Intelligent Data Analysis 22 (2018) 867–880
DOI 10.3233/IDA-173586
IOS Press

Towards semi-automatic human performance evaluation: The case study of a contact center

Andrea Brunello[a,*], Fernando Jiménez[b], Enrico Marzano[c], José Palma[b], Gracia Sánchez[b] and Guido Sciavicco[d]
[a]*Department of Mathematics, Physics, and Computer Science, University of Udine, Udine, Italy*
[b]*Faculty of Computer Science, University of Murcia, Murcia, Spain*
[c]*R&D Department, Gap Srlu, Trieste, Italy*
[d]*Department of Mathematics and Computer Science, University of Ferrara, Ferrara, Italy*

Abstract. Evaluating in a correct, fair, systematic and reliable way the quality of the work is a central problem in modern business. Both from the psychological and the social point of view, this problem is very far away from being solved, let alone from being managed by a (semi-) automatic decision support system. In this paper we consider the case study of evaluating the operators' work quality in a medium-sized contact center, and, in particular, the problem of selecting the correct variables to be used in such an evaluation. Starting from a data set representative of the company's range and size of activities, that allowed no usable predictive model for evaluating the skills of the agents, we were able to devise a reproducible methodology, along with an *a posteriori* optimization process, to select the essential variables that should be used to objectively evaluate the quality of the agents' work. These results may be used in a support system helping the supervisors in evaluating the agents' performances. Moreover, we believe that our methodology may be extrapolated and reused in other comparable contexts characterized by the measurability of the human operators' performance.

Keywords: Feature selection, quality evaluation, contact center

1. Introduction

Evaluating the quality of the work that is being done by the employees is a central problem in modern business; such an evaluation should be correct, fair, systematic and reliable, and, to this end, it should be measurable. In this paper, we considered the problem of evaluating the quality of the work of operators (also called agents) in a contact center of average dimensions.

A *call center* is a set of resources, personnel, computers, and telecommunication equipment, which enable the delivery of services via the telephone. Thanks to the advancements in information technology, call centers are gradually evolving into *contact centers*, in which the phone-operator role of the agents is complemented, and sometimes substituted, by services offered through other technologies, such as faxing, instant messaging, web portals. Contact centers handle both *inbound* and *outbound* communications, with different purposes, including customer care and follow-up, as well as marketing and quality control. The distinguishing feature of a *multi-service* contact center is that the offered services vary over

*Corresponding author: Andrea Brunello, Department of Mathematics, Physics, and Computer Science, University of Udine, Udine, Italy. Tel.: +39 0432 558457; E-mail: andrea.brunello@uniud.it.

a wide range of possibly very different types (e.g., specific product client follow-up and travel reservation systems) [4]. The cornerstone of a contact center is the agent. An *agent* (or CSR, that is, *Customer Service Representative*) is the endpoint of a service, and his/her performances determine in large part the success rate of a transaction. The services' providers are usually able to identify a set of rules to evaluate an agent's performance; such rules are typically employed in both the training and the evaluation of an agent. Such a methodology, however, is specific for a service, and typically rules cannot be easily generalized. Therefore, the problem stands to identify a methodology that allows some sort of evaluation in a general way.

A large contact center generates vast amounts of data, which can be broadly classified as operational or service data. *Operational* data include all the technical information needed to reconstruct a detailed history of the events that take place during each communication, and include, for example, the dialled or dialling phone number, the agent(s) that has (have) been involved, possible call transfers, and timestamps. On the other hand, *service* data are specific to the particular service for which the contact has taken place, and may include, for example, all answers given by the interviewed subject during an outbound survey. Descriptive statistics of such a collection of data would be useless for the identification of the subset of variables that may or may not influence the performances of an agent. Instead, we applied a very large collection of feature selection mechanisms [20], along with a novel *a posteriori* "decision" making process in order to identify, if they exist, a subset of variables that may be thought of as *objective indicators* of the performances of an agent; in this context, *decision* making refers to effectively decide which results (subsets of features) are most indicated among those produced by the different mechanisms (and it should not be confused with "decision" in the context of management). To this end, we collected the cumulative data, for each agent, of a significant period of time and a significant range of different services, and we asked to three, independent, supervisors to evaluate each involved agent. Such an evaluation plays the role of the *expert's view* of this problem. We therefore transformed the task into a *feature selection for supervised classification* problem [10]. It turned out that this is an hard problem, as the classical classification model learning algorithms return very poor models when run on the entire range of attributes. This indicates an elevate noise rate that makes it very difficult to decide the best methodology *a priori*. Since our aim is to identify a set of meaningful attributes that may influence the judgment of an agent, *and not to build a classifier*, we formulated the problem as a *decision making* one among a very high number of selections. Intuitively, we proceeded as follows: we built a mechanism that allowed us to run a very wide range of combinations of search methods, evaluators, and model learners (categorized into univariate/multivariate, filter/wrapper, and deterministic/probabilistic), obtaining as many as 79 different *optimal* selections. Each selection has been used in three different classifier learners (two tree-based learners and one support vector machine), with four different performance indicators. Again, the problem at hand does not allow us to decide *a priori* which is the most correct indicator, as our results must be interpreted; therefore, we devised a complex automatic decision method, which may be generalized for a problem that results in n selections, for m classifiers along p measures. After a statistical pairwise analysis of the selections that allows us to exclude those that are not significant enough, our process (which, in essence, solves a multi-objective combinatorial problem), returns the k best selections. Each selection is a collection of attributes that appear to have *some* correlation with the expert's judgement of an agent; those variables that have been selected every or almost every time constitute the answer that we were looking for.

