performs better than related work on salary prediction by machine-learning approaches with a mean absolute percentage error (*MAPE*) of 17.1%. We identify reducing the number of possible values of categorical variables, training separate models as well as outlier handling as the key factors for the results achieved.

1 Introduction

Paying competitive salaries is essential for companies of any size to retain current and attract new employees. At the same time, paying more than the market price is equally undesirable from a company perspective. Determining the market value for a particular employee or candidate is challenging, as salaries are influenced by many factors. These include the profession, the region, the age, the work experience, the company industry and the company size. Estimating competitive salaries requires a database of (close to) real salaries in a good data quality. If large amounts of data records are available, comparison groups can be built to benchmark salaries and

Frank Eichinger
DATEV eG, Nuremberg, Germany e-mail: frank.eichinger@datev.de

Moritz Mayer
DATEV eG, Nuremberg and University of Bamberg, Germany, e-mail: moritz.mayer@datev.de

to visualise salary distributions, for instance, of a certain profession, and provide key numbers such as median values (see Figure 1 for an example). This can help employees, employers and consultants to find out if a certain salary is within the usual range. For the German market, there is the "Entgeltatlas" [19] of the German Federal Employment Agency and the commercial product "Personal-Benchmark online" [11] from DATEV eG which provide benchmark services based on large volumes of real data. These tools display salary distributions based on profession and region and partly on age and gender. To ensure statistical validity and privacy, distributions need to enclose a certain number of individuals. Hence not all combinations of the factors mentioned are valid and can be selected. However, these solutions do not consider further factors than profession, region and demographics. Other relevant attributes such as the company industry and size may not be looked at.

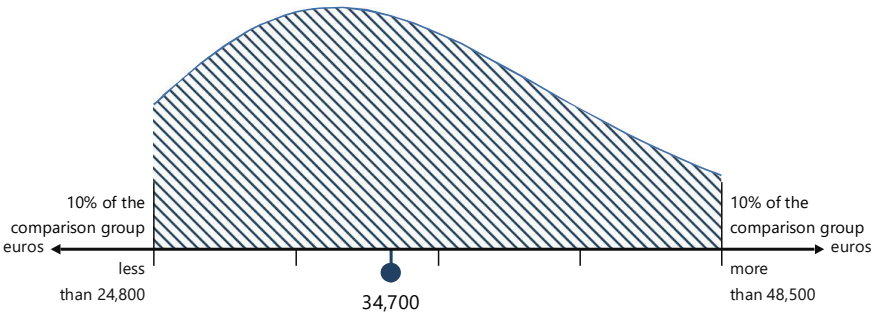


Fig. 1: Example salary distribution from [11] including median and percentiles

If fewer data records are available or if a broader range of attributes should be used to be more specific, comparison-group benchmarks cannot be used. Alternatively, regression models may be used to predict salaries. This has the advantage that the output is a numerical value in a currency which refers to a competitive salary. Such values may be easier to interpret than a salary distribution alone. Further, when a company hires a new employee, a regression-based approach can propose an adequate salary. Particularly when the regression model uses more attributes than the distribution of a comparison group, the predicted value typically is closer to the competitive salary than the average or median of a distribution. This is an advantage for human-resources managers using salary prediction tools, but also, for instance, for tax advisers who get enabled to offer business consultancy to their clients. Several commercial providers publish market overviews and individual salary predictions derived from salary data obtained from surveys or interviews of employers or employees. However, the exact statistic approaches employed are usually confidential. One approach, the "Gehaltsvergleich BETA" [21] of the German Federal Office of Statistics uses a relatively small sample of real salary data [24] and employs a specialised linear regression model to predict salaries. However, if more data records are available, machine learning, concretely more sophisticated

regression models, may lead to better predictions. A general problem, particularly in the scientific literature (see Section 2), but also for commercial offers, is the limited availability of salary data in good quality. Therefore, most studies focus on very specific industries or markets or use quite outdated or small datasets.

In this study, we investigate whether and how machine learning, particularly regression trees and random-forest regression, can achieve high-quality salary predictions on a large dataset of salary data. As a dataset for learning and evaluation, we use a sample of roughly three million real payslips each month over one year including more than 300 professions originating from the payroll software from the German company DATEV eG. In a nutshell, besides several pre-processing steps including outlier removal, we propose an ensemble regression approach which learns – for each profession – a random-forest regression model to predict salaries. In our comprehensive evaluation, we show that this approach based on a large real dataset (a) performs better than related work on smaller datasets (comparing the error measures published) and (b) that the prediction errors can be reduced by 17.8% compared to our baseline.

The contributions of this study are as follows: (1) We show that more sophisticated machine-learning models than linear regression, namely random-forest regression, are suitable to predict salaries on a large dataset. (2) We demonstrate that an ensemble of one regression model per value of a categorical independent variable may clearly outperform a single regression model in situations where this variable has many values. (3) Our evaluation is based on a real-world dataset of millions of payslips, which cannot be found in the scientific literature in a comparable data quality and quantity.

The remainder of this chapter is organised as follows: Section 2 reviews related work, Section 3 describes our approach, Section 4 presents our results, Section 5 discusses them and gives directions for future research and implementation, and Section 6 concludes.

2 Related Work

The state of research on statistical models for salary prediction is rather weak. Although there is some work on salaries and their influencing factors as well as on machine-learning approaches for prediction, the focus is usually on specific aspects. This is either a relatively narrow selection of employees, for example, only managers [34], or a relatively narrow selection of industries or sectors, for example, only hospitals [43] or universities [4]. The goal of developing a statistical predictive model that (a) covers a broad picture of salary factors and (b) is valid for as many employees as possible has been pursued in comparatively few studies.

