Journal of Behavioral and Experimental Economics 109 (2024) 102164

Contents lists available at ScienceDirect

Journal of Behavioral and Experimental Economics

journal homepage: www.elsevier.com/locate/jbee

Predicting compliance: Leveraging chat data for supervised classification in experimental research

Carina I. Hausladen [a,1,*], Martin Fochmann [b], Peter Mohr [c]

[a] Computational Social Science, ETH Zurich, Switzerland
[b] Department of Finance, Accounting, Controlling, and Taxation, Freie Universität Berlin, Berlin, Germany
[c] Behavioral Economics, esp. Neuroeconomics, Freie Universität Berlin, Berlin, Germany

ARTICLE INFO

Dataset link: https://github.com/carinahausladen/PredictingCompliance

JEL classification:
C55
C92
D83

Keywords:
Chat data
Supervised classification
Experimental research
Tax evasion
Compliance

ABSTRACT

Behavioral and experimental economics have conventionally employed text data to facilitate the interpretation of decision-making processes. This paper introduces a novel methodology, leveraging text data for predictive analytics rather than mere explanation. We detail a supervised classification framework that interprets patterns in chat text to estimate the likelihood of associated numerical outcomes. Despite the unique advantages of experimental data in correlating textual and numerical information for predictive modeling, challenges such as limited sample sizes and potential data skewness persist. To address these, we propose a comprehensive methodological framework aimed at optimizing predictive modeling configurations, particularly in small experimental behavioral research datasets. We also present behavioral experimental data from a preregistered tax evasion game (n=324), demonstrating that chat behavior is not influenced by experimenter demand effects. This establishes chat text as an unbiased variable, enhancing its validity for prediction. Our findings further indicate that beliefs about others' dishonesty, lying attitudes, and risk preferences significantly impact compliance decisions.

1. Introduction

Behavioral experimental research aims to identify the factors that influence behavior. While regression analysis has traditionally been the standard method used for theory development, it may not always be effective for practical problem-solving. Think of tax compliance or compliance with rules in organizations. Audit systems can either randomly check for submission or apply some kind of risk management. These systems aim at identifying individuals or situations with a high probability of non-compliance to target their auditing resources to these cases. Doing so increases the share of identified non-compliance while keeping their auditing resources constant.

Behavioral experimental data offers a promising opportunity for making predictions; however, the research community has yet to fully explore its potential in this regard. Most behavioral experiments are structured so that (numeric) decision data is collected alongside process data, such as text generated through group chats. A vital consideration is that these experiments often incentivize participants to engage in text chats directly related to their decisions, ensuring dependence between the chat text and the numeric variable. Consequently, this creates a dataset where the numeric decision serves as a label for the chat text, positioning it as a valuable resource for predictive modeling.

This unique characteristic makes experimental data an excellent candidate for supervised learning, which is not often the case with real-world text data, as it rarely possesses this characteristic. Although behavioral research often has access to gold-standard labeled data, this property remains largely underutilized. Instead, text data is typically treated as process data and not directly connected to the decision data. For example, some studies (van Elten & Penczynski, 2020; Fochmann, Kocher, Müller, & Wolf, 2019; Kocher, Schudy, & Spantig, 2018; Mónica Capra, 2019) assign hand-assigned labels derived from theoretical reasoning to chat texts, while others (Andres, Bruttel, & Friedrichsen, 2019; Mónica Capra, 2019) use word clouds to distinguish between treatment groups. Only a few studies (Arad & Penczynski, 2018; Burchardi & Penczynski, 2014; Georgalos & Hey, 2019; Penczynski, 2019) have leveraged (semi-)supervised learning to assign labels to text data. However, their approach differs from ours as they assign labels by hand and do not directly connect the decision data with the process data, which is the focus of our paper.

* Correspondence to: Stampfenbachstrasse 48, 8006 Zurich, Switzerland.
E-mail address: carinah@ethz.ch (C.I. Hausladen).

[1] Research conducted during the tenure as a Ph.D. student at the University of Cologne and the Max Planck Institute for Research on Collective Goods, Bonn, Germany.

https://doi.org/10.1016/j.socec.2024.102164
Received 25 June 2023; Received in revised form 19 November 2023; Accepted 4 January 2024
Available online 6 January 2024

However, experimental data often poses challenges, such as small sample sizes and imbalanced label distributions. To address these issues, we present a methodological framework that systematically compares various classification setups, leveraging the unique characteristics of the dataset. Our approach tests standard NLP classification setups and tailors the methodology to construct dependent variables in different ways and evaluate predictive performance based on these variables.

