Predicting Candidate Uptake For Online Vacancies

Corné de Ruijt[†‡], Sandjai Bhulai [†], Han Rusman[‡] and Leon Willemsens[‡]

[†]Faculty of Sciences
Vrije Universiteit
Amsterdam, the Netherlands
Email: s.bhulai@vu.nl
[‡]Endouble
The Netherlands
Email: {Corne, Han, Leon}@endouble.com

***Abstract*—The Internet has substantially changed how organizations market their vacancies and how job seekers look for a job. Although this has many benefits such as simplifying the communication, it can also cause problems. Some vacancies are obtaining more applications than can be handled by the recruitment department, while other vacancies may remain unfulfilled for a long time. Data analysis might reveal insights into what strategies are effective to solve these problems. To analyze these problems we therefore consider the predictability of the number of applications per vacancy per week, and to which extend this can be controlled using online marketing campaigns. After testing the predictive quality of several machine learning methods on a data set from a large Dutch organization we found that a Random Forest model gives the best predictions. Although these predictions provide insights into what recruiters and hiring managers can expect when publishing a vacancy, the error of these predictions can be quite large. Also, although the effect of online marketing campaigns on the number of applications is significant, predicting the effect from historic data causes problems due to collinearity and bias in the usage of these campaigns: a campaign is also a response to a small number of applicants who responded to the vacancy. Nevertheless, these predictions are insightful for both recruiters and hiring manager to manage their expectations when publishing a vacancy.**

Keywords*–*Recruitment analytics; HR analytics

I. Introduction

The internet revolution has substantially changed how job seekers look for a job and how organizations attempt to attract job seekers [1]. Already in 2003, 94% of the global Fortune 500 companies were using a corporate recruitment website to attract job seekers [2], and online sources such as social media, online professional networks, and company websites are being used for effective employer branding [3]. Furthermore, the percentage of job seekers who are using the internet is growing steadily [4].

Advantages of using corporate recruitment websites have been discussed in previous studies, showing benefits including cost effectiveness, speeding up the hiring process, and ease of use both for recruiters and job seekers [5], [6], [7]. There is, however, yet another benefit of using corporate recruitment websites which has not been explored by previous research: it enables tracking the behavior of job seekers on the website using e-commerce software. By tracking this job seeker behavior, recruitment departments can obtain valuable insights on how to attract or repulse applications. This might lead to strategies for reducing recruitment lead time and cost.

To take a first step into exploring how vacancies attract job seekers and how this might be controlled, this study considers the predictability of the number of job seekers that will apply to an online vacancy per week. This metric is referred to as the application rate. In order to predict this metric, multiple machine learning techniques including Random Forest, Support Vector Regression, and Artificial Neural Networks were applied. The data used to predict the application rate included characteristics of the vacancy such as work location, required education level, and job title. Furthermore, also data describing whether the vacancy was used in online marketing campaigns such as Google Adwords, other vacancies on the website which might compete with the vacancy, and time related attributes such as the current recruitment lead time and application rates in weeks prior to the predicting period were used.

This paper has the following structure, Section II provides an overview of previous literature on the effectiveness of online recruitment websites will be given. Section III will discuss how data was obtained and prepared for analysis. In Section IV the findings from preliminary data analysis will be discussed which affects the choice of predictive models. Section V will give an overview of the methods that were used to predict the application rate. Finally, Section VI provides an overview of the predictive quality of these methods along with its implications.

II. Related work

The ability of corporate recruitment websites to attract job seekers has been investigated in multiple studies, often by sending questionnaires to either job seekers or employers. The results of these studies differ: some show the potential in terms of cost effectiveness, reducing recruitment leadtimes, and ease of use for both recruiters and job seekers [5], [6], [7]. Other studies however show a more modest perception: Brown [8] found that 75% of all job seekers find recruitment websites too complicated. This perception is shared by Maurer and Liu [9] who identified management of potential information excess on corporate recruitment websites as one of the key design issues for e-recruitment managers. Besides the excessive information organizations might send to potential job seekers, the opposite also holds. Vacancies might receive a large number of applications, including many unsuitable ones. Parry and Tyson [10] found that this is one of the reasons why a quarter of the organizations they examined who were using internet recruitment methods found it unsuccessful.