The work of Chakraborti [9] compares the predictive performance of five machine-learning algorithms for salaries of U.S. census-data individuals. Tree-based algorithms achieved the highest performance. Yet the census data used in the study is quite outdated – from 1994 – and the salaries entailed are only available in the

form of a categorical variable (> 50,000 $ and <= 50,000 $). Chakraborti addresses this situation and reasons that having to use such an outdated dataset is illustrating the lack of available datasets and research by other authors on the topic of salary prediction. A study by Viroonluecha and Kaewkiriya [39] applies various machine-learning algorithms to assess their salary prediction performance on data crawled from a job platform in Thailand. The crawled data is relatively new – from 2018 – but also limited: only employees with academic education are considered and the dataset is relatively small (39,000 employees). The used neural network achieves the highest performance, closely followed by the random-forest model of the study. The performance metric used is the root-mean-square error (*RSME*). The result of 7,740 ฿ (approximately 210 €) is difficult to assess, as there is no further information on, for example, a relative deviation from the target, which a metric like the mean absolute percentage error (*MAPE*) would give. Neither does the study provide information on distribution metrics of the actual salaries in the dataset such as the mean, the median or percentiles.

A related field of research which may use the same kind of data is employee churn prediction [38, 41]. However, the studies in this field likewise are struggling with the lack of available data and do not provide any additional insights into the selection of machine-learning algorithms for salary data. In essence, the analysis of the academic work in the field of salary prediction leads to the conclusion that related work on salary prediction is scarce. A significant lack of publicly available, high-quality datasets on employee salaries is hindering researchers to conduct more studies researching the effectiveness of machine-learning algorithms in salary prediction.

Apart from the scientific literature, there are several commercial solutions for salary predictions. However, most of them do not publish how they compute them. One exception is the "Gehaltsvergleich BETA" [21] of the German Federal Office of Statistics. The tool employs specialised linear regression models [37] to predict salaries and is based on a relatively small sample (600,000 employees) of real salaries [24]. It is therefore the product and study closest to the research presented in this chapter.

3 Data Preparation and Random-Forest-Regression Modelling

In this section, we describe our dataset, the steps for pre-processing including outlier handling, the selection of machine-learning models and our choice of an ensemble of ensemble (random forests) approach for predicting salaries.

3.1 Data Source, Data Analysis and Feature Selection

In this study we work with a subset of a dataset of pseudonymised payslips extracted from the payroll accounting solutions of DATEV eG. This dataset comprises the

payslips of 3.14 million employees in Germany over one year not including any trainees, working students, marginal employment or employees working less than 15 hours per week. The dataset available is limited to the 330 professions which occur most frequently. In order not to be affected by any effects of the Covid-19 pandemic and the resulting large shares of short-time allowances in certain industries such as gastronomy, we have used data from the year 2019.

3.1.1 Dependent Variable

The dependent variable "salary" needs to be specified in more detail. We have chosen to predict the *annual gross income*, a numerical variable in Euros. The reason is that this value includes all special payments such as holiday pay, Christmas bonus, further bonuses etc. This makes it easier to compare as monthly salaries do not include such payments and some employees may not have them and may have higher monthly payments instead. We have extrapolated months with missing or only partial payments, for instance, when the employee was sick. Furthermore, we have extrapolated values for employees working in part-time (less weekly working hours than the company default) to the company-default weekly working hours (typically 40 hours) to have comparable numbers. Our preliminary experiments have shown that extrapolating to the company default leads to better results than extrapolated to a fixed number such as 40 hours. Our dependent variable, the *annual gross income*, has a log-normal distribution, which is in line with the literature [28]. This means that the income distribution is skewed to the right and displays a long right tail.

3.1.2 Independent variables

We have chosen the attributes listed in Table 1 from our dataset as independent variables as an input for our prediction models. Our data analyses have shown that all independent variables have a medium to high correlation with our dependent variable. If we look at their correlations for each profession, it is obvious that our variables have very different correlation values for the various professions. As one example, the *level of education* is correlated with the salary. However, looking at information-systems professionals as an example, the *level of education* only has a negligible influence on the salary as the vast majority of them has a college degree.

The categorical variables *company industry* and *federal state* have many possible instances. As this is not optimal for many machine-learning algorithms, we have grouped them. Our preliminary experiments have shown that this may slightly increase predictive performance. For the *company industry*, we use the hierarchical structure of the official taxonomy [22] and assign one of 23 sections to a company. Regarding the *federal state*, we cluster the states into four groups by minimising differences in the median of the *annual gross income* within a cluster and maximising it between the clusters. Also, the variable *profession* has many possible instances. We propose a specific handling in Section 3.4.

Table 1: Independent variables

	Variable	Type
employee	*profession* [23]	categorical
	age	numerical
	gender	categorical
	level of education [20]	ordinal
	level of professional training [20]	ordinal
	contract type (full time/part time[1], temporary/permanent) [20]	categorical
employer	*federal state*[2]	categorical
	degree of urbanisation[2][3] [14]	ordinal
	company size (number or employees)	numerical
	company industry [22]	categorical

3.2 Outlier-Handling Strategy

Our dependent variable, the *annual gross income*, has a large spread. While the distributions of the variable are quite different for the various professions, also the spread within a profession can be quite large. Many salaries exceed the median salary by a factor of more than 1.5. For example, while the median value for a secretary is around 35,000 € per year and the 90% percentile is around 55,000 € per year, there are individuals earning 100,000 € and more. Very likely, these data points are outliers resulting from mistakes when entering the profession into the payroll software or not updating it when the employee has climbed-up in career. Another example are salaries below the German minimum wage probably caused by weekly working hours incorrectly entered into the payroll software. As outliers may affect prediction models quite heavily, a well-chosen strategy for outlier removal is essential. We apply our outlier handling to our whole dataset before splitting it in training, validation and test sets.

Instead of a simple approach for outlier removal which removes the highest and lowest, say, 5% of data points per profession, we choose a more thorough approach. The inter-quartile-range – the difference between the 75% and the 25% percentile – is commonly used in box plots to create the whiskers which determine the upper and lower threshold for outliers. We have investigated the percentage of outliers in comparison to different inter-quartile-range factors (*IQRF*), which allowed us to understand the outlier situation in the data more deeply. Initially we have chosen

[1] Our analyses have shown that this may influence results even if we extrapolate to full time.

[2] We derive the *federal state* and the *degree of urbanisation* from the company zip code.

