Predicting Self-reported Customer Satisfaction of Interactions with a Corporate Call Center

Joseph Bockhorst, Shi Yu, Luisa Polania, and Glenn Fung

American Family Insurace
Strategic Data & Analytics
Machine Learning unit
6000 American Parkway
Madison, WI USA 53783

Abstract. Timely identification of dissatisfied customers would provide corporations and other customer serving enterprises the opportunity to take meaningful interventions. This work describes a fully operational system we have developed at a large US insurance company for predicting customer satisfaction following all incoming phone calls at our call center. To capture call relevant information, we integrate signals from multiple heterogeneous data sources including: speech-to-text transcriptions of calls, call metadata (duration, waiting time, etc.), customer profiles and insurance policy information. Because of its ordinal, subjective, and often highly-skewed nature, self-reported survey scores presents several modeling challenges. To deal with these issues we introduce a novel modeling workflow: First, a ranking model is trained on the customer call data fusion. Then, a convolutional fitting function is optimized to map the ranking scores to actual survey satisfaction scores. This approach produces more accurate predictions than standard regression and classification approaches that directly fit the survey scores with call data, and can be easily generalized to other customer satisfaction prediction problems. Source code and data are available at https://github.com/cyberyu/ecml2017.

1 Introduction

In a competitive customer-driven landscape where businesses are constantly competing to attract and retain customers; customer satisfaction is one of the top differentiators. While digitization and other forces continue to increase consumer choice, understanding and improving customer satisfaction are often core elements of the business strategy of modern companies. It enables service providers to unveil timely opportunities to take meaningful interventions to improve customer experience and to train customer representatives (CR) in an optimal way.

In order to measure the effectiveness of a CR during a phone interaction with a customer, generally a customer survey is taken shortly after the call takes place. However, due to survey expense, typically only a small percentage of calls are measured. When CR performance is calculated from a small sample of surveys performance scores have high variability and there is potential misrepresentation of CR performance.

The focus of this work is to describe the design and implementation of a deployed machine-learning-based system used to automatically predict customer satisfaction following phone calls. Our discovery and system design process can be divided into four stages:

1. **Extraction, processing and linking of raw data**: Raw data is collected and linked from four primary sources: call logs, historical survey scores, customer and policy databases, and call transcription and related content derived from audio recordings.
2. **Feature engineering**: Call data is processed to create informative features.
3. **Model design and creation**: In this stage we focus on the design and creating of the customer satisfaction predictive models.
4. **Aggregation of model predictions to the group level**: At the last stage, we aggregate individual model predictions to the group level (by call queue, by CR, in a given period of time, etc.). We also provide estimated bounds for the group average predictions.

2 Related Work

Research studies on emotion recognition using human-human real-life corpus extracted from call center calls are limited. In [15], a system for emotion recognition in the call center domain, using lexical and paralinguistic cues, is proposed. The goal was to classify parts of dialogs into three emotional states. Training and testing was performed on a corpus of 18 hours of real dialogs between agent and customer, collected in a service of complaints. A similar work [2], also proposes to classify call center calls between three emotional states, namely, anger, positive and neutral. The authors used classical descriptors, such as zero crossing rate and Mel-frequency cepstral coefficients, and support vector machines as the classifier. They used service complaints and medical emergency conversations from call centers, and adopted a cross-corpus methodology for the experiments, meaning that they use one corpus as training set and another corpus as test set. They attained a classification accuracy between 40% to 50% for all the experiments.

Park and Gates [10] developed a method to automatically measure customer satisfaction by analyzing call transcripts in near real-time. They identified several linguistics and prosodic features that are highly correlated with behavioral aspects of the speakers and built machine learning models that predict the degree of customer satisfaction in a scale from 1 to 5 with an accuracy of 66%. Sun *et al.* [13] adopted a different approach, based on fusion techniques, to predict the user emotional state from dialogs extracted from a Chinese Mobile call center corpus. They implemented a statistical model fusion to alleviate the data imbalance problem and combined n-gram features, sentiment word features and domain-specific words features for classification.

Recently, convolutional neural networks have been used on raw audio signals to automatically learn meaningful features that lead to successful prediction of self-reported customer satisfaction from call center conversations in Spanish [12]. This approach starts by pretraining a network on debates from French TV shows

with the goal of detecting salient information in raw speech that correlates with emotion. Then, the last layers of the network are finetuned with more than 18000 conversations from several call centers. The CNN-based system achieved comparable performance to the systems based on traditional hand-designed features.