The paper is organized as follows. In Section 2 we give the necessary preliminaries concerning the entire range of methods that we have used, and we introduce multi-objective combinatorial problems. In Section 3 we describe our data set along with the single attributes and their domain-related meaning. In Section 4 we describe our methodology, and in Section 5 we give an overview of the results of our experiment, as well as a domain expert's interpretation of them, before concluding.

2. Background and related work

In this section, we briefly review the main methods and algorithms used in our experiments. The fact that these algorithms are all included in the WEKA data mining suite [11] is very convenient: being an open-source product, we had access to the Java classes of the state-of-the-art of each algorithm. In this way, we were able to design a simple script that allowed us to execute a very wide range of experiments on the same data in a systematic way.

2.1. Feature selection

Feature selection is the process of removing features from the data set that are irrelevant to task to be performed [20]. Its main aim is to facilitate data understanding, and to reduce storage and computation time requirements for model learning, while retaining a suitably high accuracy in representing the original features; nevertheless, we defined our problem as a feature selection problem *per se*, since we are searching for a specific subset of variables with a certain set of characteristics. Feature selection algorithms may be classified into several categories, depending on the specific criterion under consideration. According to whether the training set is composed of labelled instances or not, the selection may be, respectively, *supervised* or *unsupervised*. Methods in the former category seek for correlations between attributes and class label values, whereas those in the latter employ (usually, descriptive) statistical tests over attributes, such as, for example, a near-zero-variance test. Feature selection methods consist of four steps, namely *subset generation*, *subset evaluation*, *stopping criterion*, and *result validation*. The design of such steps entails the selection of: (i) a target to which to apply the procedure; (ii) a search strategy, to guide the incremental generation of the feature set; (iii) an evaluation strategy, which depends on the target type and, in the case of supervised methodologies, may imply choosing an actual classifier; (iv) an evaluation metric used to score the candidates.

2.2. Subset generation

Subset generation methods (also called **search strategies**) are used to guide the iterative generation of the feature set, in the space of all the possible combinations of features. They can be categorized into *deterministic* and *probabilistic* methodologies, the former giving back the same set of attributes if repeatedly performed, and the latter taking non-deterministic choices during execution. Moreover, in the former category it is possible to distinguish strategies according to their search direction: *forward* search strategies start with an empty attribute set, and then grow it; *backward* search strategies begin with an initial set consisting of all attributes, and proceed by discarding elements; *bi-directional* search strategies consider an initial point in the subset space, and then proceed in both directions; on the contrary, probabilistic (or *random*) strategies do not follow a predefined search direction, for example in optimization through genetic algorithms. In this experiment, among deterministic algorithms we considered: *BestFirst* [26], *GreedyStepwise* [29], *LinearForwardSelection* [13], and *InfoGain* [7], while the employed probabilistic algorithms are: *MultiObjectiveEvolutionarySearch* [16], *PSOSearch* [24], and *GeneticSearch* [12]. *BestFirst* implements *beam search*, and searches the space of attribute subsets by greedy hill climbing augmented with a backtracking capability; the amount of backtracking may be customized by specifying the beam width. It supports forward, backward, and bi-directional search directions. *GreedyStepwise* performs a greedy forward or backward search through the space of attribute subsets, stopping when the addition (forward direction) or deletion (backward direction) of any of the

remaining attributes results in a decrease in evaluation, thus, it has no backtracking capability. *LinearForwardSelection* is an extension of *BestFirst*, supporting simple forward or floating forward search directions. The latter considers a number of consecutive single-attribute elimination steps after each forward step, as long as this results in an improvement. The algorithm takes only a restricted number of k attributes into account, with the goal of reducing the number of evaluations performed during the search and producing a compact final subset, by two possible modes of operation: *fixed-set* or *fixed-width*. According to the former, all single attributes are initially ranked, and then the top-k are passed as input to forward selection. The latter employs a similar initial ranking criterion, starting the search with the top-k attributes; however, it maintains a fixed number of k candidates also in each of the subsequent forward selection steps, by adding further attributes from the initial ranked list (as long as any remain). Finally, the *InfoGain* strategy works by listing all features, ordered by their individual scores, as determined by measuring the information gain score with respect to the class. As far as probabilistic algorithms are concerned, *genetic* (or *evolutionary*) algorithms are the most common choice. Genetic algorithms were first proposed for attribute selection in [34], and are now considered an important tool for the selection of features [35]. They are inspired by the process of natural selection and, through the application of *elitist selection*, iteratively generate better and better solutions to optimization and search problems, by employing operators such as *mutation* and *crossover*. The goodness of a solution is determined through the use of one (single-objective) or more (multi-objective) *fitness* functions. In the present work, for the purpose of attribute selection (in those cases in which we choose multi-objective optimization), two objectives are optimized: the first one is chosen by the evaluator, and it is to be maximized, while the second one is the attribute subset cardinality, and it is to be minimized. The final output is given by the non-dominated solution in the last population having the best fitness score for the first objective. *MultiObjectiveEvolutionarySearch* is a multi-objective evolutionary algorithm that explores the attribute space using the elitist Pareto-based multi-objective evolutionary algorithm ENORA, while *GeneticSearch* implements the simple, classical Golberg's (single-objective) genetic algorithm for searching. Finally, *PSOSearch* explores the attribute space employing the Particle Swarm Optimization (PSO) algorithm. PSO optimizes a problem iteratively, trying to improve a candidate solution with regard to a given measure of quality. Similarly to evolutionary computation techniques, it considers a population of candidate solutions, called particles. Elements are moved around the search space according to mathematical formulae, considering each particle's characteristics and the overall "swarm knowledge", following an agent-oriented paradigm.