[3] The *degree of urbanisation* has three possible values which are ordered:

1. *Cities* (densely populated areas)
2. *Towns and suburbs* (intermediate density areas)
3. *Rural areas* (thinly populated areas)

a conservative *IQRF* of 3 removing only 1% of the data, and we switch in our evaluation (see Experiment 4 in Section 4.2) to the commonly used *IQRF* of 1.5, removing 4% of the data.

3.3 Selection of a Machine-Learning Approach

While relatively simple linear regression models have been used in the related work [24] on smaller datasets having similar attributes, we assume that more sophisticated machine-learning models may achieve better performances when more data is available. While several models such as support-vector machines and neural networks may be used for predicting numerical salaries, we have chosen to investigate regression trees [8] and ensembles of such models, random forests [7], which have performed well for salary predictions in [39]. We detail on random forests in the subsequent section. These models offer several advantages: Regression trees and random forests are said to be good in handling categorical attributes, missing values, noise and outliers. In addition, they are said to be robust against overfitting, no separate feature-selection, scaling or transformation steps are necessary, and correlated independent variables do not affect the models by much. Furthermore, random forests are one of the most accurate machine-learning methods [5]. [10] and [16] demonstrate this for the related classification problem in large-scale evaluations.

3.3.1 Learning and Applying Random Forests

Random forests [7] are a machine-learning technique for classification or regression that constructs an ensemble of decision or regression trees. The idea behind such ensemble methods is that the prediction accuracy is increased by combining the results from multiple – possibly diverse – models [1, 5, 27, 36]. Frequently, diversity is achieved by introducing randomness into the learning algorithms. The algorithm for random forests applies, besides other modifications, the bagging technique [6] to tree-learning algorithms [33]. Details regarding decision and regression trees can be found in several textbooks, for instance, in [1, 5, 36]. Algorithm 1 describes the general process of learning a random forest *RF* from a training dataset *TS* consisting of n trees. It internally employs an arbitrary decision or regression tree learning algorithm *tree_learner*() without pruning which is modified in order to internally use a small random subset of the independent variables at each split. Hence, random forests add two kinds of randomness to model-building: Firstly, the bootstrapped sampling approach of bagging creates permutated training datasets. Secondly, using random subsets of independent variables leads to more diverse trees. The reason for using random subsets of variables is that otherwise even bagged trees having differently permutated training sets tend to choose the same independent variables at the top level, resulting in relatively similar trees [1]. The random-subset approach attempts to reduce the variance and the correlation of the predictions of the individual

trees, ultimately leading to better predictions than achieved by individual or bagged trees [1, 5, 27].

Algorithm 1 Construction of random forests

Input: training set TS; number of trees n; a tree-learning algorithm *tree_learner*( ) without pruning using a small random subset of independent variables at each split
Output: random forest RF (a set of trees)
1: $RF \leftarrow \emptyset$
2: **for** 1 **to** n **do**
3: $TS' \leftarrow$ a bootstrap sample of TS
4: $RF \leftarrow RF \cup tree\_learner(TS')$
5: **end for**
6: **return** RF

To deduce classifications or numeric predictions (in case of regression) from a random forest, all contained trees are used to predict a data record. The result is than derived by employing a majority-vote strategy (classification) or calculating an average over all n trees (regression).

3.4 An Ensemble of Random-Forest-Regression Models

We now describe our approach for predicting salaries with an ensemble of random-forest-regression models.[4] Our analyses regarding the correlation of independent variables with the salary (see Section 3.1) as well as the experiments in our evaluation (see Experiment 1–3 in Section 4.2) have shown that there is one independent variable that outweighs the others by a considerable margin. This variable is the *profession* of an employee, a categorical attribute having a high cardinality (330 possible distinct values). This is challenging, as categorical features with high cardinality are problematic for tree-based machine-learning approaches. The reason is that the learning algorithms of decision and regression trees are unlikely to determine the best split with this type of data [17, 35]. We propose to solve this problem with an ensemble of random-forest-regression models where we train one random-forest model per possible value of such a categorical feature with high cardinality and feature importance as described in the following. We have chosen this approach as possible alternative strategies such as grouping or clustering the values[5] of the high-cardinality feature would lead to the loss of potentially relevant information of such a highly predictive feature.

[4] While random forests are our main machine-learning technique for salary predictions, regression trees can be used in our approach as an alternative. They are less complex and perform worse than random forests. We compare the prediction performances of random forests versus regression trees in Section 4.2 in detail.

[5] We use such a strategy for the less predictive features *company industry* and *federal state* as described in Section 3.1.

We call a categorical independent variable having a high cardinality (denoted m) and high feature importance P. We denote the distinct values of P $p_1, ..., p_m$. In our case, the *profession* is the independent variable P and the $m = 330$ different professions are $p_1, ..., p_m$. We partition our training dataset TS by assigning all tuples $t \in TS$ having the same value p_i of P to the same partition $TS_{P=p_i}$: $TS = \bigcup TS_{P=p_i}$. We then train one random forest RF_i for each $TS_{P=p_i}$ using Algorithm 1. Correspondingly, to derive predictions, we use P to decide which random forest RF_i to use. Algorithm 2 describes our approach for the construction of an ensemble model EM of random-forest-regression models. It internally calls a function *random_forest_learner*() which learns a random forest as described in Algorithm 1.

Algorithm 2 Construction of an ensemble of random-forest-regression models

Input: training set TS containing a categorical independent variable P having a high cardinality and high feature importance; a random-forest algorithm *random_forest_learner*(), for example, Algorithm 1
Output: ensemble model EM (a set of random-forest-regression models)
1: $EM \leftarrow \emptyset$
2: partition TS into $m = |P|$ partitions $TS_{P=p_i}$ where all tuples have the same value of P p_i
3: **for** $i = 1$ **to** m **do**
4: $RF_i \leftarrow random\_forest\_learner(TS_{P=p_i})$
5: $EM \leftarrow EM \cup RF_i$
6: **end for**
7: **return** EM

In this study, we learn 330 random-forest models, one per profession. Each random forest consists internally of many regression trees. To derive a salary prediction for a specific employee, we use the employee's profession to select the corresponding random-forest model.