In addition, we propose a novel approach to ensure the robustness and generalizability of the classifier. Typically, machine learning classifiers are validated by splitting the available data into training and testing sets, with the testing set used to evaluate the classifier's *out-of-sample* performance. If the classifier performs well on the testing set, the machine learning community generally considers it predictive and suitable for use in various contexts. However, even the most advanced machine learning models may not be able to account for all possible scenarios due to a lack of training data. One approach to addressing this challenge is to test the model on increasingly challenging test datasets, simulating new and unpredictable scenarios. While these datasets can be generated artificially, they may not capture the complexity and nuances of real human behavior. Here, we suggest that the experimental community is uniquely positioned to generate more realistic and challenging test datasets. Experiments offer a highly controlled behavioral setting where experimenters can vary different parameters and collect decision data. By introducing additional variations in experimental design and decision data, we can create increasingly challenging datasets that can serve as a robust testbed for machine learning classifiers. To generate a robust test set for evaluating our classifier's *out-of-context* performance, we design and conduct a behavioral experiment that intentionally varies three key parameters from the original experiment on which the initial classifier was trained. By deliberately manipulating these parameters, we can create a challenging test set that pushes the limits of our classifier's ability to accurately predict outcomes in novel contexts.

To summarize, our paper makes several contributions to the research community. Firstly, we propose utilizing behavioral experimental data for supervised learning. Secondly, we present a methodological framework tailored to the unique properties of experimental data. Thirdly, we propose and test a novel approach to ensure the robustness and generalizability of the classifier, highlighting the experimental community's unique position to achieve this. Lastly, we collect and evaluate behavioral experimental data and investigate the linguistic predictors of (non)compliance.

The structure of this paper is as follows: In the second section, we outline the general approach for setting up and comparing machine learning models. This section explains the process of translating text data into numerical form for embedding, the selection of classifiers, and the methods used for evaluating and comparing the models. In the third section, we describe the experiments conducted to train and test different machine learning models, including the model specifications and results. Next, we analyze the newly collected behavioral experimental data. Finally, we discuss our findings, draw practical implications, and suggest avenues for future research before concluding the paper.

2. Methods: Model architecture

Supervised learning is a machine learning task where the objective is to predict a dependent variable y by a set of independent variables, represented as a vector/matrix of X. This is similar to classical statistical approaches, such as linear or logistic regressions, which are frequently used in economics. However, unlike these approaches, supervised learning can also be applied to text data such as written chat messages to predict the dependent variable y.

Text classification is a well-studied problem in the field of Natural Language Processing, resulting in a wide range of feature engineering techniques and classifiers. However, not all of these methods suit the experimental behavioral data we aim to investigate. The following section describes the different configurations we evaluated to train a classifier that can effectively learn the association between chat text and decision data.

2.1. Exploiting unique characteristics of behavioral data for improved classification performance

Experimental behavioral data exhibit unique characteristics that can be harnessed to improve classification performance. Consequently, we generate several variations of the text data and decision data and test which variation yields the best predictive performance.

Participants in our experiment engaged in group chats prior to decision-making. This presents an opportunity to explore whether the collective chat or an individual's contributions are more predictive of decisions. Note that in scenarios where three participants chat in a group, treating the entire group chat as a single input could result in threefold replication for predicting each member's independent decision. Moreover, many researchers manually assign labels or categories to the text data, which can also be leveraged for predictions. However, some chat snippets may not have a label assigned because they are just filler sentences, and excluding them from the input data could reduce noise and increase accuracy. The data that we use for training the classifier (Fochmann et al., 2019) provides 34 categories that were assigned to each chat after being read by a human. These categories describe various aspects of the chat content, such as specific numerical propositions or lying strategies (Table S1). To leverage this information, we estimate a classifier that only uses chat text to which a label was assigned as input. Consequently, we will experiment with three text data variations to determine which variation yields the best predictions: "Chat, group", reflecting the group's entire chat log; "chat, subject", representing only the messages contributed by the individual; and "chat with label", which includes only manually labeled chat excerpts. Each variation will be tested for its ability to forecast individual decisions accurately.

In behavioral experiments, decisions are often numeric and continuous but can be categorized into binary concepts. In behavioral experiments involving numerical outcomes, such as income reports from a tax evasion game, continuous data are frequently converted into binary categories to reflect compliance status. For our study, we predict tax compliance by establishing a threshold to binarize reported income as either compliant or non-compliant. Although deriving precise compliance measures is achievable in controlled experiments, real-world applications, like those used by tax authorities, often prioritize the detection of non-compliance, even at varied thresholds of income under-reporting. The selection of a suitable threshold for classification is critical and may be informed by theoretical frameworks or data-driven insights. It is imperative to choose a threshold that provides clear differentiation for the classifier, thus enhancing the reliability of predictions. In our analysis, we have binarized the reported income data at three distinct thresholds to train the classifier: maximum compliance, and the average and median levels of reported compliance.