Copyright (c) IARIA, 2016. ISBN: 978-1-61208-510-4

Data mining can play a role in managing the information spread by both the employer and job seeker. In particular, it can be used to investigate the relationship between recruitment efforts and recruitment outcomes. These relationships can be used to control the quality and quantity of applicants, and the quality of the employer's brand.

Previous research has not paid much attention to how data mining could be applied to manage the information spread by employers and job seekers apart from application selection [11], and resume parsing [12]. Although these methods automate part of the recruitment job, thereby enabling recruiters to handle a large number of applications, being able to control the quality and quantity of applicants would also decrease the workload of recruiters. Furthermore, fewer but better qualified applicants also means fewer rejections, which is beneficial for both job seekers and employers.

III. Data gathering and preparation

A. Data gathering

To investigate whether the number of applicants who apply to a vacancy can accurately be predicted and controlled data was gathered from a large Dutch company which employs over 30,000 people and has on average 150 vacancies on its corporate recruitment website.

Data was gathered from three systems: first from an application tracking system (ATS) in which vacancy characteristics are stored such as work location, required education level, and working hours. Second, data was gathered from the corporate recruitment website's Google Analytics account. In particular, how many job seekers visited the corporate recruitment website per week, and how frequent job seekers followed different paths from the website's landing page to the application submit page (the webpage visited after having submitted an application). Also, Google Analytics is capable to keep track of whether job seekers visited the website via a paid hyperlink which was part of an online marketing campaign. This data was used to determine which vacancies had been used in online marketing campaigns. Third, the number of weekly tweets the recruitment department published via their recruitment Twitter account was gathered, along with whether certain vacancies were referred to in a tweet via a hyperlink.

Combining these three data sources gives per vacancy v, per time period t (in weeks) the vacancy characteristics of v, whether v was used in certain online marketing campaigns, and how many job seekers navigated from the landing page to the vacancy's submit page during time period t. This dataset was extended with time related data such as the recruitment lead time at time t, and application rates of a vacancy in weeks prior to week t.

The data set was split into a test- and training set. The training set contained all values between 2013-08-26 and 2015-09-31, whereas the test set contained all values between 2015-10-01 and 2015-12-31. This split was chosen for two reasons: first, at the time of splitting the data set there was no knowledge of possible time dependency in the data. If the application rate would include this time dependency then validating the predictive model on the last period of the total data set would produce the most realistic evaluation. Second, three months is the maximum period for which it is safe to assume that the vacancy portfolio over that period is known.

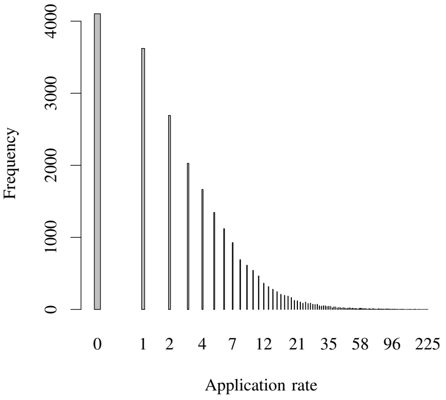

Figure 1. Histogram application rate

B. Data preparation

To improve the quality of the data set multiple operations were performed. Attributes related to work location and job title contained many possible categorical values, which was not practical for analysis. To reduce the number of categorical values the locations were clustered based on similarities in their application rate probability density. These probability densities were clustered using the K-means clustering algorithm by Hartigan and Wong [13]. To find an appropriate number of clusters the Akaike Information Criteria (AIC) was used, which was computed for $K = 1, \ldots, 10$ clusters. If a cluster had fewer than 100 observations the observations were assigned to the cluster closest to the overall mean application rate. Besides location attributes the job title had even more unique values, which made the usage of the probability density unpractical. As an alternative similar job titles were identified and clustered manually.

In order to identify attributes having a small variance, the frequency cut off from the *nearZeroVar* function of the caret package was used [14]. Since all predictors are either binary, categorical or discrete it was possible to apply this procedure on all predictors. Let $N_{i,j}$ be the frequency of a value i of category j. Furthermore, let $N_{(l),j}$ be the lth order statistic of $N_{1,j}, \ldots N_{n,j}$, then we have frequency ratio: $F_j = \frac{N_{(n),j}}{N_{(n-1),j}}$. Thus F_j gives the ratio of the most frequent and second most frequent value of attribute j. Attributes were removed from the data set if $F_j > 19$.

