Predicting Bad Job Outcomes in Online Workplaces

Aaron Michelony AMICHELONY@UPWORK.COM
Ioannis Antonellis ANTONELLIS@UPWORK.COM
Upwork 441 Logue Avenue, Mountain View, CA 94043

Ramesh Johari [1] RAMESH.JOHARI@STANFORD.EDU
Stanford University Management Science and Engineering, 475 Via Ortega, Stanford, CA 94305

Abstract

More than one billion dollars' worth of work takes place every year in online workplaces like Upwork.com and Elance.com. We analyze the structure of these jobs and build a classifier using logistic regression and gradient tree boosting to identify jobs in trouble. We then report the effectiveness of this classifier in a user experiment. This submission to the ICML crowdsourcing workshop is part of a longer work involving detecting and intervening on bad jobs and preventing bad jobs in the future.

1. Introduction

Online labor markets help match people who pay to have work done, known as *clients*, with people who perform work, known as *freelancers*. Common requests for work include language translation and web programming jobs. When a client and freelancer are matched, they engage in a *job* that lasts until the work is completed or one of them cancels the job early. Most of the time these jobs are completed successfully; however, there is a significant fraction that is in trouble. These bad jobs cause problems and drive people away from the market.

During a job, the client and freelancer can communicate via messages on the workplace platform. The freelancer may also set up a work diary where snapshots of the freelancer's screen are taken and made available to the client. When the job is completed, the client and freelancer give each other public ratings and private ratings. Often the client and freelancer will work on multiple jobs together.

Sometimes the freelancer is unable to perform the job well enough to satisfy the client. Some reasons this could occur include poor freelancer communication, not having the skills required, or overly high standards from the client. Once signs of a bad job outcome appear, it would be helpful to be able to step in and either help remedy the job or terminate it early if necessary.

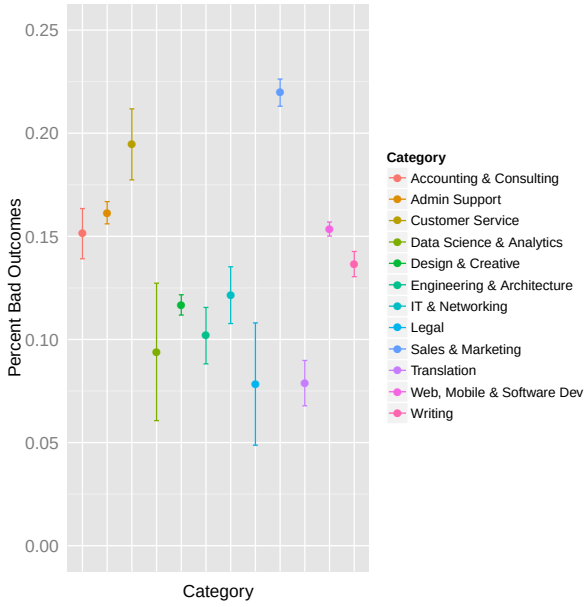

Figure 1. Percent bad outcome by job category with 95% confidence intervals

Our goal is to reduce the impact of bad jobs through targeted email intervention. We build a classifier that identifies jobs in trouble and send the client an email requesting the status of the job. If the client indicates anything other than high confidence, the client is engaged for treatment. Possible treatments to help the client include crediting the client, dispute resolution and performing a code review. We

Proceedings of the 31st International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

[1] Ramesh Johari was employed by Upwork (previously oDesk) when this work was carried out.

Feature
Freelancer has bad past outcomes
Client has bad past outcomes
Low amount spent for the job length
Long string of exclamation marks from client
Low mouse and keyboard events in work diary
Screensaver is on in work diary

Table 1. Most important features that indicate a bad outcome, ranked by information gain.