2.2. Preprocessing

For text classification, textual data must first be converted into numerical format. Prior to this conversion, preprocessing is essential to ensure data quality and consistency. Our preprocessing, included the following standard steps: We began with converting all positive emoticon symbols into a single token "smiley" for uniformity. Following this, we tokenized the content, which involves breaking down the text into its basic elements or 'tokens.' This step also included the removal of punctuation and 'stopwords'—commonly used words that offer limited analytical value. We then applied Gensim's phrase detection algorithm (Rehurek & Sojka, 2010) to form 'bi-grams' or pairs of consecutive words only for frequently co-occurring terms, while

also considering 'uni-grams' or single-word tokens. The subsequent stage involved 'lemmatization,' a process where words were converted to their base or dictionary form for accuracy, contrasting with the simpler but less precise 'stemming' technique. To refine the dataset, we excluded words that appeared fewer than five times. Furthermore, we incorporate a spellchecking process to improve text data quality. The spellchecker used in our study operates on a rule-based system designed to pinpoint possible errors and make amendments (Naber et al., 2003).

2.3. Embeddings

To use text data for decision prediction, it must be transformed into numerical input, a process known as embedding. A common and efficient baseline approach for text classification is to represent sentences as a bag of words or n-grams (Harris, 1954), where words are assigned either absolute counts or weighted counts. Since some words, like "the" or "a", are so frequent that they appear in all documents, they do not contribute much to distinguishing one document from another. Therefore, word counts are often re-weighted using "term-frequency" (tf) or "term-frequency times inverse document-frequency" (tf-idf) to assign greater weight to rarer words.

However, representing words as individual atomic units has limitations, such as the inability to capture relationships between words. To address this issue, vector representations can be used, which preserve the meaning of words. In this context, static embeddings are more suitable, as they require less data than dynamic embeddings. There are three popular algorithms for training word embeddings, namely Word2Vec (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013), GloVe (Pennington, Socher, & Manning, 2014), and fastText (Joulin, Grave, Bojanowski, & Mikolov, 2017).

Word2Vec uses a neural network to learn word embeddings from text data, capturing similarities between words based on their contexts. In contrast, GloVe focuses on co-occurrence probabilities of words across the entire corpus. By leveraging the co-occurrence matrix (Pennington et al., 2014), GloVe can extract semantic relationships between words. Finally, fastText improves upon Word2Vec by incorporating subword information, allowing for embeddings to be trained on smaller datasets, capturing partial information about the local word order, and generalizing to unseen words (Joulin et al., 2017).

In addition to training word embeddings on the actual dataset, we also use pre-trained embeddings. Since text data obtained from laboratory experiments are often limited in size, pre-trained embeddings trained on larger external datasets can provide additional context and improve performance.

To derive sentence embeddings from word embeddings, we average all word embeddings that occurred in the text, which has been shown to be a stable baseline across various tasks (Banea, Chen, Mihalcea, Cardie, & Wiebe, 2015; Hu, Lu, Li, & Chen, 2014; Socher, Chen, Manning, & Ng, 2013). However, a simple mean might not adequately capture the distribution of word embeddings across a text. To account for the importance of different features, we apply tf-idf weighting to individual feature vectors (Kenter & De Rijke, 2015).

In addition to averaging word embeddings to form sentence embeddings, we also train paragraph vectors (PV) directly using a technique called Distributed Memory (PV-DM) (Le & Mikolov, 2014). Rather than relying on a distributed bag of words, PV-DM involves randomly sampling adjacent words from a paragraph and predicting a center word from this set. The input to the prediction includes the context words and a paragraph ID, allowing the model to learn document-level information. This approach can capture information about the order of words in a sentence and the context in which they appear, which may be important for certain tasks.

2.4. Classifiers

Logistic regression, support vector machines (Joachims, 1998), or Naïve Bayes (Zhang, 2004) are usually considered efficient baselines for text classification. In addition to these models, we test one nonparametric model, namely k-nearest neighbors (Sun & Chen, 2011), and non-linear models like Random Forests (Breiman, 1998, 2001) and XGBoost (Chen, Schonger, & Wickens, 2016). We also test ensemble techniques such as bootstrap aggregating (Breiman, 1996), and model stacking (Wolpert, 1992). Furthermore, we also test a 2-layer perceptron model.

Specifically, a support vector machine finds the decision boundary to separate different classes by maximizing the margin. A Naïve Bayes classifier utilizes probability theory and Bayes' theorem to calculate the probability of each class for a given text and then chooses the class with the highest score. K-nearest neighbors classification is based on a majority vote, where a text is assigned to the class with the most representatives within the nearest neighbors of the point representing the text in space. A random forest classifier consists of multiple individual decision trees, with each tree predicting the class of a given text. The class with the most votes becomes the model's prediction for a given text. XGBoost (Chen & Guestrin, 2016) denotes a specific implementation of gradient-boosted decision trees designed for speed and performance.