There are many machine learning problems, referred to as ordinal ranking problems, where the goal is to classify patterns using a categorical scale which shows a natural order between labels, but not a meaningful numeric difference between them. For example, emotion recognition in the call center domain usually involves rating based on an ordinal scale. Indeed, psychometric studies show that human ratings of emotion do not follow an absolute scale [9,8]. Ordinal ranking is fundamentally different from nominal classification techniques in that order is relevant and the labels are not treated as independent output categories. The ordinal ranking problems may not be optimally addressed by the standard regression either since the absolute difference of output values is nearly meaningless and only their relative order matters [3].

There are several algorithms which specifically benefit from the ordering information and yield better performance than nominal classification and regression approaches. For example, Herbrich *et al.* [5] proposed a support vector machines approach based on comparing training examples in a pairwise manner. A constraint classification approach that works with binary classifiers and is based on the pairwise comparison framework was proposed by Har-Peled *et al.* [4]. Crammer and Singer [1] developed an ordinal ranking algorithm based on the online perceptron algorithm with multiple thresholds.

Some areas where ordinal ranking problems are found include medical research [11], brain computer interface [17], credit rating [7], facial beauty assessment [16], image classification [14], and more. All these works are examples of applications of ordinal ranking models, where exploiting ordering information improves their performance with respect to their nominal counterparts.

3 Overview of the proposed system

Our main goal is to develop a model to predict satisfaction scores for all incoming customer calls in order to (i) take meaningful timely interventions to improve customer experience and (ii) obtain a robust understanding on how care center performance and training can be enhanced, ultimately for our customer's benefit.

Our company recently adopted a system which automatically transcribes phone calls to text. The transcriptions generated by this system are key for our deployed system. The company customer care center monitors customer satisfaction by offering surveys conducted by a third party vendor to 10% of incoming calls. Each care center CR has around five surveys completed per month, which is only about 0.5-1% of all assigned calls. There are four topics measured by the survey: (a) If the customer felt "valued" during the call; (b) If the issue was resolved; (c) How polite the CR was, and (d) How clearly the CR communicated during the call. Scores range form 1 to 10 (1 lowest, 10 highest) and the

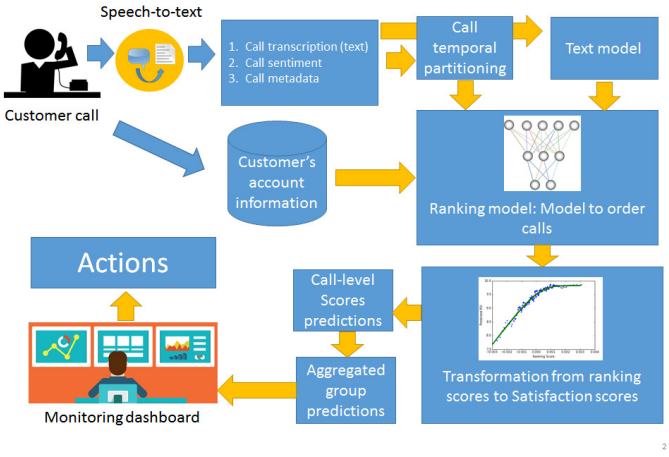


Fig. 1: Overview of the deployed system

four scores are averaged into an additional variable called RSI (Representative Satisfaction Index). In this paper we focus on predicting the RSI.

Several difficulties, in terms of modeling, are discovered after a quick initial inspection of the training data:

- The customer satisfaction scores (RSI) are highly biased towards the highest score (10), while calls with scores lower than 8 are less than 4%. This highly skewed distribution makes building a predictive model more complex.
- Survey scores are customer responses, thus are subjective, qualitative states heavily impacted by personal preferences.
- The measurement scale of survey scores is ordinal; one cannot say, for example, that a score of 10 indicates double satisfaction as a score of 5. Most, if not all, standard regression techniques implicitly assume an interval or ratio scale.