4 Evaluation

In this section, we present the evaluation of our approach as described in Section 3.4 on the dataset described in Section 3.1.

4.1 Experimental Setup, Measure of Prediction Accuracy and Baseline

We have implemented our approach in an Apache Spark cluster [42] on standard central processing units (CPUs) using the scikit-learn library [31] for machine learning. For our experiments presented in the following, we have divided our dataset into an 80% training set, a 10% validation set and a 10% test set, which is a standard procedure in machine learning [36]. For all but the last of our experiments, we use the

training set and perform a standard 5-fold cross validation to obtain the experimental results. We then use the validation set for hyper-parameter tuning and use the test set to derive our final results.

We measure the prediction accuracy using the standard mean absolute percentage error (*MAPE*), which has also been used in the related work closest to ours [24]. The *MAPE* is the average of all absolute deviation values of the predictions from the actual values divided by the actual value. As we employ an ensemble of prediction models in many of our experiments, we calculate the average weighted by the number of employees predicted by a model in this situation. The reason why we have chosen the *MAPE* is that we are convinced that it is intuitive and makes more sense from a business perspective for the problem of salary predictions than to use other measures. For example, the probably most popular measure for regression, the mean squared error (*MSE*), is not intuitive as squared Euros or Dollars do not make sense for humans. Further, as some professions have much higher average salaries than others, an accuracy measure using absolute values (for example, the *MSE*) is not a well-enough indicator for the prediction accuracy. For instance, a deviation of 1,000 Euros from the actual salary is a much better prediction for an actual salary of 100,000 Euros than for a 20,000 Euro salary. Therefore, a percentage-based accuracy measure like the *MAPE* is a better choice for our business problem.

To compare our results from the machine-learning models, we define the baseline as follows: The baseline predicts the salary of an employee with the median salary of all employees in that profession while ignoring all further variables. This is a very simple approach, but it simulates the first guess for a salary one would probably have when looking at salary distributions as shown in Figure 1. All models developed in this study are expected to perform better than this baseline. This simple baseline approach already yields a *MAPE* of 20.8%. This is not far off from the *MAPE* of 19.3% published in the related work [24] using linear regression on many variables, a by far more complex approach (but obtained on a different dataset, see Section 5.1).

4.2 Experimental Results

In Figure 2, we present the results of our five experiments. The results show that training individual models per profession and a more extensive outlier handling both are significant steps in improving the predictive performance. We always train and evaluate a regression tree and a random-forest regression model and compare it below. Note that Experiment 1 and 2 can be seen as preliminary experiments to demonstrate the effect of our full approach (as described in Section 3.4) in Experiments 3–5.

Experiment 1: One model for all professions. In our first experiment, we train one model for all professions. The result of the random-forest model is already better than our baseline. The difference is around one percentage point, which is a relatively small improvement. As described in Section 3.4, to better deal with our important

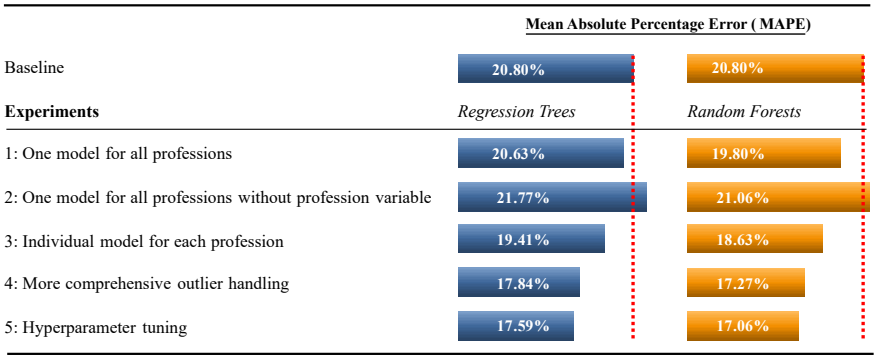


Fig. 2: Mean absolute percentage error (*MAPE*) of the baseline and our experiments

categorical independent variable *profession* and its large number of possible values, we switch to an ensemble approach in Experiment 3.

Experiment 2: One model without the profession. To demonstrate the influence of the independent variable *profession* for salary predictions, we run this experiment, which is the same as Experiment 1, but without using the variable *profession*. The results are more than one percentage point worse than Experiment 1 and even worse than our baseline which makes use of the *profession* only. This shows that this variable is essential and needs to be treated adequately.

Experiment 3: Separate models for each profession. The ensemble approach (random forest) reduces the *MAPE* by more than another percentage point compared to Experiment 1. This is a little more than two percentage points better than our baseline.

Experiment 4: More comprehensive outlier handling. As discussed in Section 3.2, resulting from the fact that our dataset inherently contains some incorrect data points, we deal carefully with the outliers in our dataset. In this experiment, we switch from our conservative approach of outlier removal ($IQRF = 3$) to a less conservative approach ($IQRF = 1.5$). This yields another improvement. The random-forest performance (*MAPE*) improves by one percentage point compared to Experiment 2 and is 3.5 percentage points better than the baseline.

Experiment 5: Hyper-parameter tuning. Regression trees and random forests come with several parameters to control and steer the machine learning process. These should be adopted to the dataset. We try several settings for the three parameters with the highest possible impact [32] in the scikit-learn implementation [31] using the training set for learning and the validation set for evaluation: (1) number of variables randomly sampled as candidates at each split, (2) minimum number of samples required to be at a leaf node and (3) number of trees in the forest. The result of the hyper-parameter tuning yields another slight increase in predictive performance compared to Experiment 4. The resulting *MAPE* of 17.06% is roughly four

percentage points better than the baseline, a relative improvement of 17.8% (random forests).

Finally, we have trained our models with the new parameters on the unified training set (training set and validation set; 90% of all data) and have evaluated the results on the test set (10% of all data). To ensure the model fit, we compared the *MAPE* on the unified training set (16.57% for the regression trees, 16.39% for the random forests) with the *MAPE* on the up to this moment unseen test set (17.60% for the regression trees, 17.08% for the random forests). The differences of 6.2% for the regression trees and 4.2% for the random forests are marginal. Hence, it can be concluded that our models do not overfit, and it can be assumed that the models generalise well to unseen data.