During the last data preparation step categorical attributes were dummified into binary vectors. The predictor values x_{ij} were normalized using $\tilde{x}_{ij} = \frac{x_{ij} - \bar{x}_j}{s(x_j)}$. Here $\bar{x}_j$ and $s(x_j)$ are the mean and standard deviation over the values of attribute j respectively.

IV. Exploratory data analysis

A. The application rate

When considering possible probability distributions of the application rate a Poisson distribution would come first to mind. However, as Fig. 1 suggests, the Poisson distribution does not seem to fit the data well: the application rate's distribution is more zero inflated and overdispersed than a Poisson distribution. Dependent on the nature of the vacancy, a log-normal or negative binomial distribution is more appropriate. The distribution also confirms previous research stating that some vacancies can attract a large number of applications [10]. In fact, 10% of the rates accounts for 53% of all applications.

B. The total number of applications and sessions

Besides considering the distribution of the application rate also the predictability of the total number of applications per week was considered. To predict these metrics the structure of the vacancy portfolio and the used online marketing campaigns were used as predictors. This analysis might already give an indication of how online marketing campaigns can affect both the traffic to the website and the number of applications. Furthermore, if the residuals of this model would be highly correlated this could be an indication of time dependency, which would effect the selection of predictive models for the application rate.

To predict the number of applications and sessions per week, the status of the online vacancy portfolio and the number of vacancies subject to certain online marketing campaigns were used as predictors. The status of the vacancy portfolio was determined by counting the number of online vacancies having certain characteristics such as the same location, work area, job description, and required education level. A linear regression model was used to determine the effect of the online vacancy portfolio and online marketing campaigns on the total number of sessions and applications. This linear regression model did not include any interaction effects. Although also more sophisticated methods can be applied, the number of observations was relatively small compared to the number of predictors. Therefore, using more sophisticated methods was likely to cause overfitting. A backwards AIC algorithm was used to include only those predictors having the most predictive value.

Applying these methods gives an R^2 value of 0.68 and 0.56 when predicting the number of sessions and number of applications respectively. The models also indicate that it is difficult to determine the exact effect of online marketing campaigns on the total number of weekly sessions and applications. Although the marketing campaigns are significant, the campaigns are frequently used in combination with each other which makes it difficult to distinguish the effect of a single campaign.

This can easily be seen if we compute the condition indices and variance decomposition proportions as proposed by [15]. If we add up the variance decomposition proportions obtained from the number of vacancies subject to Facebook, Indeed and Google campaigns over the largest condition indices (85, 102 and 128 resp.), this sum becomes 0.692, 0.797 and 0.91 respectively, which are larger than the threshold value of 0.5. A possible remedy for this collinearity is to add more characteristics of the campaigns to the data set, such as the profiles used in a Facebook campaign. This was however not considered in this study.

Besides collinearity, an increase in online marketing campaigns can also be a response to a small number of applications, which makes the estimated effect of online marketing campaigns on the number of session and applicants biased.

When considering the residuals of the models predicting the number of weekly sessions and applications, a Box-Pierce test showed that these residuals were correlated. However, when examining the autocorrelation and partial autocorrelation functions this correlation turns out to be small: both the auto correlation and partial autocorrelation show a maximum absolute correlation of 0.21, at lag 2 and 1 respectively. Therefore, for simplicity, it was found acceptable to assume that the residuals were uncorrelated. As a result it was assumed that the total number of applications per week is independent of the date of the measurement.

C. Best sources

Also the relationship between the number of sessions originating from different websites via different devices and the number of weekly applications was considered. Again a linear regression model was used to avoid overfitting. Since visitors to the corporate recruitment website can originate from many different sources only the top four sources causing most traffic were considered, whereas smaller sources were combined in an ‘other’ category. Interestingly, the source device combinations causing most traffic to the website did not produce most applications. Where visitors originating from Google on a desktop produced most traffic to the website, changes in direct traffic on either desktop, mobile or tablet and traffic from the corporate website were the main drivers for changes in the number of weekly applications.