Figure 1 displays an overview of the deployed system. The system workflow can be summarized by the following steps:

1. After a call ends, a transcript of the call is automatically produced by a speech-to-text system developed by Voci (vocitec.com).
2. Calls are partitioned into temporal segments and non-text features are engineered. The rationale of temporal segmentation is that certain events are more relevant depending of when they occur in the call. For example: detecting negative sentiment trends in the first quarter of the call but positive at the end may lead to a higher satisfaction score than when the opposite is true.
3. Textual features are constructed and merged with non-text features. The fused feature vectors are used as input features for the models described in the next step.

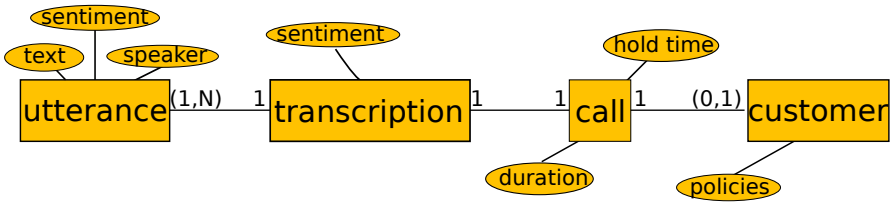


Fig. 2: Partial entity-relationship data model of input data. Numbers indicate cardinality ratios between entities. Not all attributes are shown.

4. Ranking model scoring. The ranking model is trained by sampling ordered pairs based on satisfaction scores.
5. Mapping from raking scores to satisfaction scores using an isotonic model. Individual (per call) satisfaction predictions are generated.
6. Aggregation of calls at the group level are stored in a database. Example groups include: per CR, per queue and per time period.
7. Aggregations are used for real-time reporting though a monitoring dashboard.

4 Representation

This section describes the pipeline of extracting features from various types of input data sources related to a phone call which are passed on to the models. Available input data sources are call transcriptions, call logs, and other customer and policy data. Figure 2 displays our input data model.

Calls are transcribed to sequences of non-overlapping *utterances*, chunks of semi-continuous speech by a single speaker flanked on either side by either a change of speaker or a break in speech. Each utterance contains the transcribed text along with related attributes including the predicted speaker, either customer or company representative, start and end times, and predicted sentiment. Concatenating the text of all utterances gives us the full transcribed text of a call. In addition to the call transcription, we generate features from the telephony system logs. Examples of log level attributes are assigned call-center queue, waiting time and transfer indicators. For calls that are linked to specific customers we use addtional customer and policy data.

4.1 Feature Engineering

Our feature engineering process takes linked input data for a call and produces a fixed-length feature vector.

Table 1: A temporal segment feature is created for each of the 300 combinations ($5 \times 3 \times 4 \times 5$) of component values.

Component	**Possible Values**
Utterance function	negSent(), negCount(), duration(), consNeg(), sentScore()
Speaker	representative, customer, either
Aggregate function	min(), max(), mean(), std()
Temporal range	[0.0, 0.25), [0.25, 0.5), [0.5, 0.75), [0.75, 1.0], [0.9, 1.0]

Temporal Segment features Each temporal segment feature represents an aspect of the call in a certain temporal range, for example, the minimum sentiment score of any customer utterance in the last quarter of the call. A temporal segment feature is defined by i) a numerical utterance function[1], ii) a speaker, iii) an aggregate function and, iv) a temporal range (see Table 1).

Temporal Segment Text-features The text of each transcribed customer call can also be viewed as a linear sequence of temporal elements (words) thus can be decomposed into temporal textual segmentations. In fact, each customer call consists of several natural temporal segmentations, which usually starts with greetings, then customer personal information authentication, next followed by customer's narrations of problems or requests, and then responses and resolutions provided by the representative, and finishes by ending courtesies of both parties. To predict customer satisfaction, we assume that late segmentations of a call (i.e., problem explanations, resolutions) are more informative than early parts (i.e., greetings, authentication), therefore we create separate textual models by decomposing the transcribed text of a call into different temporal segments.