In all experiments, the random forests perform better than the regression trees. This confirms the findings from the literature [5, 10, 16] which have been obtained from classification problems. Our results show that random forests also increase the predictive performance when it comes to regression.

4.3 Runtime

Learning our ensemble of random forests as described in Section 3.4 comes with a considerable computational cost. We have measured runtimes in the range of a few to several hours when learning our 330 random forests. As we have done all experiments on standard central processing units (CPUs), and as random forests are known to benefit from parallel computations in graphical processing units (GPUs), it can be expected that GPUs can speed-up computations considerably. Preliminary experiments of ours with neural networks on CPUs have shown that they perform considerably worse than our random forests in terms of runtime (runtimes more than doubled on the same hardware). This however depends to a large degree on the chosen network topology, and neural networks likewise largely benefit from GPUs. Particularly as salary information is usually not updated more frequently than monthly, spending some hours of computation time each month for model learning seems to be not problematic. Besides model learning, predicting salaries for individual employees can be done faster than in one second, which allows for integration in interactive software.

5 Discussion of the Results and Future Directions

We now discuss the results from the evaluation (Section 4) and give future directions.

5.1 Comparison to Related Work

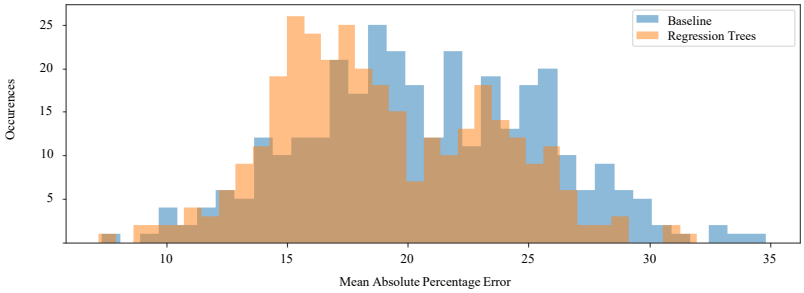
Comparing the mean absolute percentage error (*MAPE*) with values obtained from different datasets is not easy, as the dataset itself and questions of pre-processing – in particular outlier handling – affect the results (as shown in Experiment 4). However, if there are *MAPE* values published, one might obtain a rough idea if the *MAPE* values are in the same magnitude or not.

The authors of [24] publish a *MAPE* of 19.27%, which is a little more than two percentage points higher than our results. However, the authors have used a smaller dataset (roughly factor 5). As we do not know how this dataset was assembled and if and how they have possibly eliminated outliers, a direct comparison is not possible. The values nevertheless suggest that using random-forest regression on a large dataset of salary data is worth the computational effort in comparison to the results from the simpler linear regression model on a smaller dataset. The other studies discussed in the related work do not publish *MAPE* values or have employed classification algorithms to predict classes of salaries. This makes it impossible to obtain meaningful error percentages. Furthermore, the same problems regarding datasets and pre-processing apply. From all results published, we got the impression that our *MAPE* of 17.1% is a respectable result, and that it will be hard to obtain much better results using a data-driven approach without incorporating further data than the data from payslips as investigated in our study.

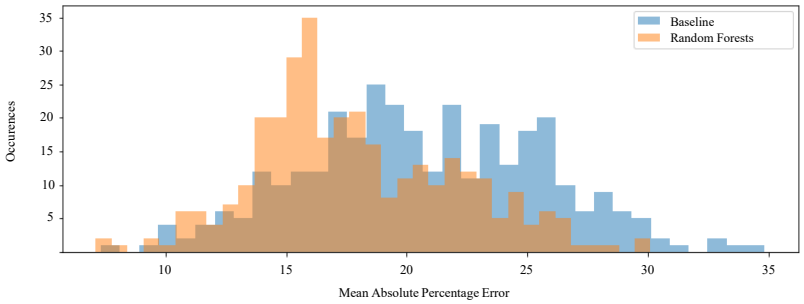
5.2 Analysis of the Regression Trees and Random-Forest Results

In Section 4.2, we have presented the (average) *MAPE* of 17.1% for our random-forest-regression ensemble. Following up on the promising overall performance of our approach, we are interested in how the individual models perform (one per profession).

Figure 3 presents histograms of the distributions of the *MAPE* values for the baseline and our regression trees (Figure 3a) and the baseline and our random forests (Figure 3b). The latter illustrates that both the distributions of the baseline and the random forests have a relatively large range from roughly 10% to 30% (baseline) or 35% (random forests), while the distribution of the random forest *MAPE* values is clearly shifted to the left. As the average *MAPE* of the random forest ensemble approach is lower, this left shift is plausible. It can be an indication that the input parameters and data used in our models are more useful predictors for the salaries of some professions than they are for others. For the models yielding comparably weak prediction performances, factors not reflected in the data, but influencing the salary of those professions in reality, may be the cause. The wide *MAPE* value range can be observed in the regression trees, random forests and baseline approach alike. We now discuss possible reasons for the characteristic that not all jobs can be predicted with the same *MAPE*.