Problem	Example Terms
Computer	computer problems, computer issues
Death	death, died, passed away
Family	my wife, my kids, family problems
Late	late, delay
Sick	sick, fever, hospital, flu
Sorry	apologize, sorry, my fault
Vacation	on holiday, on vacation

Table 2. Some message terms that indicate a problem.

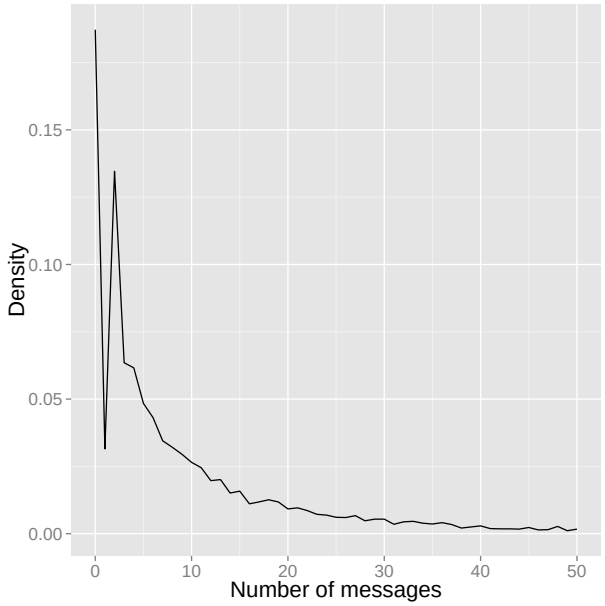

Figure 2. Distribution of the number of messages per job. Jobs with more than 50 messages are ignored. The dip at one and peak at two is due to many jobs where the client sends a message and the freelancer replies and that is the only communication. Notice that nearly one out of five jobs has no messages.

note that even if a client is refunded for a bad job, the client cannot get back the time spent on the job. Therefore, it is important to identify and stop bad jobs as soon as possible.

We have two main contributions:

- We perform the first known analysis of bad jobs in online labor data.
- We build a classifier to detect which jobs are likely to end in failure and show it performs better than a baseline of random guessing.

In contrast to most human computation or crowdsourcing jobs, which tend to be short, our jobs are longer, with a median time of two weeks in length, and tend to involve many higher-level tasks such as software engineering.

2. Intervention Classifier

2.1. Problem Definition

We want to predict bad outcomes for jobs that have begun on an online workplace, but have not yet finished. We build a classifier to identify bad jobs.

2.1.1. Job Outcome

When a job is finished, client and freelancer may rate each other using optional public and private feedback. Public feedback consists of a short text statement and scores from one to five on six different aspects: availability, communication, cooperation, deadlines, quality and skills. A total score is constructed based on these scores. A factor analysis shows two main factors: soft skills (availability, communication, cooperation, deadlines) and hard skills (quality, skills). Private feedback scores exist because clients and freelancers tend to avoid publicy giving poor scores (Horton & Golden, 2015). This private feedback frequently differs from the public feedback. From these feedback scores an outcome label is generated, which is either good, bad or neutral. Neutral jobs typically represent jobs with a lack of feedback.

2.2. Features

The most important features used to identify a bad job are in table 1. Additionally, the client rehiring a freelancer is the strongest signal that the job will end with a good outcome.

2.2.1. Messages

Messages are text sent between a client and freelancer. Most communication takes place off of the platform, so we are able to see only a small part of the overall communication. See figure 2 for the distribution of messages for jobs. We parse the messages and tag them using the Stanford part of speech (Toutanova et al., 2003) and named entity recognition (NER) taggers (Finkel et al., 2005). We aggregate text tokens into the NER tags "person", "organization" and "location". We apply standard natural language process-

Time	Sender	Message
0 days	Client B	Hi A, last year you helped me with an English-Polish translation. I have a new, 1600 word, translation job. The content consists of a manual and packaging text (see attached). The content is not very technical or complex. Are you available? If yes, what would be a fair fee? Looking forward to your reply, B
2 days	Client B	Hi A, Did you receive my message? Thanks, B
3 days	Freelancer A	Hello B, sorry for such a late response, but due to upcoming Easter it's really crazy here and I spend most of my time in my car. I would be able to translate the text around Thursday-Friday next week. Please let me know if that suits you. The estimated price for 1600 words is $88. Please let me know whether it suits you and have a grat Easter :) Kind regards, A

Table 3. All the messages for the first three days of a job. Time is the number of days since the job started. The outcome of this job was good, despite the freelancer having a delayed response to the client.