V. Methods

A. Method selection

To determine which methods would be most suited to predict the application rate a number of considerations were taken. First, since the application rate is count data its prediction is considered as a regression problem. Second, exploratory data analysis found that the data is more zero inflated and overdispersed than a Poisson distribution. Therefore, predictive models which incorporate zero inflation and overdispersion are preferred. Third, during exploratory data analysis it was found that when predicting the total number of applications per week the residuals of this model are only slightly correlated. As a result it was assumed that the total number of applications per week is independent of the date of the measurement, though it still can be dependent on other time indicators such as the current recruitment lead time.

Fourth, the data set still contained a large number of attributes, some of which might not be useful for the predictive model. To reduce the number of attributes, methods which included variable selection were preferred. Fifth, since a grid search was applied to find good model parameters, methods which were able to produce good results within reasonable time were preferred (i.e., methods that took more than 1 hour to compute a single predictive model using a 1.6 GHz dual-core Intel Core i5 processor were disregarded). Sixth, methods which have been applied successfully in other regression application were preferred.

Using these criteria seven methods were identified: Linear elastic net, Poisson elastic net, Tweedie elastic net, Classification And Regression Trees (CART), Random Forest, Support Vector Regression (SVR), and Artificial Neural Networks (ANN).

Copyright (c) IARIA, 2016. ISBN: 978-1-61208-510-4

B. Method overview

1) Linear elastic net: Linear elastic net is a method which attempts to minimize the sum of squared error plus a linear combination of the lasso and Ridge penalty. Let $PSSE(\lambda, \beta, \alpha)$ be the penalized sum of squared error with λ the weight of the penalty. α indicates to which extend either the Ridge or lasso penalty is taken into account, and β is the effect vector to be estimated. $PSSE(\lambda, \beta, \alpha)$ is given by:

$$\begin{aligned} PSSE(\lambda, \boldsymbol{\beta}, \alpha) = & \tfrac{1}{2N} \textstyle\sum_{i=1}^{N} (y_i - \beta_0 - x_i^T \boldsymbol{\beta})^2 \\ & + \lambda \left[(1-\alpha)\tfrac{1}{2} ||\boldsymbol{\beta}||^2 + \alpha ||\boldsymbol{\beta}||_1 \right] \end{aligned} \tag{1}$$

To minimize (1) the glmnet R package was utilized, which applies a coordinate descent algorithm to estimate β [16]. To determine good values for λ and α a grid search was applied. For λ, $K = 100$ uniformly spread values between $\lambda_{max} = \frac{max_l |\langle x_l, y \rangle|}{N\alpha}$ and $\lambda_{min} = \epsilon \lambda_{max}$ were used. To find a good value for α, values from 0 up to 1 with increasing steps of 0.2 were used.

2) Poisson elastic net: Poisson elastic net is a combination of a generalized linear regression model and elastic net using the link function $g(\mu_i) = log(\mu_i)$, where $\mu_i = \mathbb{E}(y_i|\mathbf{x}_i)$. Instead of the sum of squared error the log-likelihood is used to estimate $\boldsymbol{\beta}$. Let PLL be the penalized log-likelihood, then $\boldsymbol{\beta}$ is found by maximizing (2).

$$\begin{aligned} PLL(\lambda, \boldsymbol{\beta}, \alpha) = & \tfrac{1}{2N} \textstyle\sum_{i=1}^{N} \left[y_i \mathbf{x}_i^T \boldsymbol{\beta} - \exp\left(\mathbf{x}_i^T \boldsymbol{\beta} \right) \right] \\ & - \lambda \left[(1-\alpha)\tfrac{1}{2} ||\boldsymbol{\beta}||^2 + \alpha ||\boldsymbol{\beta}||_1 \right] \end{aligned} \tag{2}$$

To maximize (2), again the glmnet R package was used. In case of Poisson regression, glmnet iteratively creates a second order Taylor expansion of (2) without the penalty, using current estimates for $\boldsymbol{\beta}$. This Taylor expansion is then used in a coordinate descent algorithm to update $\boldsymbol{\beta}$ [16], [17]. To find appropriate values for λ and α the same grid search as in linear elastic net was applied.