(a) Regression trees



(b) Random forests

Fig. 3: Distribution of the mean absolute percentage error (*MAPE*), the vertical axis shows the number of models having a certain *MAPE* value

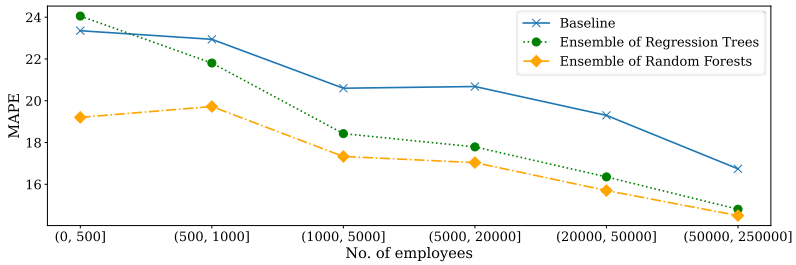


Fig. 4: The mean absolute percentage error (*MAPE*) versus the number of employees per model (per profession)

Figure 4 displays the *MAPE* in relationship to the number of employees per model. In general, professions with fewer employees have larger *MAPE* values, which seems to be intuitive as there are fewer training examples. Figure 5 displays the *MAPE* in relationship to the median salary. Here we can clearly see the best performance in the lower incomes. High incomes of 40,000 € per year and more are difficult to

predict (the general median salary in Germany is around 39,000 € [25]). The reason is probably that many of the lower-paid professions are kind of more standardised, and the salaries do not vary much. Probably, there are more collective agreements in the professions where lower salaries are paid. The variation of salaries is a lot higher in the professions with a higher median salary, leading to the fact that they are harder to predict.

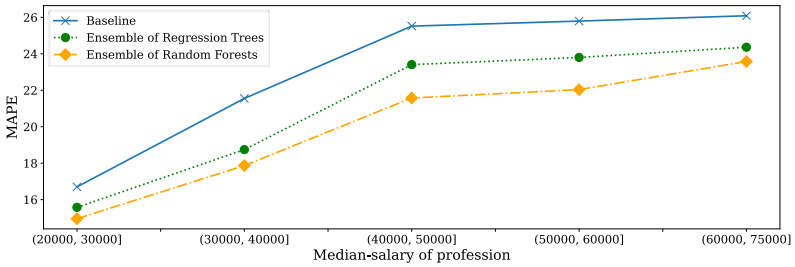


Fig. 5: The mean absolute percentage error (*MAPE*) versus the median salary (dependent variable *annual gross income*)

While most salaries were predicted more accurately by the random-forest ensemble than with the baseline model (in 309 out of 330 professions), we found that the salaries of 21 professions were on average predicted more accurately by the baseline model using nothing but the median of the profession. While there is no significant influence of the number of employees per profession on the *MAPE* of the respective model, the expertise level of those professions may be an explaining factor: Eleven out of the mentioned 21 professions are "helper activities" (lowest expertise level) and seven are of "professional expertise" (second lowest expertise level, according to the German Classification of Occupations [23]), having both a below average income. Probably, these employees have quite differing backgrounds and are thus hard to predict.

5.3 Feature Importances

In order to better understand our random-forest models, we have conducted permutation-feature-importance analyses [18] using the implementation of scikit-learn [31]. Due to the high computational costs of these analyses, we have done this for the models of a small sample of professions.

Figures 6 and 7 contain the results of two permutation-feature-importance analyses of two random-forest models (professions). For these two models, the *company size* is by far the most important independent variable, and the *federal state* is important in both models. For the *company industry*, the importance is high in Figure 6

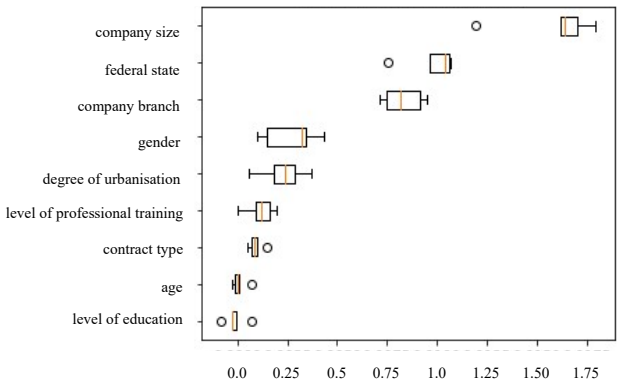


Fig. 6: Box plot of the permutation feature importances in percentage points of the random-forest regression model representing the profession 'textile sewer'

and low in Figure 7. One explanation for the low value for the computer scientists (Figure 7) is, that many of them work in the same industry. Hence, the industry does not say much about the salary. The for the *age* and the *level of professional training* it is the other way round: Both variables are important for the computer scientists (Figure 7) where the more experienced and better educated employees earn more. They are less important for the textile sewers (Figure 6) where the spread of salaries is not as big. The two examples from Figures 6 and 7 illustrate that the feature importances vary quite largely in the different professions. Taking more than these two examples into account, it is observable that some variables – in particular the *level of education* and the *gender* – have a comparably low influences on the salary.

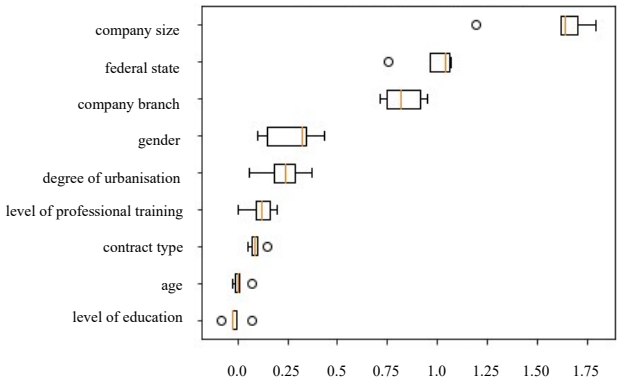


Fig. 7: Box plot of the permutation feature importances in percentage points of the random-forest regression model representing the profession 'computer scientist'

5.4 Usage of the Variable *gender* and Further Improvements

In this study, we make use of the variable *gender*. This makes sense from the perspective that it is known that there is a gender pay gap [15] which we have confirmed in our analyses (see Section 3.1). It might otherwise not be desirable to use this variable in a real product, as those should not contribute to increase the gender pay gap. In other settings, such as the "Gehaltsvergleich BETA" of the German Federal Office of Statistics [21], the gender is used as well. The rationale is that its provider has the mission to create transparency of salaries including the gender. As we have observed the – maybe surprising – low variable importance of the *gender* in the permutation-feature-importance analyses in Section 5.3, we have conducted additional experiments: We have investigated the influence of the variable *gender* on the predictions for all professions by removing it from our dataset. This decreases the error values from the regression trees and the random forests in Experiment 5 (see Section 4.2) by less than half a percentage point. Hence, our approach can work without the variable *gender* as it affects the predictive performance not by much.