We denote $\mathcal{D}$ as the corpus of transcribed text of all calls, where $d_i \in \mathcal{D}$, $i = 1...N$ is a document of transcribed text of the i-th call. Each d_i is composed of a sequence of words $w_{i,j}$, $j = 1...M_i$ where M_i is the total number of words in d_i. And we further decompose all the words in a document into four sub-documents $q_{i1}, q_{i2}, q_{i3}, q_{i4}$, where

$$
\begin{aligned}
q_{i1} &= \{w_{i,1}, ..., w_{i,s_1}\}, \\
q_{i2} &= \{w_{i,s_1+1}, ..., w_{i,s_2}\}, \\
q_{i3} &= \{w_{i,s_2+1}, ..., w_{i,s_3}\}, \\
q_{i4} &= \{w_{i,s_3+1}, ..., w_{i,M_i}\}.
\end{aligned}
$$

Since each call has different lengths, and we haven't applied any method to automatically segment a call according to the content, we simply set s_1, s_2, s_3 respectively to the rounded integers of $\frac{M_i}{4}$, $\frac{2M_i}{4}$, $\frac{3M_i}{4}$, thus gives us four even

[1] negSent() is an indicator that is 1 if the utterance sentiment label is *Negative*, negCount() is the number of *Negative* or *Mostly Negative* sentiment phrases in the utterance, duration() is the length of the utterance in seconds, consNeg() is an indicator that is 1 if the current and previous utterance have negative sentiment, and sentScore() maps utterance sentiment labels (*Negative, Mostly Negative, Neutral, Mostly Positive, Positive*) to numerical scores (-1, -0.5, 0, 0.5, 1).

temporal segments, where each segment contains words appeared in a quarter part, from beginning to end, of a call and we call them *quarter documents*.

Analogously, using the same s_1, s_2, s_3 chosen before, we define four sets of *tail documents*

$$
\begin{aligned}
t_{i1} &= \{w_{i,1}, ..., w_{i,M_i}\}, \\
t_{i2} &= \{w_{s_1+1,1}, ..., w_{i,M_i}\}, \\
t_{i3} &= \{w_{s_2+1,1}, ..., w_{i,M_i}\}, \\
t_{i4} &= \{w_{s_3+1,1}, ..., w_{i,M_i}\},
\end{aligned}
$$

as segmented documents of various lengths. Notice that t_{i1} is equivalent to d_i, and t_{i2}, t_{i3}, t_{i4} are respectively the remaining $75\%, 50\%, 25\%$ part of a call.

Thus, we obtain eight corpora of call text (four quarter documents and four tail documents) and each corpus represents a temporally segmented snapshot of the textual content. Next, we construct standard TF-IDF profiles on each individual corpus, where a row represents a call, and benchmark the best corpora using a held-out training and validation set. We find that the corpus composed by t_{i3}, represented by 5000 TFIDF weights, gives the best performance and we select that for modeling.

Other features Additional features are created from telephony logs, such as duration of call, queue, in-queue waiting time, and policy count information such as the number of auto policies, number of property policies, *etc.* held by the customer's household. Our system has a total of 5,340 natural features, and following one-hot-encoding of categorical features the final model ready dataset contains 5,501 features.

5 Models

Here we describe our approach to learning a predictive model of ordinal satisfaction ratings, such as RSI. The modeling task is to learn a function $f(\mathbf{x}) = \hat{y}$ mapping feature vector $\mathbf{x}$ to predicted RSI $\hat{y}$ such that on average the difference between the predicted score and actual score y is small. Our approach involves two models: a linear ranking model $r(\mathbf{x})$ that maps examples to rank scores and a non-decreasing, non-linear model $s(\hat{r})$ mapping rank scores to RSI. We form $f()$ through composition: $f(\mathbf{x}) = s(r(\mathbf{x}))$. We term this approach RS+IR for "rank score + isotonic regression".

Unlike standard linear and non-linear regression methods that directly model y, the RS+IR approach is consistent with the ordinal scale of the satisfaction score. A second advantage of RS+IR is that since the rank score model is learned from pairs of examples (see below), a larger pool of training examples are available and the class labels of the training set can be balanced, which is especially important for data sets like those considered here that are strongly skewed towards the high end of the satisfaction scale.