2.3. Subset evaluation

According to the target of the selection procedure, it is possible to classify evaluation strategies into *univariate* and *multivariate*. Strategies that belong to the former category evaluate attributes independently; as a result, they are computational less demanding than those that belong to the latter, which consider subsets of attributes as a whole. Moreover, multivariate approaches can also take into account complex relationships between features, such as redundancy. Here we have taken into consideration supervised methods, and, in particular, *filter* and *wrapper* models. Filter models are independent from the successive classifier learning phase, and are based only on general measures such as the correlation or consistency with the variable to predict. Filter techniques scale well with the size of data sets; however, since they ignore the classification performance, they might not always provide the best results [8,27]. Wrapper models, on the other hand, evaluate the predictive accuracy of the attribute set with a selected classifier. These techniques typically offer better results than filters, at the cost of being computationally more demanding, and more prone to overfitting [22]. We considered the following univariate filters:

(i) *GainRatioAttributeEval* [18], (ii) *SignificanceAttributeEval* [1], (iii) *SymmetricalUncertAttributeEval* [2], the univariate wrapper *ClassifierAttributeEval* [32], the following multivariate filters: (i) *CfsSubsetEval* [14], (ii) *ConsistencySubsetEval* [21], and the multivariate wrapper *WrapperSubsetEval* [19]. As far as univariate filters are concerned, *GainRatioAttributeEval* evaluates the worthiness of a single attribute by measuring its *gain ratio* value with respect to the class labels. Gain ratio is a well-known, commonly used assessment measure, calculated as the difference between the entropy of class distribution minus the conditional entropy of the classes given the values of the attribute, divided by the entropy of the attribute itself; *SignificanceAttributeEval* scores a single attribute by computing its *probabilistic significance* as a two-way function of its association to the class decision, and the intuition behind this algorithm is that if an attribute is significant with respect to the class labels, then it is expected that different sets of elements with complementary sets of values for the attribute will also belong to complementary sets of classes; finally, *SymmetricalUncertAttributeEval* evaluates the worthiness of a given attribute by measuring its *symmetrical uncertainty* with respect to the class. The univariate wrapper *ClassifierAttributeEval* scores an attribute by employing a user-selected classifier, evaluating its performance with respect to a specified evaluation metric (e.g., classification accuracy). For the purpose of this paper, we use it in conjunction with the classifiers *J48* (C4.5 [28]), *LibSVM* [5] and *RandomForest* [3]. *J48* is a Java implementation of the widely-used decision tree learner C4.5, which is known to be computationally efficient. The learning algorithm builds a decision tree from a set of labelled training instances in a recursive fashion, starting from the root node, by using the *information gain ratio* criterion. *LibSVM* is a library for *support vector machines* learning. A support vector machine is a supervised machine learning algorithm, which can be used for both regression and (typically binary) classification problems. Each instance is mapped to a point in n-dimensional space, where n is the number of features characterizing the instance. Then, in a binary classification setting, a hyperplane is constructed, that optimally divides the instances in homogeneous groups with respect to the class labels. *RandomForest* is an *ensemble learning* method which constructs a forest of random trees, for classification or regression purposes. A typical problem of decision trees is their propensity to overfit, if not properly pruned: in the literature, they are regarded as models having low bias, but high variance. In *RandomForest* each tree is built from a separate part of the same training set, reducing the variance, thus contrasting the tendency of a large, single tree to overfit. Given a new instance to classify, the final output is obtained by combining the results given by the different trained models. The multivariate filter *CfsSubsetEval* evaluates the worthiness of an entire subset of features by considering the individual predictive power of each attribute, together with the degree of redundancy between them; subsets containing attributes that are highly correlated with the class, and not strongly correlated with one another, are preferred. On the contrary, *ConsistencySubsetEval* scores a subset of features as a whole, by projecting the training instances according to the attribute subset, and considering the consistency of class values in the obtained instance sets. Finally, the multivariate *WrapperSubsetEval* scores a set of attributes by employing a user-selected classifier, evaluating its performance with respect to a specified evaluation metric (e.g., classification accuracy). Again, for the purpose of this paper, we use it in conjunction with *J48*, *LibSVM*, and *RandomForest*.

2.4. Evaluation metrics

Evaluation metrics are used to assign a numerical score to each candidate during the feature selection process. The metrics employed in the present work include: *accuracy* (for classification), *weighted area under ROC* (for classification), the *root mean squared error* (for regression and binary classification), and the *model size*. The *accuracy* (ACC) measures the amount of correctly labelled instances,

as classified by a model. It is given by the ratio between the number of correctly classified instances and the number of total instances. The *weighted area under ROC* (WAUC) metric is calculated on a ROC curve [9,23], which is a graphical representation of the *sensitivity* versus *specificity* for a classifier system, obtained by varying the model class discrimination threshold. The AUC value belongs to the interval [0,1]; a score of 1 represents the perfect classifier, while 0.5 is typical of a random classification behaviour; in the weighted version (WAUC), this number is computed taking into account also the cardinality of each class. The *root mean squared error* (RMSE) measures the difference between values predicted by a model and the values actually observed. Finally, the *model size* (MS) simply measures how big a classification/regression model is. Typically, ACC and WAUC are to be maximized, while RMSE and MS are to be minimized.