Sender	Message
Freelancer A	Thanks for being patient. I have solved the problem now my account is resumed. Odesk temporarily locked my account and I have failed to logged in for 2 days... I hope you will get your site tonight. It is 8 am and I am starting work and will be finished.... Thank you so much for being patient.
Client B	A, Tell me frankly, what is going on at your end? You are continuously asking for more time while you are unable to deliver!!! [. . .]

Table 4. Some messages for a job with a bad outcome.

ing techniques such as stemming and stopword elimination. We also apply TF-IDF to the messages by the client, by the freelancer and by both, giving us three sets of TF-IDF features. This is superior to only using TF-IDF that combines freelancer and client messages.

We use the following features:

- 1-, 2- and 3-grams of TF-IDF for client, freelancer and both
- Counts of messages, sentences, and words for client and freelancer, including the differences between client and freelancer
- Counts of how long the freelancer took to respond to the client's questions
- Punctuation counts, such as exclamation marks, question marks and emoticons
- Counts of capitalized words

We also find that freelancers tend to give excuses for poor performance. Some of these terms are in table 2. We stress that these features are indicative of a problem, but it does not necessarily mean the job will end poorly. For example, if the freelancer says "sorry", then it could be that the freelancer made a serious mistake or it could be that the freelancer made a minor mistake but is very polite. Tables 3 and 4 have example messages.

2.2.2. Work Diary

The work diary is generated by taking snapshots of a freelancer's desktop six times every hour. The counts of keyboard and mouse events are recorded along with an optional memo from the freelancer describing what is being worked on. Whether the freelancer's screensaver is on is also recorded.

2.2.3. Current Job Data

We also look at features for the current job such as total amount spent, hours worked, job category, job length, and bonus payment from the client to the freelancer. The job category, for example translation or software development, is important as well. We notice that software development jobs have bad outcomes at twice the rate of translation jobs. Figure 1 shows an overview of how categories vary by bad outcome. In our training and testing data, we only had available the current job features for the end of the job, which resulted in some data leakage. Since we wanted the classifier to be aware that some features, such as a bonus payment, affect the outcome, we left these features in, but it inflated the performance of the classifier.

2.2.4. Past Job Data

The past data for a freelancer and client are the most important features for our classifier. We look at features such as average job outcome and the public and private feedback given by and to the client and freelancer. We are able to achieve an area under the curve (AUC) of 0.73 using only these features on positive and negative jobs before the job starts.

2.3. Data Segmentation

We observe that if a freelancer and client stop communicating for a length of time and then one reinitiates communication, the gap in communication should be noted. Often communication will occur in cycles where the freelancer and client will discuss the work to be performed, followed by a period of no messages during which the freelancer completes part of the work. If there is at least a 48 hour gap in messages, we split the data so everything after the last message is put into a different segment. Each segment will have message data, work diary data, and current job data.

For each of the features we analyze, we determine a value for each segment. If a job has 10 segments, this will result in 10 values for each feature. We then take these 10 values and sort them to produce a seven-number summary (2nd, 9th, 25th, 50th, 75th, 91st and 98th percentiles) of each feature for each job. We are also able to normalize the job length this way. The exception is that we do not do this for the TF-IDF features.