To improve accuracy, we also implement ensemble techniques such as bagging, which combines the results of multiple classifiers trained on different subsamples of the same data set, and model stacking, which combines the predictions of several base models. The two-layer perceptron is a simple feedforward neural network composed of an input layer that receives the text, one hidden layer, and an output layer that predicts the class.

2.5. Model training

In the process of developing a predictive machine learning model, it is crucial to split the dataset into distinct sets—training and testing. The training set is utilized for the model's learning phase, while the testing set is reserved for evaluating the model's predictive capabilities.

The datasets we use possess a nested structure, wherein individual participants are embedded within groups. To preserve the integrity of the evaluation, groups part of the training set are entirely excluded from the testing set. This precaution helps to prevent information leakage.

For classification tasks with imbalanced outcome categories, stratification is standard to ensure that splits of the data mirror the full dataset's dependent variable distribution. This approach safeguards a model's external validity and the robustness of inferences. In our study, stratification during data splitting preserved the ratio of compliant to non-compliant decisions in both the training and test sets, mirroring the overall dataset's distribution.

Moreover, when addressing class imbalance within the dataset, particularly where a significant skew towards one category exists, it is a widely accepted method to implement oversampling techniques (Chawla, Bowyer, Hall, & Philip Kegelmeyer, 2002). For our study, random oversampling was employed to augment the minority class, which is the set of compliant decisions.

2.6. Evaluation

To assess the performance of our models, we calculate the standard performance metrics of accuracy, precision, recall, F1 score, and AUC. These metrics capture the model's capacity to distinguish between compliant and non-compliant reports—the central objective of our research. Our datasets exhibit a significant class imbalance, with non-compliance as the prevalent category.

Accuracy,

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Observations}}, \tag{1}$$

quantifies the proportion of correct predictions made by the model across all observations. This metric, while straightforward, can yield

a skewed perception of model performance in imbalanced scenarios. A model that predominantly predicts the majority class (non-compliant) may attain high accuracy, overshadowing its effectiveness at identifying the minority class (compliant).

Precision quantifies the exactness of the model in predicting compliance,

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}, \tag{2}$$

where a "True Positive" is a correctly identified compliant case, and a "False Positive" is a non-compliant case incorrectly labeled as compliant. This metric measures the model's effectiveness in avoiding the misclassification of non-compliance.

Recall is the metric that determines the model's capacity to identify all actual instances of compliance, calculated as

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{3}$$

. This ratio of correctly detected compliant cases to all compliant instances is key for ensuring that the model captures as many compliant behaviors as possible. High recall is imperative in studies where failing to detect compliance can entail significant costs, such as missing out on identifying taxpayers who accurately report their taxes.

The F1 Score, which harmonizes precision and recall, is given by:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{4}$$

The F1 Score is particularly valuable when it is necessary to manage a balance between different types of prediction errors in the model's performance.

The Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) provides a single summary metric of a model's ability to distinguish between binary outcomes, such as compliant and non-compliant reports in tax experiments. Unlike metrics that may depend on a specific classification threshold or data distribution, AUC remains constant across different data scales and threshold settings. It effectively captures the probability that the model ranks a randomly chosen compliant (positive) instance higher than a randomly chosen non-compliant (non-positive) instance. In our research, we adjust our model to test various thresholds for defining compliance to identify the most effective indicator for tax reporting behaviors.

2.7. Step-wise approach

To determine the optimal model configuration from numerous possible variations without incurring excessive computational costs, we employed a systematic step-wise selection method. Our procedure unfolds in three phases:

We initially explore various text data representations and binarization thresholds to identify the most effective setup. This step determines the best manner to process and categorize the chat data into binary outcomes reflective of tax compliance. Following the initial selection, we proceed to evaluate different embedding techniques to ascertain which yields the most predictive features for compliance detection. With the refined data representation and embedding, we then evaluate a range of classifiers to find the one that offers the best performance in predicting compliance. The culmination of these steps results in an optimized model based on the training data from the first tax compliance experiment. We then apply the most effective classifier to a second dataset from a subsequent experiment to validate its predictive power on new, unseen data. For more details on the implementation of the algorithm, please refer to Table S2 and Table S3.

3. Computational experiments

This section has three parts: First, we evaluate various configurations to train a classifier on data from a tax evasion experiment. Next, we use this classifier to make predictions in a separate experiment. Finally, we qualitatively assess the potential of chat text from a behavioral experiment to predict compliance.