Rank score model We learn a model to rank examples by RSI using the pairwise transform [6]. The pairwise transform induces a rank score function $r(\mathbf{x})$ by learning a linear binary classifier from an auxiliary training set of examples $(\mathbf{u}, v)$ that are formed from pairs of examples $(\mathbf{x_i}, y_i), (\mathbf{x_j}, y_j)$ in the original ordinal training set that have different satisfaction scores[2]

The features of the auxiliary examples are the component-wise difference between the original examples, $\mathbf{u}_{ij} = \mathbf{x_i} - \mathbf{x_j}$. The binary class value v_{ij} indicates whether or not example i has a higher satisfaction than example j: v_{ij} is +1 if $y_i > y_j$ and -1 if $y_i < y_j$. The linear binary classifier $r(\mathbf{u})$, which is learned from the auxiliary training set to predict which of two examples has a higher satisfaction score, is subsequently used as a rank score function $r(\mathbf{x})$. That $r()$ can be used as a ranking function follows from its linearity $r(\mathbf{x}_i - \mathbf{x}_j) = r(\mathbf{x}_i) - r(\mathbf{x}_j)$ and by noticing that $r(\mathbf{x}_i) > r(\mathbf{x}_j)$ is consistent with the prediction that y_i is larger than y_j.

Rank score to satisfaction The second sub-model $s(r)$ is one-dimensional, non-decreasing function mapping rank scores to satisfaction scores. After learning the rank score model $r()$ we calculate the rank score of all examples in the original training set, order examples by their rank scores and smooth the resulting sequence of satisfaction scores. We then fit an isotonic regression model using training examples sampled uniformly from the smoothed function.

6 Results

In this section we describe the results of experiments conducted on a data set of 8,726 incoming phone calls from between March 23, 2015 and Dec 29, 2015 for which we have customer satisfaction survey results. We randomly selected 75% (6,108) for the training set and the remainder served as our test set.

6.1 Individual Predictions

To assess the value of our "rank score + isotonic regression" (RS+IR) approach to predicting phone call representative satisfaction index (RSI) scores we compared it with three standard regression methods (ridge regression, Lasso and random forest regression) and one classification method, linear support vector machine[3]. Ridge regression and Lasso are both penalized linear regression methods, but use different loss functions: L2 for ridge and L1 for lasso. Random forest regression is a non-linear approach that trains different ensembles of least-squares linear models for non-overlapping partitions of the input space. We use cross-validation on the training set to set hyperparameters (α for ridge and lasso, max_depth and min_samples_per_split for random forest, and C for SVM).

[2] All the auxiliary examples may not be needed. We have found that while there are over 10 million auxiliary examples that can be formed from our training set, the rank score model is well converged when trained with 10,000 examples. We experimented with various techniques for sampling the auxiliary examples (biased for large RSI difference, small RSI difference, *etc*), and found that simple uniform sampling works best.

[3] All comparison models trained using the scikit-learn Python package.

	Pe.	Sp.	MAE
Ridge	0.300	0.231	0.811
Lasso	0.303	0.227	0.815
Random forest	0.149	0.150	0.835
Rank Score	0.255	**0.239**	*
RS+IR	**0.312**	**0.239**	**0.784**

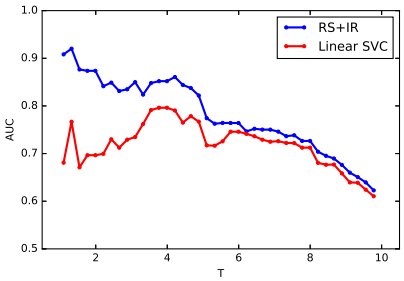
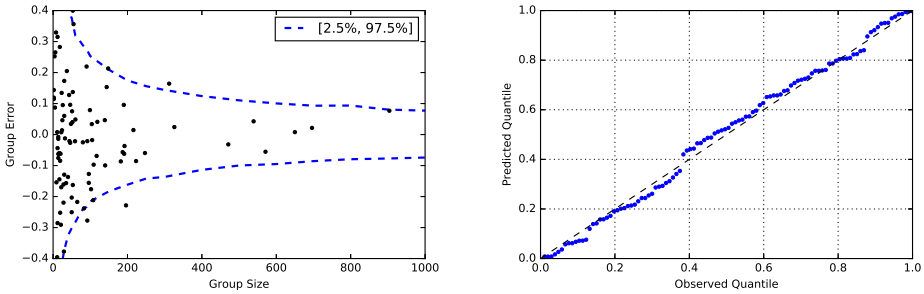


Fig. 3: (Left) Regression results (Pe:Pearson correlation, Sp:Spearman correlation, MAE: mean absolute error). (Right) classification results

Figure 3(left) shows test set results. The RS+IR model outperforms the other models in terms of Pearson correlation, Spearman correlation and mean absolute error (MAE). Also, RS+IR has better Pearson correlation than the rank score alone, showing the value of the non-linear mapping from rank score to prediction. If actions are taken in response to model predictions, for example reaching out to potentially dissatisfied customers, when predicted RSI falls below a given threshold T classification models are more appropriate than regression. The right panel shows the area under an ROC curve as T varies for our approach and linear SVM. Even though we trained a different SVM model for each value of T and only a single RS+IR model, the AUC of the RS+RI model dominates the SVM over the whole range of T, especially for smaller thresholds.