For example, if a job has five segments, each feature will result in five values. One feature we want is a calculation of the longest string of exclamation marks. If the first segment has one exclamation mark, the second segment has two, the third segment has zero, the fourth segment has nine, and the fifth segment has zero, we end up with the values one, two, zero, nine, and zero for the feature of longest string of exclamation marks. We sort these values and summarize them using the seven-number summary. These seven numbers are then used as features representing the longest string of exclamation marks for this job.

2.3.1. Model

We built three models. The training data for them is taken from jobs started between January 2014 and June 2014, and ended before October 2014. The testing data is the jobs on October 1st, 2014. The first model is logistic regression with L_2 regularization, trained on 40,000 positive and 40,000 negative examples.

The second model is stochastic gradient tree boosting (Friedman, 1999) (Pedregosa et al., 2011). Gradient boosting iteratively fits and weights weak learners to obtain the function:

$$L(x) = \sum_{i=1}^{M} \gamma_i h_i(x) + const \tag{1}$$

where x is the input, M is the number of iterations, $h_i(x)$ is a weak learner and γ_i is the weak learner weight. Stochastic gradient boosting adds bagging for variance reduction. Our weak learner is a decision tree, limited to a depth of three.

Model	AUC
Logistic Regression	0.79
Gradient Boosting	0.81
Stacked Ensemble	0.82

Table 5. Model AUC on test data

This model was trained on 10,000 positive and 10,000 negative examples. We used a linear support vector machine with L_1 regularization to perform feature selection by taking the most significant 20,000 features.

We visually inspected the errors made by each model and saw they tended to make different errors. We then made an ensemble (Sculley et al., 2011) by weighting the two models using stacking (Sigletos et al., 2005) with logistic regression as the meta-level classifier. We are interested in selecting the worst jobs and intervening, so we compare the AUC. The results are in table 5, showing that the ensemble method is superior. We note that these results do not take neutral jobs into account, so the AUC on all jobs may be lower.

3. Experiment

To demonstrate the effectiveness of the classifier in practice, we ran an experiment. We randomly assigned clients into two buckets, both of which could receive intervention emails. The first bucket received emails if the probability of failure from the classifier was above a threshold, which was set so the worst tenth would receive emails. The other bucket was set so one tenth would receive emails randomly. We set the experiment to run for a month and ran it on a small sample of users.

The intervention email has five options as seen in figure 3. There are four selectable emoji representing confidence in the freelancer and a fifth option representing that the email is sent too early. The email is worded to avoid biasing the client into thinking there may be something wrong with the job.

The results of the experiment are in figure 4. We see that the classifier is superior at identifying jobs that are in trouble. The responses for "a little concerned" are statistically significant at the 95% level. The responses for "very confident" are significant at the 90% level.

We ended up with 53 responses for the control bucket and 60 for the treatment bucket. 93% of email respondents did not give feedback. The probability of responding is independent of the classifier prediction and the length of the job. We note that any response other than "very confident" indicates that a job has a higher-than-average probability of failure. The feedback is highly correlated with the job outcome; see figure 5. Therefore, our classifier is able to

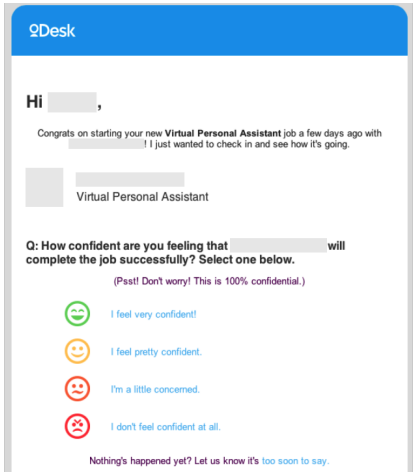

Figure 3. Email intervention with five choices. Responses other than "very confident" indicate a higher than expected probability of failure.