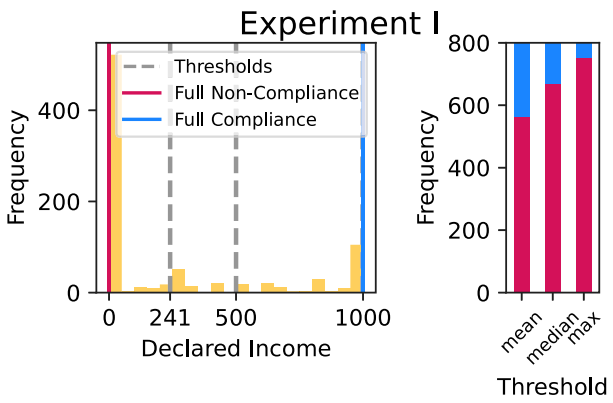

Fig. 1. Declared Income in Tax Compliance Experiment. (Left) Participants' reported income distribution in a tax game, where 0 indicates full non-compliance and 1000 marks full compliance. Three thresholds (mean: 241, median: 500, max: 1000) illustrate the thresholds for binarizing the reporting income. (Right) The frequency of reported incomes categorized by these thresholds demonstrates the discretization of continuous income data for predictive modeling.

3.1. Identification of the best configuration for training the classifier

We utilize a publicly available dataset, collected by Fochmann and Wolf (2019), to train and test our configurations. This dataset captures participant interactions in a tax evasion game, where participants are presented with a vignette detailing their taxable income (1000 ECU) and are then asked to discuss whether or not to report their income in a group chat truthfully. Following the discussion, participants individually reported their income. The dataset includes information from 141 participants who were grouped into 47 groups, and 855 decisions were recorded (Fochmann et al., 2019).

Our study aims to predict tax compliance from text communication by categorizing reported incomes into compliant and non-compliant classes, using full compliance (1000), mean (241), and median (500) reported income thresholds (Fig. 1). We assess three data variations: complete group chats, individual messages, and manually labeled chat segments. Text data are processed using tokenization, emoticon normalization, punctuation removal, and lemmatization, further refined by excluding infrequent tokens, resulting in a corpus of 2547 unique words. Additionally, we evaluate the impact of spellchecking on preprocessing. The dataset division yields 648 training samples, expanded to 858 post-oversampling, and 207 testing samples.

Best Combinations of X and y: First, we test several X- and Y-combinations. For this step, we chose a baseline setup where tf-idf vectorized text is input to a support vector machine (SVM). The regularization parameter C, the kernel k, and the degree of the polynomial kernel function d were subject to a five-fold cross-validated grid search, where we choose $C \in [-1, 4]$, $k \in$ linear, polynomial, radial basis function, and $d \in [2, 3]$. Furthermore, based on the best parameters proposed by the grid search, the model is refitted ten times to reduce prediction variance. Within each fit, the classification threshold is chosen such that it maximizes the F1 score (Lipton, Elkan, & Naryanaswamy, 2014). The reported model metrics are averaged over the ten fits.

Table 1 presents the optimal performance combinations across different configurations of *Data*, *y*, and *X Variation*. All variations have higher recall than precision, indicating a tendency towards more falsely labeled cases as compliant. Generally, a high recall is more readily attainable for models with limited predictive capability, like ours, when the minority class is a focal point for precision improvement. Furthermore, the AUC scores are moderate to low, ranging from 35.5 to 64.5, with 50 indicating a random guess. This suggests that the model's ability to differentiate between compliant and non-compliant instances is not very strong. Interestingly, results are not disparate

Table 1
Best combinations by input variation.

Variable	F1 score	precision	recall	AUC	accuracy
Data					
original	24.3	21.4	57.1	64.5	76.1
spell checked	24.3	21.4	57.1	64.5	76.1
Y Variation					
<max	24.3	21.4	57.1	64.5	76.1
<mean	36.8	25.2	74.1	57.1	50.4
<median	31.2	21.8	81.8	55.6	48.8
X Variation					
chat, group	33.6	32.4	44.4	56.6	69.3
chat, subject	36.8	25.2	74.1	57.1	50.4
chat with label	32.4	20.4	100.0	35.5	21.1

Note: Numbers in %. The classifier utilizes tf-idf vectorization as input to an SVM optimized via a five-fold cross-validation grid search. The model variance was reduced by tenfold refitting, optimizing for the F1 threshold. Metrics are the means of these refits. F1 score, precision, and recall for the "compliant" class (= 1) are reported. <max indicates that a reported income of 0–999 was labeled non-compliant, and only a report of 1000 was labeled compliant. *chat, group* indicates that the whole group chat was used to predict an individual's decision. *chat, group* uses group chat for individual prediction. *chat, subject* uses a subject's chat. *chat with label* uses manually labeled chat excerpts for prediction.