3) Tweedie elastic net: To incorporate the fact that the application rate is more zero inflated and overdispersed than a Poisson distribution the Tweedie compound Poisson model was used. The Tweedie compound Poisson model can be represented by $Y = \sum_{i=1}^{n} X_i$, where Y is the responds vector, n a Poisson random variable, and X_i are i.i.d. Gamma distributed with parameters α and γ. The penalized negative log-likelihood is given by (3) [18].

$$\begin{aligned} PLL(\lambda, \boldsymbol{\beta}, \alpha) = & \textstyle\sum_{i=1}^{n} \left[\frac{y_i \exp\left[-(\rho-1)(\mathbf{x}_i^T \beta) \right]}{\rho-1} + \frac{\exp\left[(2-\rho)(\mathbf{x}_i^T \beta) \right]}{2-\rho} \right] \\ & + \lambda \left[(1-\alpha)\tfrac{1}{2} ||\boldsymbol{\beta}||^2 + \alpha ||\boldsymbol{\beta}||_1 \right] \end{aligned} \tag{3}$$

To minimize (3), the HDTweedie R package was used. This method applies an iterative reweighted least squares (IRLS) algorithm combined with a blockwise majorization descent (BMD) [18]. To find appropriate values for λ the standard procedure from HDTweedie was used, which first computes λ_{max} such that $\beta = 0$, and then sets $\lambda_{min} = 0.001\lambda_{max}$. The other $m-2$ values for λ are found by projecting them uniformly on a log-scale on the range $[\lambda_{min}, \lambda_{max}]$. For α the values from 0.1 to 0.9 with an increase of 0.2 were used. For ρ we used $\rho = 1.5$.

4) Classification And Regression Trees (CART): To construct a regression tree the rpart implementation in R was used [19]. This implementation first constructs a binary tree by maximizing in each node $SS_T - (SS_R + SS_L)$, where SS_T is the sum of squared error of the entire tree, and SS_R and SS_L are the sum of squared errors of the left and right branch respectively. The tree constructions stops when further splits would violate a constraint on the minimum number of observations in each node.

Second, the constructed tree is split into m sub-trees. Let $R(T)$ be the risk of tree T, which is the sum of squared error in the terminal nodes of T. CART computes the risk of each sub-tree tree, which is defined by $R_\alpha(T) = R(T) + \alpha|T|$ using K-fold cross validation. The term $\alpha|T|$ is an additional penalty on the size of the tree. The final tree is the sub-tree which minimizes the average sum of squared error over the K-fold cross validation. The method described here is referred to as the “ANOVA” method.

Alternatively, rpart also has the option to maximize the deviance $D_T - (D_L + D_R)$, where D is the within node deviance assuming that the response originates from a Poisson distribution. To find an appropriate value for α a grid search was applied using $\alpha \in \{0.001, 0.01, 0.1, 0.3\}$, both for the ANOVA and Poisson models.

5) Random Forest: A Random Forest model was produced using the RandomForest package in R [20]. Random Forest constructs T unpruned regression trees T_i, where in each split only d randomly chosen predictors are considered. A prediction $\hat{y}_i$ is then created by $\hat{y}_i = \frac{1}{T}\sum_{i=1}^{T} T_i(x)$, thus the average over all trees. To find the appropriate number of trees a grid search was applied using 50, 100 and 500 trees. Furthermore, at each split 61 randomly sampled attributes were considered.

6) Support Vector Regression: Support Vector Regression is the regression alternative of Support Vector Machines. Given the linear regression problem: $y_i = \mathbf{w}^T \mathbf{x}_i + b + \epsilon$, SVR attempts to find the flattest hyperplane $\mathbf{w}^T \mathbf{x}_i + b$ such that for all data points $i = 1, \ldots, N$ we have $|y_i - (\mathbf{w}^T \mathbf{x}_i + b)| < \epsilon$. Also incorporating slack variables ζ_i and ζ_i^*, the problem can be described as (4).