2.5. Multi-objective combinatorial optimization

Optimization [17] indicates the process of selecting a best element with respect to some criteria; *mathematical programming* is the discipline that studies the theory, the algorithms, and the techniques to represent and solve optimization problems. While some of the subset generation methods described above (precisely, the probabilistic subset generation algorithms) are defined as *multi-objective optimization* (MOO) problems [6], our interest here is in defining a decision making process as such. A *minimizing* MOO problem can be formally defined as:

$$Min\ f_i(\bar{x})$$

for $i = 1, \ldots, l$, where each f_i may be linear or non-linear. Variables may be continuous or discrete; in the latter case, the problem is an optimization *combinatorial* problem. In combinatorial problems, we are looking for objects in a countable set $\mathcal{C}$, typically the set of integers, sets, permutations, or graphs; the variables $\bar{x} \in \mathcal{C}^k$ is the set of *decision* variables. Optimization problems may be minimizing, maximizing, or both. A solution $\bar{x} \in \mathcal{C}^k$ is said to be a *non dominated* (or *Pareto optimal*) if and only if there exists no $\bar{y} \in \mathcal{C}^k$ for which: (i) there exists $1 \leqslant i \leqslant l$ such that $f_i(\bar{y})$ improves $f_i(\bar{x})$, and (ii) for each $j \neq i$, $f_j(\bar{x})$ does not improve $f_j(\bar{y})$. The set of non dominated solutions from $\mathcal{C}^k$ is called *Pareto front*. Solving a MOO entails finding the Pareto front, or an approximation to it; depending on the particular problem, one may later choose a specific solution from the front. A MOO with only linear functions is called *linear programming problem*, for which efficient algorithms exist to obtain the optimal solution (i.e., the *simplex method* [33]). If at least one of the functions is nonlinear, the MOO is a *nonlinear programming problem* [33]. A nonlinear programming problem in which the objectives are arbitrary functions is, in general, intractable, and, typically, sub-optimal search algorithms are used to approach them; these are precisely those recalled above in this section, and include branch and bound, heuristics and metaheuristics such as evolutionary algorithms and PSO.

3. Data sets and problem definition

The data we used have been provided by Northern Italy company Gap Srlu, and consist of the cumulative performances of 77 agents over a period of 6 months. Contacts in Gap are managed and organized as follows. The *flux* of information is categorized into *inbound* (that is, contacts that Gap receives, such as phone calls) and *outbound* (i.e., surveys made by Gap). Each of these is classified by commissions: a *commission* is the unit of contract between Gap and a client (i.e., the ACME airline company commissions to Gap the phone ticket selling service for their customers), and each commission may be declined

Table 1
Variables related to the agent and variables related to the switching frequency of the agent

Attribute	Semantics
Agent related variables	
Agent_seniority	# of days of service of the agent
Agent_gender	Whether the agent is a male or female
Agent_age	Age of the agent
Agent_education	Level of education of the agent
Agent_skill	Weekly avg. and var. of agents' skill
Diversity variables	
Num_sessions	Daily avg. and var. of the # of distinct sessions
Num_commissions	Daily avg. and var. of the # of distinct commissions
Switch_index	Daily avg. and var. of (all) switches
Switch_index_flow_type	Daily avg. and var. of flow switches
Switch_index_ser_type	Daily avg. and var. of service switches
Switch_index_ser_same_type	Daily avg. and var. of sub-service type switches
Icc_inbound_av	Daily avg. and var. of avg. icc index in inbound
Icc_outbound_av	Daily avg. and var. of avg. icc index in outbound
Icc_inbound_var	Daily avg. and var. of var. icc index in inbound
Icc_outbound_var	Daily avg. and var. of var. icc index in outbound

into several services. A *service* is a specific type of interaction that the client wants Gap to operate with (i.e., ACME wants Gap to deal with ticket selling but not lost-and-found), and each service includes several *sub-types* (i.e., ACME ticket selling includes a channel for information, a channel for reservation managing, and so on). For the purpose of this experiment, we considered phone-based communications only. Of all agents, 56 were employed for outbound, inbound, and backoffice services, while the remaining 21 had no inbound communications. The work of all agents has been described via 69 attributes, while for those agents with at least some inbound communications over the analyzed period, we were able to add 6 more features (that make sense for inbound communications only). Compared to previous data mining experiments on contact center databases, the quality of the information at our disposal is considerably higher. Not only did previous experiments such as [25] made no use of feature selection; they did also operate on a very restricted set of attributes, consequently limiting the significance of their results. Moreover, all previous experiments, including [15,30], were not designed to evaluate the performances of the agents.

For a better understanding, the set of variables common to both data sets (the one containing the cumulative performance indicators of all agents and the one containing the cumulative performance indicators of only those agents that had inbound communications) can be classified into several categories, depending on the particular aspect they describe. The first category is *agent related variables* (see Table 1 – top),[1] and includes their seniority (from 6 months to 5 and an half year), their gender (31 males versus 46 females), their age (from 19 to 65 years old), their level of education (from 1-minimum compulsory education, to 5-university degree or more), and their *skill* average and variance: Gap has internally engineered a skill-function that takes into account several aspects, recomputed weekly for each agent, and of which we consider the average and the variance over the entire period. A second category of variables is *work's diversity*, by means of which we want to measure how heterogeneous has been the agent's work in the analyzed period. This category includes the number of *distinct sessions*[2] *and distinct*

[1] Unless otherwise specified, every numeric variable is in fact a pair of variables that takes into account average and variance of each aspect.

[2] A *session* is the most basic unit of work done by the agent, to which it is possible to assign a result, for example a phone call.