between the original text and its spellchecked counterpart. This non-existent difference in performance may be attributed to the fact that the spellchecker only amended 21.51% of tokens in the dataset. This relatively low proportion of corrections suggests that the text modifications were insufficient to impact the classification outcomes significantly. *Y Variations* refer to different thresholds used to define the binary classification of the dependent variable. This reclassification alters the data distribution, making it crucial to evaluate model performance in terms of the AUC. When applying a compliance threshold of 1000, the model records its highest AUC at 64.5%, which declines to 57.1% for the mean and 55.6% for the median threshold. However, Fig. 1 indicates a scarcity of fully compliant reports, suggesting an extreme imbalance in the dataset that could hinder the performance of subsequent classifiers. Binarizing at the mean or median achieves a more balanced label distribution. Therefore, setting the threshold at the mean is preferred for training subsequent models. *X Variations* denote different versions of chat text data used for modeling. As these do not change the underlying distribution of labels, we evaluate these variations by comparing F1 scores additionally to AUC. The highest F1 score of 36.8% is achieved using texts written solely by the subject to predict their decisions. This score decreases to 33.8% when considering entire group chat conversations and falls further to 32.4% when using only those chat excerpts with manually annotated labels. The initial decline in score can be attributed to the ambiguity introduced when group chat texts, despite identical, correspond to varied individual decisions, diluting clear patterns for the classifier to learn from. Further reduction in score with labeled chat snippets results from the decreased volume of text data available, which likely limits the classifier's learning and, consequently, its performance. Overall, Table 1 implies binarizing Y based on the mean of the reported income and deploying chat texts grouped by individuals as input X.

Features. We constructed text features by training embeddings directly on our corpus and leveraging pre-trained embeddings. The latter were sourced from deepset.ai,[2] based on the German Wikipedia dataset of 2015. To assess the efficacy of these text representations, we employed a linear logistic regression model, with the dependent variable binarized at the mean and the chat messages grouped by subjects.

Before calculating performance metrics, we visually analyze feature representations using PCA, reducing embedding vectors to two dimensions (Fig. 2). This visualization shows distinctions between corpus-specific and general corpora language processing in word embeddings. Generally, accurate embedding distances are critical for capturing the experimental nuances relevant to subsequent classification or clustering tasks. Experiment-specific embeddings, notably Word2Vec, show tight clustering due to a focused vocabulary, resulting in denser representations. In contrast, the wider scope of pre-trained models' training results in more dispersed vectors, indicating a capture of generalized language patterns. Despite distinct training approaches, pretrained Word2Vec and GloVe demonstrate comparable dispersion in vector representation. The figure reveals 'Einkommen' (income), 'Risiko' (risk), and 'Strafe' (fine) – three words central to the experiment – in varying proximities across embeddings. In GloVe, their clustering suggests that global textual patterns reflect their interrelation. At the same time, their dispersion in Word2Vec and experiment-specific embeddings, particularly Word2Vec, might indicate infrequent co-occurrence or more diverse contexts within our experimental data.

Table 2 organizes the feature representations in descending order of their F1 Scores. Similarly to results presented in Table 1, overall results depict a trend of high recall yet low precision across the models. Additionally, it is observed that accuracy is frequently below the 50% mark. This occurrence is attributed to adopting a prediction threshold configured to optimize the F1 score, subsequently influencing the accuracy detrimentally. Remarkably, a tf-idf weighted Bag of Words depicts the best F1 score of 56.6%. Leaving out the weighting slightly diminishes the F1 score to 55.1%. Three variants of Word2Vec exhibit similarly high F1 scores, spanning from 51.3% to 52.2%. Conversely, other models, such as Doc2Vec, GloVe, and FastText, exhibit lower F1 scores ranging from 33.4% to 35.0%. The findings suggest that, in the context of the experiment, keyword detection capabilities of simpler models may be more critical for predictive accuracy than the nuanced semantic comprehension offered by complex word embedding techniques. We consequently choose to utilize the tf-idf weighted Bag of Words model for subsequent model training.

Classifiers. All classification models were tuned using a 5-fold cross-validated grid search to optimize their hyperparameters (Table S2). The parameters to be tuned were specific to each model. Based on the best parameters identified by the grid search, each model was refitted ten times to reduce prediction variance. The classification threshold within each fit was selected to maximize the F1 score (Lipton et al., 2014). The reported model metrics were averaged over the ten fits.

Like the embedding outcomes detailed in Table 2, the analysis of various classifiers in Table 3 also reveals a tendency towards higher recall than precision, with accuracy frequently falling below 50%. The F1 scores for different models are closely clustered, spanning a small range from 51.1% to 56.3%. This suggests that the choice of embeddings may have a more significant impact on performance than selecting a classifier.

Table 3 shows that the Stacking classifier outperforms other models based on its F1 score (56.3%). It achieves substantial recall (76.4%), important for identifying most non-compliant cases, even though it generates more false positives as indicated by a lower precision. In the Stacking classifier, ensemble weights determine how much influence each base model has on the final output. The heavy weighting (83%) for the Neural Network (NN) in the ensemble suggests that the Stacking model relies more on the NN for its prediction than on other models in the ensemble.

The Logistic Regression with Lasso provides slightly lower performance metrics, e.g. with an F1 score of 54.5%. The remainder of the models achieve F1 scores ranging from 51.1 to 52.8%. Our primary evaluation metric is the F1 score, with a notable difference of 1.8 percentage points between the top two models. This relatively large margin leads us to choose the stacking classifier for future use.