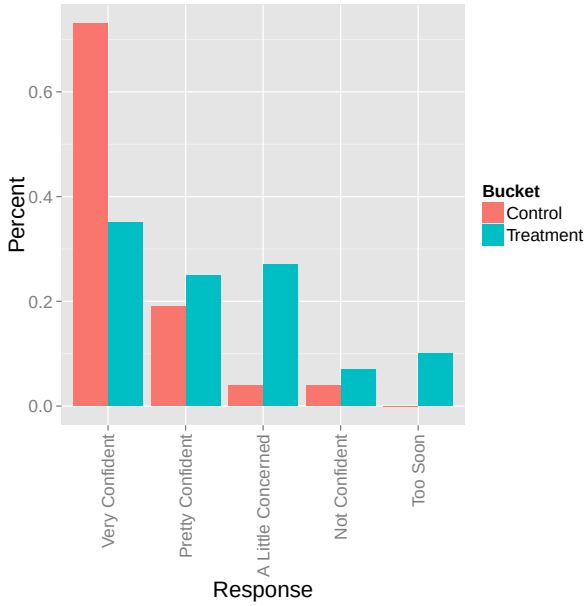

Figure 4. Experiment Results. The proportion of responses with "a little concerned" is statistically significant at the 95% level. The proportion of responses with "very confident" is statistically significant at the 90% level.

predict more bad outcomes than random.

4. Related Work

The intervention we perform is related to the field of recommender systems. Recommender systems push recommendations to users of a system in order to increase the user's utility. One similar recommendation system comes from giving users recommendations of new jobs, in terms of employment, in order to switch jobs (Wang et al., 2013). They use a hierarchical Bayesian proportional hazards model to give users recommendations on a job website in order to pick which users receive recommendations and also which time they receive them.

Cyberbulling is another field that involves intervention between online parties. One framework for cyberbulling uses four I's: identity, inference, influence and interaction/intervention (Chen et al., 2012). Our work is similar in that we infer undesirable state from messages and perform intervention, but different in that we don't have problems identifying people. We also differ in that we don't have influence, which is social network effects. In our platform, bad messages do not propogate through the system as they may on social media.

Our work is also related to human computation and crowdsourcing. This work differs in that it deals with jobs that are longer than most crowdsourcing jobs, the jobs tend to involve higher skill than most crowdsourcing jobs and we focus on the state of the job before it ends. See (Law & von Ahn, 2011) for an overview of human computation.

5. Conclusions and Future Work

We find that we are able to identify jobs with bad outcomes to some degree, but it is a challenging task. We identified that the most important features for determining the outcome of a job in an online marketplace are the past performance of the client and freelancer. We ran an experiment to determine the effectiveness of our classifier and showed it demonstrated superior performance in identifying jobs with bad outcomes compared to random selection.

This work is part of a larger work involving detecting jobs that are likely to end in failure and creating successful client and freelancer interventions that are able to improve future performance. Our future work involves three things:

- Improving the precision of our classifier.
- Sending more strongly-worded emails to clients who have jobs that are likely to fail.
- Designing effective treatment plans for freelancers and clients to prevent future bad outcomes.

We plan on working to improve the precision of the classifier in order to support more strongly-worded emails. As other system improvements take place, such as an improved recommendation engine for matching freelancers to jobs, bad jobs are expected to decrease both in number and severity. This would make finding bad jobs more difficult and makes a high precision classifier harder to build. There are