Table 2
Variables related to the agent's work distribution and heterogeneity, and turn distribution

Attribute	Semantics
Work distribution variables	
Management	Daily avg. and var. of # min. working
Management_inbound	Daily avg. and var. of # min. working on inbound comm.
Management_outbound	Daily avg. and var. of # min. working on outbound comm.
Management_backoffice	Daily avg. and var. of # min. working on backoffice
Fraction_inbound	Daily avg. and var. of the % of min. on inbound
Fraction_outbound	Daily avg. and var. of the % of min. on outbound
Fraction_backoffice	Daily avg. and var. of the % of min. on backoffice
Available_sessions	Daily avg. and var. of the # of available sessions
Available	Daily avg. and var. of # min. available
Break_sessions	Daily avg. and var. of the # of break sessions
Break	Daily avg. and var. of # min. on break
Inactive_sessions	Daily avg. and var. of the # of inactive sessions
Inactive	Daily avg. and var. of # min. inactive
Turn distribution variables	
Turn_duration	Daily avg. and var. of turn length in # min.
Fraction_weekend	Fraction of weekend workdays
Fraction_night	Daily avg. and var. of the % of min. working during nights
Fraction_morning	Daily avg. and var. of the % of min. working during mornings
Fraction_early_afternoon	Daily avg. and var. of the % of min. working during early aft.
Fraction_late_afternoon	Daily avg. and var. of the % of min. working during late aft.
Fraction_evening	Daily avg. and var. of the % of min. working during evening
Inactivity_time	Fraction of total inactivity time over total turn duration
Availabile_time	Fraction of total availability time over total turn duration
Break_time	Fraction of total break time over total turn duration

commissions the agent has worked on, the daily frequency of *context switches* (that takes into account switching between flows, or services, or service sub-types, weighted: farthest jumps weight the most), the daily frequency of *flow switches* (inbound vs. outbound), the daily frequency of *service switches*, and the the daily frequency of *sub-type switches*, and it is given in Table 1 (bottom). Moreover, we have taken into account how the agents' work has been *distributed* (Table 2 – top), by including the average and the variance over days of the number of minutes during which he/she has been effectively working (*management*), on inbound (*management inbound*), on outbound (*management outbound*) communications, or on backoffice (*management backoffice*), along with their fraction on the entire workload, that takes into account how many times the agent has declared him/herself *available* (in idle state), for how many minutes in total, *on break*, and for how many minutes, and *inactive* (that is, on break or available). The distribution takes also into account the *icc* index, which is an internal evaluation of the importance, complexity and criticality of the service being worked on. Finally, Table 2 (bottom) shows the variables relative to agents' *turns distribution*, that take into account in which part of the day and of the week each agent's shifts are mainly scheduled, as well as the fraction, over the entire observed period, of break, available, and inactive time of the agent.

Six more attributes have been considered for those agents whose job during the observed period included inbound communications. Such variables take into account the structure, the understandability, and the type of call-related *notes* written by the agent. These may be *abbreviated*, *articulated*, *non-articulated*, *domain-related*, *hybrid*, or *unrecognizable*.

Our choice of attributes naturally led to two distinct data sets, hereafter called ALL_AGENTS and INBOUND_AGENTS; the former contains 69 attributes and 77 instances, and the latter contains 75 attributes and 56 agents. Both data sets have been enriched with a variable that describes the agent

performance value. This has been obtained by asking to three independent supervisors a fair judgement of each agent *to the best of their expertise*. Their judgement, on a scale from 1 (lowest) to 5 (highest), takes into account the overall impression of the agents and their performances; then, the three votes have been combined into a single one by averaging them. The purpose of this work is to answer the following question: *which are, if they exist, the performance-related variables that influence the expert judgment on an agent?*

4. Methodology

For each of the two data sets we applied a simple preprocessing methodology. First, we have replaced all the missing values with their respective mean; to this end, the procedure *ReplaceMissingValues* from the *weka.filters.unsupervised.attribute* package has been used. Second, we have searched for those features with too small variation by using *RemoveUseLess* from the same package: no features have been eliminated via this process, indicating that, potentially, all of them might influence the agent judgment.

After the preprocessing, we have systematically applied 79 different feature selection mechanisms,[3] as in Fig. 1. Each mechanism is the result of a specific choice among the subset generation algorithms, the subset evaluation algorithms, and the evaluation metric (all explained in Section 2). In particular, among all choices, consider the multivariate wrapper and filter obtained by using the multi-objective evolutionary algorithm as search strategy: independently from the chosen measure (either accuracy, area under ROC curve, model size, or RMSE), an (internal) decision making process is necessary; indeed, by optimizing two parameters, namely the number of selected attributes and the performance indicator, the result is a *population* of solutions. In order to choose one of them (out of 30 executions with population size 100 for 100 evaluations, see, e.g. [16]), we applied a particular case of cross validation. In particular, with less than one hundred instances, the best choice is the so-called *leave-one-out* cross validation [31]; the best individual in term of the chosen measure over 10 runs has been selected.

The result of this process is composed of 79 different selections of attributes, each one of them optimized following a different criterion. The test phase consisted of training a model with each of the 79 corresponding data sets, via: (i) a decision tree learner (*J48*); (ii) a support vector machine (*LibSVM*); (iii) a random forest learner (*RandomForest*). For each of the resulting models, we measured, after a leave-one-out cross-validation test: (i) accuracy (ACC); (ii) (weighted) area under the ROC curve (WAUC); (iii) root mean squared error (RMSE); (iv) serialized model size (MS). Finally, we applied the following decision making strategy. In order to highlight possible significant statistical differences among the resulting selections, we performed a non-parametric *Friedman test* [36] with significance level $\alpha = 0.05$ for each of the measures. Second, we applied multi-objective combinatorial optimization, as seen in Section 2. A generic method for multi-objective optimization that can be used to identify the "best" data bases among n data bases evaluated with m classifiers consists in simply optimizing:

$$f_i(x) = \frac{1}{m}\sum_{j=1}^{m} M_i(x,j),\ i = 1,\ldots,l$$

where $M_i(x,j)$ is the value of the performance metric M_i for the classifier $j \in CL$ evaluated in the data base $x \in DB$, and l is the number of performance metrics. In our particular case, we have that the

[3]The hardware that we have used is a machine with 8 processors Intel Xeon X7550 @ 2.00 GHz, RAM 1TByte at 1067 MHz and storage Lustre Distributed File System v2.5.2; interconnection network: Infiniband QDR (40 Gbps).