$$\begin{array}{ll} \min & \tfrac{1}{2}|\mathbf{w}||^2 + C \displaystyle\sum_{i=1}^{n} (\zeta_i + \zeta_i^*) \\ \text{s.t.} & \begin{array}{rcll} y_i - \mathbf{w}^T \mathbf{x}_i - b & \leq & \epsilon + \zeta_i, & i = 1, \ldots, N \\ \mathbf{w}^T \mathbf{x}_i + b - y_i & \leq & \epsilon + \zeta_i^*, & i = 1, \ldots, N \\ \zeta_i, \zeta_i^* \geq 0 & & & \end{array} \end{array} \tag{4}$$

Since the solution to (4) only depends on inner products between vectors $\mathbf{x}_i$, the problem can be transformed into a higher dimension without much extra computation using kernels [21]. For the computation of the SVR the R kernlab package was used [22]. Although in this study initially both a linear kernel (hence no kernel), and the RBF kernel: $\kappa(x_i, x_j) = \exp\left(-\frac{||x_i-x_j||^2}{2\sigma^2}\right)$ were considered, not using a kernel surprisingly had a large negative effect on the runtime and was therefore disregarded. To find appropriate values for ϵ and C a grid search was applied using: $\epsilon \in \{0.01, 0.1, 1\}$ and $C \in \{1, 10\}$

7) Artificial Neural Networks: In this study we considered a feed-forward Artificial Neural Network with a single hidden layer. To find the weights the nnet R package was used, which utilizes an L-BFGS algorithm to find the appropriate weights [23], [24]. A grid search was applied to find an appropriate number of units in the hidden layer. During the grid search 1, 5, 10, 30, and 50 hidden units were considered.

C. Method evaluation

To evaluate the quality of predictive models two scenarios were distinguished. The first scenario assumes that application rates in weeks prior to the predicting period are known, which is comparable with predicting one week ahead. The second scenario assumes these application rates to be absent, and is more comparable with predicting 2 to 12 weeks ahead. These two scenarios are indicated by including PAR, and excluding PAR.

To evaluate the quality of the predictions two error measures are used: the root mean squared error: $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y_i} - y_i)^2}$, where $\hat{y_i}$ is the predicted value for actual y_i, and the determination coefficient: $R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$, with $SS_{res} = \sum_{i=1}^{n}(\hat{y_i} - y_i)^2$, the residual sum of squares, and $SS_{tot} = \sum_{i=1}^{n}(y_i - \bar{y})^2$, the total sum of squares. A 10-fold cross validation was applied to obtain accurate estimates for the quality of the predictions over the training set in both scenarios. Furthermore, a final prediction over the test set was made using the model showing the best results over the training set to estimate out of sample performance.

VI. RESULTS

A. Method comparison

Table I shows the best results per model when applying a 10-fold cross validation on the training set. The table indicates that Random Forest produced the best results both when predicting with and without PAR. Table I also indicates that multiple methods such as artificial neural networks without PAR, Poisson elastic net and Tweedie elastic net with PAR did not produce accurate results. Furthermore, Table I indicates that the added value of including previous application rates into the model is relatively small. Therefore, the predictive model would only produce slightly better results when predicting short term (1 week), in comparison with prediction long term (2 to 12 weeks).

When considering the quality of the models indicated in Table I it is important to note that the $RMSE$ is largely influenced by some large application rates which are difficult to predict. This can also be derived from the errors the Random Forest model makes on the test set (Fig. 2). In fact, 90% of the errors are smaller than 9.63, and the average absolute error over this 90% is 2.43.

TABLE I. RESULTS 10-FOLD CROSS VALIDATION

Method	Best $RMSE$ including PAR	Best R^2 including PAR	Best $RMSE$ excluding PAR	Best R^2 excluding PAR
Linear elastic net	11.82	0.35	11.87	0.34
Poisson elastic net	15.53	0	NA	NA
CART ANOVA	10.72	0.46	11.12	0.42
CART Poisson	10.17	0.52	10.75	0.46
Random Forest	9.38	0.59	9.93	0.54
Tweedie elastic net	13.86	0.03	11.62	0.37
SVR	10.58	0.46	11.28	0.41
ANN	11.14	0.42	17.35	0

While predicting the application rate, also the predictive value of online marketing campaigns was considered. To determine the importance of these campaigns the following procedure was used. For each tree of the forest the reduction in the sum of squared error when splitting on one of the D randomly selected attributes was computed. These reductions are summed up over all trees for each variable to obtain an overall picture of the decrease in residual sum of squares per predictor.