Monetary Implications. Beyond investigating performance metrics, an alternative way to assess the predictive quality of the classifier is to evaluate its monetary implications. In the context of the tax evasion experiment that produced the data the classifier was trained

[2] https://www.deepset.ai/german-word-embeddings

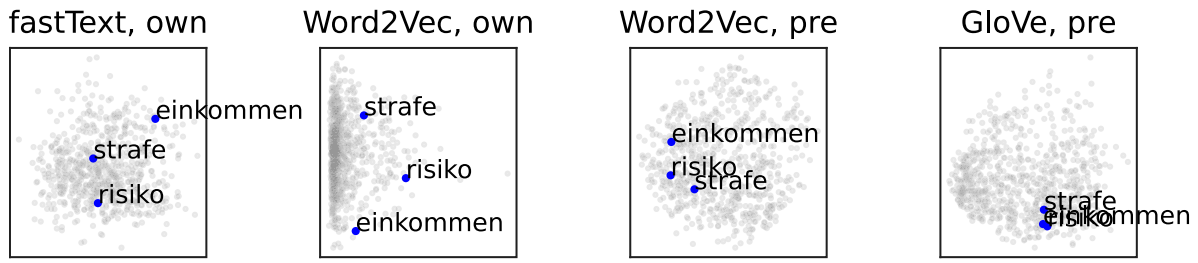

Fig. 2. Visualization of Word Embeddings with PCA Reduction. The scatter plots illustrate the 2D representation of word embeddings from different algorithms, reduced in dimensionality, using Principal Component Analysis (PCA). Three sample words – "Einkommen" (income), "Strafe" (fine), and "Risiko" (risk) – are highlighted to facilitate comparison across the visualizations. "Pre" and "Own" denote embeddings that are pretrained and custom-trained, respectively. The pretrained embeddings were sourced from the German Wikipedia corpus of 2015, while "Own" embeddings were specifically trained on chat data from a behavioral experiment. Axes ticks are omitted as the focus is on the relative, not absolute, positioning of words.

Table 2
Performance of different embeddings.

	F1 score	Precision	Recall	AUC	Accuracy
bag of words (tf-idf)	56.6	42.3	88.3	62.1	53.6
bag of words	55.1	43.1	79.7	60.9	55.6
Word2Vec (pre, tf-idf)	52.2	36.7	94.0	51.5	41.0
Word2Vec (pre, avg)	52.1	37.5	89.3	52.9	43.7
Word2Vec (own, tf-idf)	51.3	35.0	100.0	45.7	34.8
GloVe (pre, tf-idf)	35.0	22.1	93.0	51.9	32.4
GloVe (pre, avg)	34.8	21.9	94.1	52.1	30.9
Word2Vec (own, avg)	34.7	21.9	91.3	53.4	32.7
Doc2Vec	34.4	23.2	79.1	54.8	40.3
fastText (own, avg)	33.4	20.7	94.4	50.8	26.3

Note: Numbers in %. The table shows results from a logistic regression classifier with a squared Euclidean norm penalty. Metrics target the "compliant" label. "Pre" denotes pretrained embeddings from the 2015 German Wikipedia corpus. "Own" indicates embeddings trained on lab experiment chat text. "tf-idf" signifies weighting applied, while "avg" represents an unweighted average.

Table 3
Performance metrics for all classifiers tested.

	F1 score	Precision	Recall	AUC	Accuracy
Stacking	56.3	45.5	76.4	60.8	59.4
LLR	54.5	44.8	72.2	58.9	58.9
RF	52.8	55.0	61.1	63.8	67.1
NN	52.8	40.4	79.2	60.7	51.7
XGBoost	51.5	35.3	98.6	47.5	36.2
SVM	51.3	35.0	100.0	54.5	34.8
Bagging	51.3	34.8	100.0	56.7	35.7
KNN	51.1	34.8	100.0	54.2	34.3

Note: Numbers in %. The reported performance metrics are based on the minority class (label = 1) "compliant". The table tests the following models: Stacking (an ensemble model), Random Forest (RF), Neural Network (NN), Logistic Regression with Lasso (LLR), eXtreme Gradient Boosting (XGBoost), Bagging (another ensemble method), k-Nearest Neighbors (KNN), and Support Vector Machine (SVM). All models were tuned with a 5-fold cross-validated grid search to optimize their hyperparameters. The specific parameters to be tuned were dependent on each model. Based on the best parameters from the grid search, each model was refitted ten times to reduce prediction variance. The classification threshold for each fit was selected to maximize the F1 score. Reported metrics were averaged over the ten fits. Stacking classifier model weights: NN (0.75), SVM (0.56), RF (0.29), KNN (0.34), LR (0.21), XGB (0.04).