6.2 Group Predictions

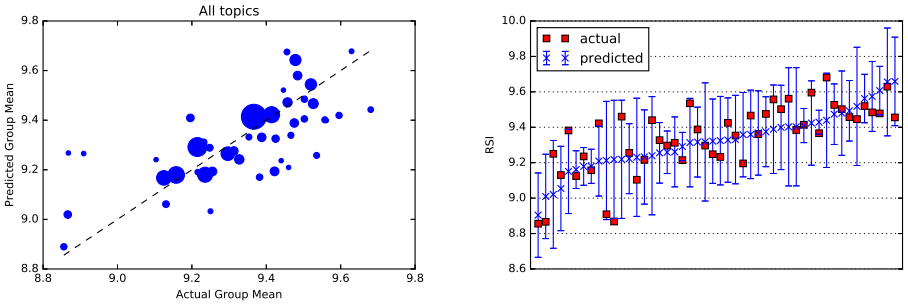


(a) (b)

Fig. 4: (a) Dashed lines indicate 95% confidence band for randomly selected groups. Points are observed group errors of topics groups containing more than 50 calls. (b) Quantile/quantile plot of group errors for the topics groups.

Since users of the productionalized system view reports on mean predicted satisfaction scores for various collections of calls, for example by department, call-center queue, and hour-of-day, we have investigated our system's accuracy for call groups. We use two kinds of groupings: random and by *topic*. We formed random groups of a given size by sampling calls with replacement from the test set. For the topic groups we used hand-crafted text-based predicates, which were created by another business unit for tagging calls related to various products and services and aspects of the customer journey. Each topic predicate is a Boolean function that takes a single sentence as input. A call belongs to a topic T if $T(s)$ is `true` for any sentence s in the call. Thus, a given call may belong to zero, one or many topics. There are a total of 107 topics groups with group sizes ranging from 1 to 1,560.

We define the group error to be the difference between the mean of the predicted scores for all calls in the group and the mean of the actual satisfaction scores. We form random groups with between 10 and 1,000 calls and for each group size we formed 5,000 replicate random groups. The dashed blue lines in Figure 4(a) show 95% confidence bands for the group error of the random groups. That is, for a given group size the group error of 95% of groups of that size in our simulation fell between the bands. We can see from this figure that group error decreases with group size.



(a) (b)

Fig. 5: (a) Mean predicted RSI vs. mean satisfaction RSI for the topics groups. Bubble area is proportional to group size. Group sizes range from 50 to 1,560. (b) Mean predicted RSI for the topics groups with 95% confidence intervals.

We use the bands of Figure 4(a) to determine tolerance levels for deciding when to raise alarms due to differences between predicted and actual satisfaction scores. The points represent the errors of the topics groups. The errors for 45 of the 48 topics groups (93.75%) with more than 50 calls fall between the 95% confidence bands. This provides evidence that the topic groups have similar error profiles to natural groupings by topic. Figure 4(b) shows the quantile/quantile plot for the group error of all 107 topics groups using the errors of random groups

of similar size to compute the observed percentile. As the points lie close to the ideal diagonal line, we conclude that the error profiles of random groups and topics group are similar.

Figure 5(a) shows the predicted and actual mean satisfaction for topics groups with at least 50 calls. The area of each bubble is proportional to the number of calls in the group, which ranges from a minimum of 50 to a maximum of 1,560. There is a general agreement (Pearson correlation = 0.73) between the predicted and actual group means. And in general, as with the random groups, larger groups have smaller within group errors. Figure 5(b) shows the predicted group mean with 95% confidence interval (dependent on the group size) and the actual group mean for these same 48 groups. Again, as this is a different view of the same data represented by the points of Figure 4(a), we see that the confidence bounds determined by random group errors do an excellent job of describing the distribution of errors in the topics groups.