To predict salaries more accurately, it would be desirable to incorporate the work experience as another independent variable, as more experienced employees usually have a higher salary. When working with payroll data, this information is typically not available. Therefore, we have used the variable *age* as a surrogate and will in the future additionally incorporate the period of employment with the same employer. This might slightly improve the predictive performance as it approximates the work experience. However, it would be desirable to capture the work-experience information when hiring new employees and to keep it in the payroll software.

Our further ideas for performance improvements focus on pre-processing and data engineering. One idea is to use more complex methods in the outlier-handling process. For instance, clustering or anomaly detection. Another idea is to cluster the different professions into groups of professions. For example, software developers and programmers are two distinct professions in the classification of professions [23], having very similar characteristics (for instance, regarding salary and level of professional training). We plan to apply clustering techniques to merge such similar professions into clusters to capture more professions, to have more data per model to learn from and to possibly reduce the number of individual models. We see this as a promising approach as we could already show that the number of employees in the training data of the individual models correlates negatively with the prediction error (see Section 5.2).

Other data-engineering ideas of ours concern the regional information: We plan to further enrich our dataset with external data such as the population or the purchasing-power index of the place of employment. It could also be interesting to calculate a regional salary index based on our dataset and to use this as additional information to characterise the place of employment. These ideas aim at capturing the regional information better than with the federal state and the degree of urbanisation alone. This characterisation might be necessary for improving the predictive performance, as we know that the region has a significant influence on the salary, and as using zip

codes or names of municipalities directly in the models might not work due to the extremely large number of possible values of these categorical variables.

5.5 Assessment of Prediction Quality for Real-World Software Products and Questions of Deployment

Even if further improvements as described before including enrichments with external data will likely lead to improved values of the mean absolute percentage error (*MAPE*), we do not expect these improvements to be large without having new data sources describing the employees. One promising new source to improve predictive performance could be the results of systematic employee ratings if companies have implemented respective evaluation and assessment processes, but this would be hard to compare between companies and would raise privacy questions. The assumption that predictions might not be improved largely leads to the question whether a *MAPE* of around 17.1%, which is roughly four percentage points better than the simple baseline, justifies a rather complex and computationally expensive machine-learning effort and will be accepted by customers. The question becomes even more severe, when we improve our baseline approach, for instance by using median values of the professions in the same region.

Deploying and running machine learning in productive environments is still a rather complicated endeavour [12]. It includes monitoring and detection of concept drift [30], model management and re-training [3] and questions of data privacy. Attackers might, for instance, use an implementation of our approach to reconstruct the original learning data. This needs to be prevented by data-security measures such as limiting the number of requests that can be sent to the model API or by advanced privacy technology [2, 29]. Solutions in this domain could be differential privacy [13] – in particular subsampling [40] – or privacy-guided training [26].

Even if results may be improved by some degree, customers might still assume that a *MAPE* of, say, 15% or more is relatively high. It is therefore important to explain to (potential) customers of a product implementing our approach, how the *MAPE* values are calculated. First, *MAPE* values are strongly affected by incorrect data, and we must assume that not all wrong inputs regarding, for example, the profession or the weekly working hours, can be captured by outlier handling. Second, we have derived all *MAPE* values by predicting the salaries of real employees which we have not used for learning. In consequence, large deviations do not necessarily mean that the model is wrong, but might indicate that there are employees being underpaid or overpaid, which likely happens in real world. Besides this, from a customer perspective, predictions based on regression are more useful than having nothing but the median of a distribution of a rather large population (as in [11], see Figure 1). Furthermore, the results of our random forests outperform the baseline approaches in 309 out of 330 professions (93.6%).

Taking all arguments in this section into account, we conclude that our approach with the predictive performance described is valuable for the customers when inte-

grated into suitable software tools and justifies the efforts described. One limitation to our approach might be the applicability to large companies and possibly certain under-represented professions, as our dataset focusses on small and medium-size companies. However, if a more complete dataset than ours is available for learning, we see no obstacles in applying our approach to it.

6 Conclusion

In this study, we have investigated an ensemble-of-ensembles approach for predicting salaries based on salary data, where we have learned one random-forest-regression model per profession. In our comprehensive evaluation on a large real dataset, we have achieved a mean absolute percentage error (*MAPE*) of 17.1%. This is an improvement of 17.8% compared to our baseline, and it is two percentage points better than the results published of the related work (on a different dataset). Thus, we have shown that sophisticated machine-learning models are suitable to predict salaries on a wide range of professions and employees and that our ensemble-of-ensembles approach clearly outperforms other approaches, such as simply setting the prediction to the median per profession, or using linear regression. Our approach can be integrated into salary-analysis solutions to help HR managers and tax consultants to determine market prices for current and prospective employees.

Acknowledgements

We thank Professor Dr. Sven Overhage for his ongoing support when conducting this research.

References

1. C. C. Aggarwal. *Data Mining: The Textbook*. Springer, 2015.
2. M. Al-Rubaie and J. M. Chang. Privacy-Preserving Machine Learning: Threats and Solutions. *IEEE Security & Privacy*, 17(2):49–58, 2019.
3. E. Ameisen. *Building Machine Learning Powered Applications*. O'Reilly UK Ltd., 2020.
4. D. A. Barbezat and J. W. Hughes. Salary Structure Effects and the Gender Pay Gap in Academia. *Research in Higher Education*, 46(6):621–640, 2005.
5. M. R. Berthold, C. Borgelt, F. Höppner, and F. Klawonn. *Guide to Intelligent Data Analysis: How to Intelligently Make Sense of Real Data*, volume 42 of *Texts in Computer Science*. Springer, 2010.
6. L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, aug 1996.
7. L. Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.
8. L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth International Group, 1984.