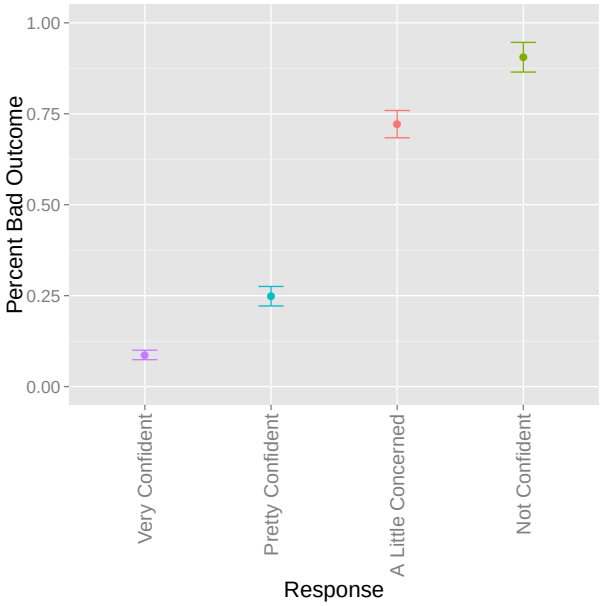

Figure 5. Percent bad outcome compared to responses to the intervention email with 95% confidence intervals. Here we include responses outside of the experiment. Given a response, the job outcome is much more dependent on the response than the classifier prediction.

two ways to combat this: the first is to allow the client to raise a flag if he or she perceives a bad outcome will happen; however, this depends on the client's initiative. Another is to add more features to the system to better monitor the state of a job.

More strongly-worded emails would allow us to exert more influence on clients. We need to experiment with different emails in order to understand the influence an email will have, and balance this influence with the precision of the classifier. We could also send multiple emails per job. We also hope this will remedy the low response rate.

Finally, we would like to design effective treatment plans. If a client is unhappy with the work a freelancer has done, we currently offer several treatments including crediting the client, dispute resolution and performing a code review. We would like to monitor these treatments to see which, if any, has a positive effect on future job performance. Then we can design more effective treatments, including educating clients and freelancers about best practices for success, such as regular communication. Additionally, many freelancers come from foreign countries and may not understand online workplace culture; therefore, simply educating them about what is normal and expected may help.

Acknowledgements

We would like to thank David Abramson and Abby Tyo of Upwork for help implementing the experiment.

References

Chen, Yunfei, Zhang, Lanbo, Michelony, Aaron, and Zhang, Yi. 4is of social bully filtering: identity, inference, influence, and intervention. *Proceedings of the 21st ACM international conference on Information and knowledge*, pp. 2677–2679, 2012.

Finkel, Jenny Rose, Grenager, Trond, and Manning, Christopher. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pp. 363–370, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219885. URL http://dx.doi.org/10.3115/1219840.1219885.

Friedman, Jerome H. Stochastic gradient boosting. *Computational Statistics and Data Analysis*, 38:367–378, 1999.

Horton, John J and Golden, Joseph M. Reputation inflation: Evidence from an online labor market. 2015.

Law, Edith and von Ahn, Luis. *Human Computation*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2011. doi: 10.2200/S00371ED1V01Y201107AIM013. URL http://dx.doi.org/10.2200/S00371ED1V01Y201107AIM013.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine Learning in Python . *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Sculley, D., Otey, Matthew Eric, Pohl, Michael, Spitznagel, Bridget, Hainsworth, John, and Zhou, Yunkai. Detecting adversarial advertisements in the wild. In *Proceedings of the 17th ACM SIGKDD International Conference on Data Mining and Knowledge Discovery*, 2011. URL http://www.eecs.tufts.edu/~dsculley/papers/adversarial-ads.pdf.

Sigletos, Georgios, Paliouras, Georgios, Spyropoulos, Constantine D., and Hatzopoulos, Michael. Combining information extraction systems using voting and stacked generalization. *Journal of Machine Learning Research*, 6:1751–1782, 2005. URL http://jmlr.csail.mit.edu/papers/volume6/sigletos05a/sigletos05a.pdf.

Toutanova, Kristina, Klein, Dan, Manning, Christopher D., and Singer, Yoram. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pp. 173–180, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1073445.1073478. URL http://dx.doi.org/10.3115/1073445.1073478.

Wang, Jian, Zhang, Yi, Posse, Christian, and Bhasin, Anmol. Is it time for a career switch? In *Proceedings of the 22nd international conference on World Wide Web*, pp. 1377–1388. International World Wide Web Conferences Steering Committee, 2013.