7 Conclusions and lessons learned

This paper presents an efficient and accurate method for predicting self-reported satisfaction scores of customer phone calls. Our approach has been implemented into a production system that is currently predicting caller satisfaction of approximately 30,000 incoming calls each business day and generating frequent reports read by call-center mangers and decision makers in our company.

We described several techniques that we suspect will generalize to related tasks. (i) Rather than applying regression models directly on the ordinal data, we use a linear ranking sub-model along with a non-linear isotonic regression sub-model for predicting satisfaction. We presented empirical evaluation that shows this approach yields more accurate satisfaction predictions than standard regression models. (2) Temporally segmented features constructed from call meta-information and transcribed text are shown to be useful to capture informative signals relevant to customer satisfaction. (3) The average satisfaction prediction for groups of calls, instead of by only individual calls, agrees very strongly with actual satisfaction scores, especially for large groups. (4) We provided methods for determining system tolerance levels based on the deviation between predicted and actual group predictions that we use to verify that the production system is performing as expected.

References

1. K. Crammer and Y. Singer. Online ranking by projecting. *Neural Computation*, 17(1):145–175, 2005.
2. L. Devillers, C. Vaudable, and C. Chastagnol. Real-life emotion-related states detection in call centers: a cross-corpora study. In *INTERSPEECH*, volume 10, pages 2350–2353, 2010.
3. P. Gutiérrez, M. Perez-Ortiz, J. Sanchez-Monedero, F. Fernández-Navarro, and C. Hervas-Martinez. Ordinal regression methods: survey and experimental study. *IEEE Transactions on Knowledge and Data Engineering*, 28(1):127–146, 2016.

4. S. Har-Peled, D. Roth, and D. Zimak. Constraint classification: A new approach to multiclass classification. In *International Conference on Algorithmic Learning Theory*, pages 365–379. Springer, 2002.
5. R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. *Advances in neural information processing systems*, pages 115–132, 1999.
6. R. Herbrich, T. Graepel, and K. Obermayer. Support vector learning for ordinal regression. In *In International Conference on Artificial Neural Networks*, pages 97–102, 1999.
7. K. Kim and H. Ahn. A corporate credit rating model using multiclass support vector machines with an ordinal pairwise partitioning approach. *Computers and Operations Research*, 39(8):1800–1811, 2012.
8. A. Metallinou and S. Narayanan. Annotation and processing of continuous emotional attributes: Challenges and opportunities. In *IEEE International Conference and Workshops on Automatic Face and Gesture Recognition*, pages 1–8, 2013.
9. S. Ovadia. Ratings and rankings: Reconsidering the structure of values and their measurement. *International Journal of Social Research Methodology*, 7(5):403–414, 2004.
10. Y. Park and S. Gates. Towards real-time measurement of customer satisfaction using automatically generated call transcripts. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1387–1396. ACM, 2009.
11. M. Pérez-Ortiz, M. Cruz-Ramírez, M. Ayllón-Terán, N. Heaton, R. Ciria, and C. Hervás-Martínez. An organ allocation system for liver transplantation based on ordinal regression. *Applied Soft Computing*, 14:88–98, 2014.
12. C. Segura, D. Balcells, M. Umbert, J. Arias, and J. Luque. Automatic speech feature learning for continuous prediction of customer satisfaction in contact center phone calls. In *Advances in Speech and Language Technologies for Iberian Languages: Third International Conference, IberSPEECH 2016, Lisbon, Portugal, November 23-25, 2016*, pages 255–265. Springer, 2016.
13. J. Sun, W. Xu, Y. Yan, C. Wang, Z. Ren, P. Cong, H. Wang, and J. Feng. Information fusion in automatic user satisfaction analysis in call center. In *International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, volume 1, pages 425–428, 2016.
14. Q. Tian, S. Chen, and X. Tan. Comparative study among three strategies of incorporating spatial structures to ordinal image regression. *Neurocomputing*, 136:152–161, 2014.
15. C. Vaudable and L. Devillers. Negative emotions detection as an indicator of dialogs quality in call centers. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5109–5112, 2012.
16. H. Yan. Cost-sensitive ordinal regression for fully automatic facial beauty assessment. *Neurocomputing*, 129:334–342, 2014.
17. J. Yoon, S. Roberts, M. Dyson, and J. Gan. Bayesian inference for an adaptive Ordered Probit model: An application to brain computer interfacing. *Neural Networks*, 24(7):726–734, 2011.