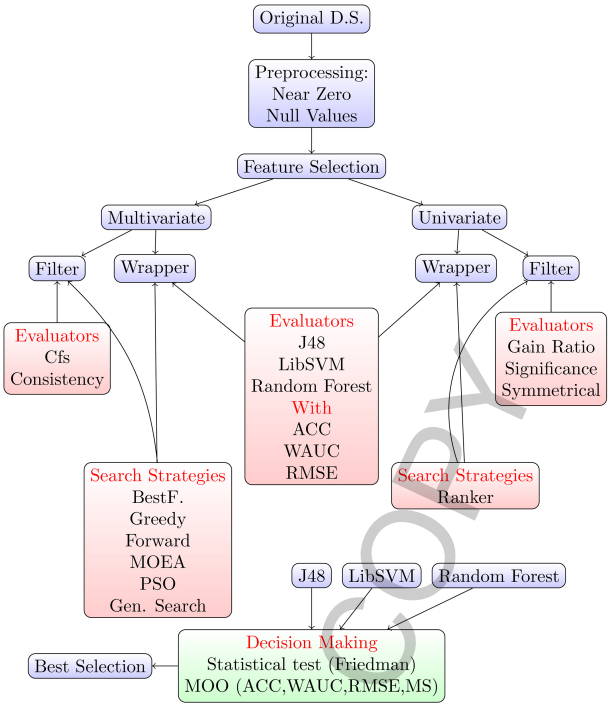

Fig. 1. Proposed methodology for feature selection.

problem becomes as in the following set of formulas:

$$\begin{cases} Max & f_1(x) = \frac{1}{m}\sum_{j=1}^{m} ACC(x,j) \\ Max & f_2(x) = \frac{1}{m}\sum_{j=1}^{m} WAUC(x,j) \\ Min & f_3(x) = \frac{1}{m}\sum_{j=1}^{m} RMSE(x,j) \\ Min & f_4(x) = \frac{1}{m}\sum_{j=1}^{m} MS(x,j) \end{cases}$$

As the last step, we considered the best selections obtained in this way, and we analyzed the corresponding features: the most common ones are those that, in fact, influence the experts' judgments. The entire methodology, applied to both ALL_AGENTS and ALL_INBOUND, is displayed in Fig 1. Algorithm 1 presents the pseudo code summarizing the overall approach.

5. Analysis of the results

The Friedman test showed no statistical differences among the selections, for both the (original) data sets. This means that we solved the MOO (optimizing the four chosen objectives) among the 79 selections, once for each problem. Recall that our problem presented a very low susceptibility to classification, and that we are *not* interested in building a classifier, but in identifying a meaningful subset of variables. The MOO objectives are designed precisely to this aim (see Section 4), as they optimize the *average* performance degree without committing to a specific classifying model learner. As a consequence, we are not interested in their absolute values. It turns out that 12 solutions from the ALL_AGENTS, and 5 from the INBOUND_AGENTS problem are not dominated; these are shown in Table 4. Several conclusions may be drawn from inspecting Table 4:

Algorithm 1 Pseudo-code of the generic analysis approach

```
1: function ANALYZE
2:     Missing values imputation
3:     Remove attributes with too small variation
4:     Perform feature selection methods
5:     Classification with J48, RandomForest and LibSVM over the reduced databases
6:     Statistical test (Friedman) to determine if the observed differences are statistically significant,
       and to eliminate the worse selections
7:     Multi-objective combinatorial optimization over the remaining reduced databases with signifi-
       cant statistical differences
8:     Analyze features of the non-dominated reduced databases
```

Table 3
Variables related to the agent's notes

Attribute	Semantics
Notes' structure variables	
Fraction_abbreviated	Fraction of abbreviated notes
Fraction_articulated	Fraction of articulated notes
Fraction_non_articulated	Fraction of non articulated notes
Fraction_hybrid	Fraction of hybrid notes
Fraction_unrecognized	Fraction of unrecognized notes
Fraction_domain	Fraction of domain-related notes

- Wrapper methods have shown better performance than filter methods;
- Multivariate methods have shown better performance than univariate methods;
- Among wrapper methods, tree-based ones have shown better performances than every other learning methods, especially when combined with our evolutionary search strategy;
- The run time of *RandomForest* is acceptable in wrapper methods after limiting the number of iterations to 10, and the method is not very sensitive to the variation of its parameters. However, *RandomForest* is prone to overfitting and generates larger regression models.

In synthesis, multi-objective evolutionary search with ENORA resulted to be the most successful search strategy, and RandomForest to be the most precise classifier to be used in wrappers.