By comparing the reduction in the sum of squared error for each variable a comparison can be made between the effect of online marketing campaigns and other attributes. From this comparison we found that the effect of online marketing campaigns on the the application rate is small. Most variance is explained by predictors related to the application rate in prior weeks, the job title, the current recruitment lead time, and the contractual hours required. However, in contrast to the model predicting the total number of applicants, the online marketing campaigns do show a positive effect on the application rate.

B. Test set evaluation

Since Random Forest produced the most promising results when applying 10-fold cross validation this model was evaluated on the test set. The results are shown is Table II, whereas the distribution of the error on the test set is given in Fig. 2. The quality of the prediction was slightly worse than the average error obtained from 10-fold cross validation. Furthermore, just as the training set also the test set contained some large application rates which had a large negative effect on the $RMSE$.

TABLE II. RESULTS APPLYING RANDOM FOREST ON TEST SET

Performance metric	Value including PAR	Value excluding PAR
MAE	5.25	6.35
MSE	100.44	123.99
RMSE	10.02	11.13
Residual mean	1.53	1.81
Residual sd	9.90	10.98
R^2	0.44	0.32

VII. CONCLUSION

This paper considered the predictability of the number of weekly applications per vacancy and to which extend this metric can be controlled. To investigate this question a dataset from a large Dutch organization employing more than 30,000 employees was considered. To predict the number of weekly

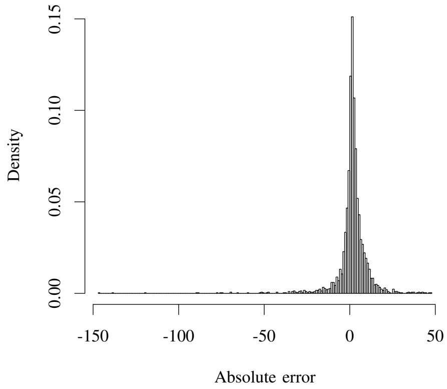

Figure 2. Test set error with PAR

applications per vacancy multiple machine learning methods were applied, of which Random Forest returned the best results with a root mean squared error of 9.38 and 9.93 when the predictors included and excluded application rates in weeks prior to the predicted week.

From closer examination of the errors two conclusions can be drawn. First, even though the predictions are quite accurate in most situations, i.e., have an error of less than 5 applicants, some vacancies can attract a large number of job seekers which the model finds hard to predict. As a result it will be more difficult to act on these predictions. On the other hand, both the predictions and insights into the variability of these predictions are helpful to manage the expectations recruiters and hiring managers might have when publishing a vacancy. In particular, recruiters should manage vacancies expecting a large number of applications carefully to avoid excess of applications. Also, recruiters could consider how attractive vacancies can be used to market less attractive vacancies, for example by generalizing the vacancy such that it may refer to both popular and less popular job positions.

Second, from analyzing the effect of online marketing campaigns on the number of applications per vacancy per week this effect is positive, though quite small. Also, it is likely that when estimating the effect of online marketing campaigns from historic data this effect will be biased: vacancies which do not attract many applications are more likely to be used in online marketing campaigns and there might be collinearity between the campaigns. Therefore, we were not able to draw a clear conclusion on the effect of online marketing campaigns and how this can be used to control the number of applications.

VIII. Further work

Although the amount of data recruitment departments are storing is increasing, the usability of this data for research can be limited. This study did not take into account the quality of new employees, individual job seeker behavior, and other incentives than online marketing campaigns due to limitations in obtaining this data either due to legal constraints, time constraints, or incomplete data sources. Including these data sources could provide new insights into how the quantity and quality of applications can be controlled.

References

[1] D. L. Van Rooy, A. Alonso, and Z. Fairchild, “In with the new, out with the old: Has the technological revolution eliminated the traditional job search process?” ”International journal of selection and assessment”, vol. 11, no. 2-3, 2003, pp. 170–174.

[2] R. Greenspan, Job seekers have choices, 2003, URL: https://www.clickz.com/job-seekers-have-choices/76679/ [accessed 2016-07-25].