9. S. Chakraborti. A Comparative Study of Performances of Various Classification Algorithms for Predicting Salary Classes of Employees. *International Journal of Computer Science and Information Technologies*, 5(2):1964–1972, 2014.
10. R. Couronné, P. Probst, and A.-L. Boulesteix. Random Forest versus Logistic Regression: A Large-Scale Benchmark Experiment. *BMC Bioinformatics*, 19(1), 2018.
11. DATEV eG. Personal-Benchmark online. `https://datev.de/web/de/mydatev/online-anwendungen/datev-personal-benchmark-online/`. Accessed: 2022-01-23.
12. T. Davenport and K. Malone. Deployment as a Critical Business Data Science Discipline. *Harvard Data Science Review*, Issue 3.1, Winter 2021, 2021.
13. C. Dwork. Differential Privacy. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, 2006.
14. Eurostat, European Commission. Degree of Urbanisation. `https://ec.europa.eu/eurostat/web/degree-of-urbanisation/methodology`. Accessed: 2022-01-23.
15. Eurostat, European Commission. Gender Pay Gap Statistics. `https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Gender_pay_gap_statistics`. Accessed: 2022-01-23.
16. M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim. Do We Need Hundreds of Classifiers to Solve Real World Classification Problems? *Journal of Machine Learning Research*, 15:3133–3181, 2014.
17. J. J. Filho and J. Wainer. Using a Hierarchical Bayesian Model to Handle High Cardinality Attributes with Relevant Interactions in a Classification Problem. In *International Joint Conference on Artifical Intelligence*, 2007.
18. A. Fisher, C. Rudin, and F. Dominici. All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously. *Journal of Machine Learning Research*, 20(177):1–81, 2019.
19. German Federal Employment Agency. Entgeltatlas. `https://con.arbeitsagentur.de/prod/entgeltatlas/`. Accessed: 2022-01-23.
20. German Federal Employment Agency. Occupation Codes for Statistical Messages in Germany. `https://www.arbeitsagentur.de/betriebsnummern-service/taetigkeitsschluessel`. Accessed: 2022-01-23.
21. German Federal Office of Statistics. Gehaltsvergleich BETA. `https://service.destatis.de/DE/gehaltsvergleich/`. Accessed: 2022-01-23.
22. German Federal Office of Statistics. German Classification of Economic Activities 2008. `https://www.destatis.de/DE/Methoden/Klassifikationen/Gueter-Wirtschaftsklassifikationen/Downloads/klassifikation-wz-2008-englisch.html`. Accessed: 2022-01-23.
23. German Federal Office of Statistics. German Classification of Occupations 2010. `https://statistik.arbeitsagentur.de/DE/Navigation/Grundlagen/Klassifikationen/Klassifikation-der-Berufe/KldB2010/Arbeitshilfen/EnglischeKldB2010/KldBEnglischl-Nav.html`. Accessed: 2022-01-23.
24. German Federal Office of Statistics. Interaktiver Gehaltsvergleich. `https://www.destatis.de/DE/Service/Statistik-Visualisiert/Gehaltsvergleich/Methoden/Methodenbericht.pdf`. Accessed: 2022-01-24.
25. German Pension Insurance. Durchschnittseinkommen. `https://www.deutsche-rentenversicherung.de/SharedDocs/Glossareintraege/DE/D/durchschnittseinkommen.html`. Accessed: 2022-01-23.
26. A. Goldsteen, G. Ezov, and A. Farkash. Reducing Risk of Model Inversion Using Privacy-Guided Training. *Computing Research Repository (CoRR)*, abs/2006.15877, June 2020.
27. L. I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. John Wiley & Sons, 2004.
28. E. Limpert, W. A. Stahel, and M. Abbt. Log-Normal Distributions across the Sciences: Keys and Clues. *BioScience*, 51(5):341–352, 2001.
29. X. Liu, L. Xie, Y. Wang, J. Zou, J. Xiong, Z. Ying, and A. V. Vasilakos. Privacy and Security Issues in Deep Learning: A Survey. *IEEE Access*, 9:4566–4593, 2021.

30. J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang. Learning under Concept Drift: A Review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12):2346–2363, Dec. 2019.
31. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011.
32. P. Probst, M. N. Wright, and A.-L. Boulesteix. Hyperparameters and Tuning Strategies for Random Forest. *WIREs Data Mining and Knowledge Discovery*, 9(3):e1301, 2019.
33. J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
34. R. Rahim, T. Husni, Yurniwati, and Desyetti. The Relation Between Cash Compensation of Banking Executives, Charter Value, Capital Requirements and Risk Taking. *International Journal of Business*, 25(5):399–420, 2020.
35. Rakesh Ravi. One-Hot Encoding is making your Tree-Based Ensembles worse, here's why? `https://bit.ly/3Fg81tS`. Published inTowards Data Science, accessed: 2022-05-04.
36. S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach (4th Edition)*. Pearson, 2020.
37. SAS Institute Inc. The SURVEYREG Procedure. In *SAS/STAT 13.1 User's Guide*, chapter 98, pages 8353–8442. SAS Institute Inc., Dec. 2013.
38. D. S. Sisodia, S. Vishwakarma, and A. Pujahari. Evaluation of Machine Learning Models for Employee Churn Prediction. In *International Conference on Inventive Computing and Informatics (ICICI)*, 2017.
39. P. Viroonluecha and T. Kaewkiriya. Salary Predictor System for Thailand Labour Workforce using Deep Learning. In *International Symposium on Communications and Information Technologies (ISCIT)*, 2018.
40. Y.-X. Wang, B. Balle, and S. P. Kasiviswanathan. Subsampled Renyi Differential Privacy and Analytical Moments Accountant. *Journal of Machine Learning Research*, 89:1226–1235, 2019.
41. I. O. Yigit and H. Shourabizadeh. An Approach for Predicting Employee Churn by Using Data Mining. In *International Artificial Intelligence and Data Processing Symposium (IDAP)*, 2017.
42. M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin, A. Ghodsi, J. Gonzalez, S. Shenker, and I. Stoica. Apache Spark. *Communications of the ACM*, 59(11):56–65, 2016.
43. C. Zhang and Y. Liu. The Salary of Physicians in Chinese Public Tertiary Hospitals: A National Cross-Sectional and Follow-Up Study. *BMC Health Services Research*, 18(661), 2018.