Having reduced the number of solutions to 17, we can now analyze them, and, in particular, we can examine the attributes that have been selected. Table 5 shows the most common ones, which must be interpreted as those having the highest influence on the judgement. A first, immediate, observation is that separating our initial objective into two sub-problems has been the right choice: there exists a substantial difference in the results of the selections, meaning that the agents that work on both inbound and outbound communications behave in a substantially different way from those who concentrate solely on outbound. Focusing on the group of variables that, apparently, may influence the judgement on all agents, we notice that these are taken exclusively from the set of variables that describe the work and the turn distribution. On top of this observation, we notice that *break and inactivity periods* (in particular, the average number of breaks over a single day of work, and the total inactivity time over the entire observed period) are the most influential characteristics. Moreover, the *average workload* seems to have some relevance in determining the quality of an agent. Finally, it also seems that the average amount, his/her *turn distribution in the mornings and in the nights* varied over the observed period had played a role in determining the judgement.

Table 4
Non-dominated solutions

Eval.	Search str.	Meas.	Avg. ACC	Avg. WAUC	Avg. RMSE	Avg. MS
All agents						
LibSVM	BestF.	ACC	52.88	0.65	0.42	18630.69
LibSVM	BestF.	WAUC	52.88	0.65	0.42	18630.69
Rand. F.	BestF.	ACC	52.82	0.65	0.44	16947.65
Rand. F.	BestF.	RMSE	54.46	0.66	0.43	18205.39
LibSVM	Forward	WAUC	54.92	0.67	0.41	19133.20
J48	Evol.	RMSE	56.29	0.67	0.42	19741.76
LibSVM	Evol.	RMSE	54.15	0.65	0.42	19098.85
Rand F.	Evol.	ACC	55.81	0.66	0.43	17201.62
Rand F.	Evol.	RMSE	56.39	0.68	0.42	18286.98
LibSVM	PSO	RMSE	55.56	0.67	0.42	19924.34
J48	InfoGain	RMSE	32.51	0.50	0.48	4322.00
Rand F.	InfoGain	RMSE	32.51	0.50	0.48	4322.00
Inbound agents						
Rand. F.	BestF.	RMSE	41.60	0.50	0.4555	5410.95
LibSVN	Genetic	ACC	55.11	0.66	0.39	14920.77
J48	Evol.	ACC	61.11	0.70	0.39	13588.77
Rand. F	Evol.	WAUC	60.65	0.70	0.39	13942.54
Rand F.	InfoGain	RMSE	31.84	0.5	0.49	4337

Table 5
Most commonly selected attributes

Attribute	Relative frequency
Most commonly selected attributes: ALL_AGENTS problem	
Avg_break_sessions	3
Inactivity_time	3
Available_time	2
Avg_management	2
Break_time	2
Var_available	2
Var_fraction_morning	2
Var_fraction_night	2
Var_available_sessions	2
Most commonly selected attributes: INBOUND_AGENTS problem	
Agent_education	4
Avg_management	4
Avg_break_sessions	4
Var_fraction_morning	4
Agent_gender	3
Agent_age	3
Avg_num_commissions	3
Avg_fraction_inbound	3
Var_num_commissions	3

Focusing on on the results for the group of agents that had both inbound and outbound work, we discover some interesting differences. Unlike the previous case, agent's *education level, age, and gender* do play a role in determining the quality of his/her work. This makes perfect sense: inbound sessions are essentially different from outbound ones, and it emerges that education and age may make the difference. The average *number of break sessions* during a working day still has a relevant role (which means that this aspect is transversal to the type of agent), as well as the variance of his/her *turn distribution in the mornings*.

Finally, we notice that the structure of the written notes taken by the inbound agents seems not to have any essential role in determining the overall impression on them. Similarly, and maybe more interestingly, in both groups, the skill level of the agents (determined internally in Gap), as well as the entire range of indicators that depend on the heterogeneity and on the relative importance of the work assigned to the agent, seem not to influence the judgement in any way. This may depend, among other reasons, from the fact that the skill level is an internal evaluation based on technical aspects, and it is independent from the judgment that we used as class (and that we wanted to predict).

6. Conclusions

The problem of evaluating in a correct, fair, systematic and reliable way the quality of the work is central in modern business. As a case study, we considered a group of customer service representatives, or agents, in a medium-sized contact center, and we associated a very subjective evaluation of their performance in a six-months period (obtained by combining three, independent, expert evaluations) with a synthesis of the operational and service data generated by their activity in the same period. Our aim was to identify the subset of parameters that (implicitly) influenced their evaluation, and therefore help the experts in designing a (semi) automatic system for evaluating the agents. Since such a problem is not susceptible of a classical learning approach, we applied a very large collection of feature selection mechanisms along with a novel *a posteriori* decision making process in order to identify optimal subsets of variables that may be thought of as objective indicators of the performances of an agent. We found, first, that those agents that work on both inbound and outbound communications behave in a substantially different way from those concentrating solely on the outbound. Moreover, we discovered that for a generic agent (regardless of him/her being assigned inbound services or not), work and turn distributions seem to have some influence in his/her performance, as well as break and inactivity periods; also, the average workload and turn distribution in the mornings and in the nights seems to have some relevance in determining the quality of their work. Interestingly, education level, age, and gender of an agent has some influence only on those assigned to inbound work.

We believe that our methodology may be extrapolated and reused in other comparable contexts characterized by the measurability of the human operators' performance.

Acknowledgments

This study was supported by computing facilities of Extremadura Research Centre for Advanced Technologies CETA-CIEMAT), funded by the European Regional Development Fund (ERDF). CETA-CIEMAT belongs to CIEMAT and the Government of Spain.