[3] L. Abbot, R. Batty, and S. Bevegni, Global Recruiting Trends 2016, 2015, URL: https://business.linkedin.com/content/dam/business/talent-solutions/global/en_us/c/pdfs/GRT16_GlobalRecruiting_100815.pdf, [accessed 2016-07-27].

[4] F. Suvankulov, “Job search on the internet, e-recruitment, and labor market outcomes,” DTIC Document, Tech. Rep., 2010.

[5] R. R. Zusman and R. S. Landis, “Applicant preferences for web-based versus traditional job postings,” Computers in Human Behavior, vol. 18, no. 3, 2002, pp. 285–296.

[6] I. Lee, “The evolution of e-recruiting: A content analysis of fortune 100 career web sites,” Journal of Electronic Commerce in Organizations (JECO), vol. 3, no. 3, 2005, pp. 57–68.

[7] P. Gibson and J. Swift, “e2c: Maximising electronic resources for cruise recruitment,” Journal of Hospitality and Tourism Management, vol. 18, no. 01, 2011, pp. 61–69.

[8] D. Brown, “Unwanted online job seekers swamp HR staff,” Canadian HR reporter, vol. 17, no. 7, 2004, pp. 1–2.

[9] S. D. Maurer and Y. Liu, “Developing effective e-recruiting websites: Insights for managers from marketers,” Business horizons, vol. 50, no. 4, 2007, pp. 305–314.

[10] E. Parry and S. Tyson, “An analysis of the use and success of online recruitment methods in the uk,” Human Resource Management Journal, vol. 18, no. 3, 2008, pp. 257–274.

[11] S. Strohmeier and F. Piazza, “Domain driven data mining in human resource management: A review of current research,” Expert Systems with Applications, vol. 40, no. 7, 2013, pp. 2410–2420.

[12] D. Çelik and A. Elçi, “An ontology-based information extraction approach for résumés,” in Joint International Conference on Pervasive Computing and the Networked World. Springer, 2012, pp. 165–179.

[13] J. A. Hartigan and M. A. Wong, “Algorithm as 136: A k-means clustering algorithm,” Journal of the Royal Statistical Society. Series C (Applied Statistics), vol. 28, no. 1, 1979, pp. 100–108.

[14] M. Kuhn, “Building predictive models in R using the caret package,” Journal of Statistical Software, vol. 28, no. 5, 2008, pp. 1–26.

[15] D. A. Belsley, “A guide to using the collinearity diagnostics,” Computer Science in Economics and Management, vol. 4, no. 1, 1991, pp. 33–50.

[16] J. Friedman, T. Hastie, and R. Tibshirani, “Regularization paths for generalized linear models via coordinate descent,” Journal of statistical software, vol. 33, no. 1, 2010, p. 1.

[17] T. Hastie and J. Qian, Glmnet Vignette, 2014, URL: http://www.web.stanford.edu/∼hastie/Papers/Glmnet\textunderscoreVignette.pdf [accessed 2016-07-25].

[18] W. Qian, Y. Yang, and H. Zou, “Tweedies compound poisson model with grouped elastic net,” Journal of Computational and Graphical Statistics, vol. 25, no. 2, 2016, pp. 606–625.

[19] E. J. Atkinson and T. M. Therneau, “An introduction to recursive partitioning using the rpart routines,” Rochester: Mayo Foundation, 2000.

[20] A. Liaw and M. Wiener, “Classification and regression by randomforest,” R news, vol. 2, no. 3, 2002, pp. 18–22.

[21] A. J. Smola and B. Schölkopf, “A tutorial on support vector regression,” Statistics and computing, vol. 14, no. 3, 2004, pp. 199–222.

[22] A. Karatzoglou, A. Smola, and K. Hornik, The kernlab package, 2007, URL: https://cran.r-project.org/web/packages/kernlab/kernlab.pdf [accessed 2016-07-25].

[23] D. C. Liu and J. Nocedal, “On the limited memory bfgs method for large scale optimization,” Mathematical programming, vol. 45, no. 1-3, 1989, pp. 503–528.

[24] B. Ripley and W. Venables, Package ‘nnet’, 2016, URL: https://cran.r-project.org/web/packages/nnet/nnet.pdf [accessed 2016-07-25].