References

[1] A. Ahmad and L. Dey, A feature selection technique for classificatory analysis, *Pattern Recognition Letters* **26**(1) (2005), 43–56.
[2] S.I. Ali and W. Shahzad, A feature subset selection method based on symmetric uncertainty and ant colony optimization, *International Journal of Computer Applications* **60** (2012), 5–10.
[3] L. Breiman, Random forests, *Machine learning* **45**(1) (2001), 5–32.
[4] A. Brunello, A data warehouse for a contact center with multiple channels and skills, Master's thesis, University of Udine, 2015.

[5] C.-C. Chang and C. J. Lin, Libsvm: A library for support vector machines, *ACM Transactions on Intelligent Systems and Technology* **2**(3) (2011).
[6] Y. Collette and P. Siarry, Multiobjective Optimization: Principles and Case Studies, Springer Berlin Heidelberg, 2004.
[7] S. Dinakaran and P.R.J. Thangaiah, Role of attribute selection in classification algorithms, *International Journal of Scientific & Engineering Research* **4**(6) (2013), 67–71.
[8] R.O. Duda, P.E. Hart and D.G. Stork, Pattern classification, John Wiley & Sons, 2012.
[9] T. Fawcett, An introduction to ROC analysis, *Pattern Recognition Letters* **27**(8) (2006), 861–874.
[10] U.M. Fayyad, G. Piatetsky-Shapiro and P. Smyth, From data mining to knowledge discovery: An overview, in: U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy, eds, *Advances in Knowledge Discovery and Data Mining*, AAAI, 1996, pp. 1–34.
[11] E. Frank, M.A. Hall and I.H. Witten, The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, 4 edition, 2016.
[12] D.E. Goldberg, Genetic algorithms in search, optimization and machine learning, Addison-Wesley, 1989.
[13] M. Gutlein, E. Frank, M. Hall and A. Karwath, Large-scale attribute selection using wrappers, in: *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining*, 2009, pp. 332–339.
[14] M.A. Hall, Correlation-based feature selection for machine learning, PhD thesis, The University of Waikato, 1999.
[15] F. Jiménez, E. Marzano, G. Sánchez, G. Sciavicco and N. Vitacolonna, Attribute selection via multi-objective evolutionary computation applied to multi-skill contact center data classification, in: *Proc. of the IEEE Symposium on Computational Intelligence in Big Data (IEEE CIBD 15)*, IEEE, 2015, pp. 488–495.
[16] F. Jiménez, G. Sánchez, J. García, G. Sciavicco and L. Miralles, Multi-objective evolutionary feature selection for online sales forecasting, Neurocomputing, 2016.
[17] H. Jongen, Optimization Theory, Kluwer Academic Publishing, 2004.
[18] A.G. Karegowda, A.S. Manjunath and M.A. Jayaram, Comparative study of attribute selection using gain ratio and correlation based feature selection, *International Journal of Information Technology and Knowledge Management* **2**(2) (2010), 271–277.
[19] R. Kohavi and G.H. John, Wrappers for feature subset selection, *Artificial Intelligence* **97**(1–2) (1997), 273–324.
[20] H. Liu and H. Motoda, Feature selection for knowledge discovery and data mining, volume 454, Springer Science & Business Media, 2012.
[21] H. Liu and R. Setiono, A probabilistic approach to feature selection – a filter solution, in: *Proceedings of the 13th International Conference on Machine Learning (ICML)*, Vol. 96, 1996, pp. 319–327.
[22] H. Liu and L. Yu, Toward integrating feature selection algorithms for classification and clustering, *IEEE Transactions on Knowledge and Data Engineering* **17**(4) (2005), 491–502.
[23] C.E. Metz, Basic principles of ROC analysis, in: *Seminars in Nuclear Medicine*, Vol. 8, 1978, pp. 283–298.
[24] A. Moraglio, C. Di Chio, J. Togelius and R. Poli, Geometric particle swarm optimization, in: *Proceedings of the 10th European Conference on Genetic Programming*, 2007, pp. 125–136.
[25] M. Paprzycki, A. Abraham, R. Guo and S. Mukkamala, Data mining approach for analyzing call center performance, in *Proc. of the 17th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE)*, 2004, pp. 1092–1101.
[26] J. Pearl, Heuristics: Intelligent Search Strategies for Computer Problem Solving, Addison-Wesley Publishing Company, 1984.
[27] H. Peng, F. Long and C. Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**(8) (2005), 1226–1238.
[28] J.R. Quinlan, C45: Programs for Machine Learning, Morgan Kaufmann, 1992.
[29] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, Prentice-Hall, 2 edition, 2003.
[30] S. Salcedo-Sanz, M. Naldi, A. Pérez-Bellido, J. Portilla-Figueras and E. Ortíz-García, Evolutionary optimization of service times in interactive voice response systems, *IEEE Trans. Evolutionary Computation* **14**(4) (2010), 602–617.
[31] C. Sammut and G. Webb, Encyclopedia of Machine Learning, Springer Publishing Company, Incorporated, 2011.
[32] R. Schäfer, Accurate and efficient general-purpose boilerplate detection for crawled web corpora, Language Resources and Evaluation, 2016, 1–17.
[33] A. Schrijver, Theory of Linear and Integer Programming, Wiley, 1986.
[34] W. Siedlecki and J. Sklansky, A note on genetic algorithms for large-scale feature selection, *Pattern Recognition Letters* **10**(5) (1989), 335–347.
[35] H. Vafaie and K. De Jong, Genetic algorithms as a tool for feature selection in machine learning, in: *Proceedings of the 4th International Conference on Tools with Artificial Intelligence (TAI)*, 1992, pp. 200–203.
[36] D. Zimmerman and B. Zumbo, Relative power of the wilcoxon test, the friedman test, and repeated-measures anova on ranks, *Journal of Experimental Education* **62** (1993), 75–86.