on, participants were informed that tax evasion, if detected, would need to repay the evaded tax plus an equivalent fine. For the following example, we set the chance to audit any given individual at 50%. Our test set comprises 207 cases, from which we select a subset of 103 for auditing based on this probability. In an ideal scenario with complete visibility into compliance behavior, we could cherry-pick the 103 individuals with the lowest compliance rates for auditing. Doing so would yield an audited revenue sum of 206,000 ECU (Experimental Currency Units). On the other hand, if we were to choose individuals for audit purely at random from our test set – reflecting a scenario where we audit 103 individuals without any guiding data – the expected revenue is significantly lower, at 169,800 ECU. However, the scenario improves with the introduction of our classifier. When we use the classifier's insights to select the 103 individuals it deems most likely non-compliant, the audited revenue climbs to 170,500 ECU. This represents an incremental rise of roughly 0.41% over a random selection method. While the improvement is not substantial, it is nonetheless a step in the right direction, indicating that the classifier can make audit programs more efficient.

3.2. Testing generalizability via a team coordination experiment

So far, we have identified several factors that contribute to the best performance of our classification model. Specifically, binarizing the dependent variable at the mean and grouping the chat texts by subjects are effective strategies. Regarding text representation, a tf-idf weighted bag of words is the most reliable representation. Finally, stacking multiple classifiers outperforms using a single classifier, resulting in an F1 score of 56.3%. While this score may appear low, it is important to consider the context of our dataset: we only have a small number of training samples, and the labels are highly imbalanced. Nevertheless, our results demonstrate that carefully selecting the model configuration can lead to significant gains in performance—the lowest F1 score we observed was 24.4%.

Ideally, this classifier trained generalizes to another experimental setting. Therefore, we collected chat data via a new behavioral experiment.[3] The experimental design was preregistered,[4] and the corresponding software was programmed with *oTree*[5] (Chen et al., 2016).

[3] The study was approved by the Institutional Review Board/Ethics Committee of the German Association for Experimental Economic Research e.V. Approval number No. EUFf7PP5 was obtained, and written informed consent was obtained from all participants before participating in the study.

[4] https://doi.org/10.1257/rct.5049-1.2000000000000002

[5] The oTree code is available on Github: https://github.com/carinahausladen/PredictingCompliance

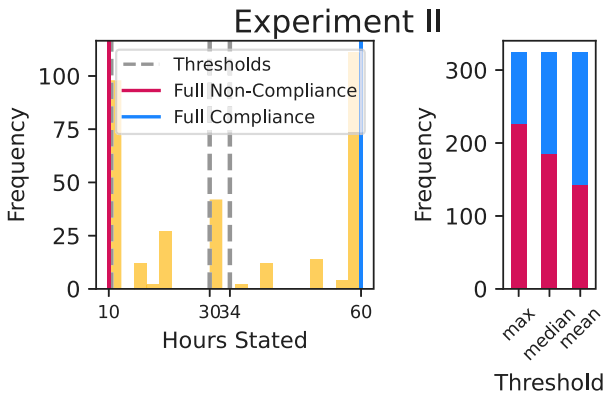

Fig. 3. Reported Surplus Hours in Team Coordination Experiment. (Left) Distribution of surplus hours reported by participants acting as employees in a scenario where coordinating on higher reported hours increased their salary. Reporting 10 h indicates full compliance, while 60 h signifies full non-compliance. Key thresholds used for analysis are marked (min: 10, median: 30, mean: 34). (Right) Categorization of reported hours using these thresholds to create predictive groups.

Participants were recruited via the MPI Decision Lab,[6] and the sessions were conducted online[7] on servers provided by the latter in May 2020.

The experimental design involved participants taking on the role of an employee for a fictitious company. Working alongside a team member, each participant was tasked with completing a project, with both team members working the same number of surplus hours. Participants were allowed to coordinate with their team members via chat about the number of surplus hours they wished to report. The more surplus hours reported, the higher the salary the fictitious company would pay. However, if the reports from the two team members diverged, they were subject to audit. Consequently, participants were motivated to use the chat to align their reporting intentions rather than discussing unrelated topics. Specifically, couples that reported the same amount of hours were randomly selected for an audit with a probability of 30%. Participants who were audited and reported more than ten surplus hours had to pay a fine. The experiment instructions can be found in the Appendix. The experimental design introduced in this study differs from that of Fochmann et al. (2019) in three important ways. Firstly, the context is no longer focused on tax evasion but on reporting surplus hours in a work-related project. Secondly, the optimal direction of lying is reversed: while participants were incentivized to underreport in the tax evasion setting, they were encouraged to overreport in the surplus hours context. Finally, the size of the reporting group was reduced from three to two participants.

Data. A total of 324 observations were collected, with participants being on average 24.8 years old and a female representation of 60%. The distribution of the stated surplus hours is displayed in Fig. 3, which shows that the majority of participants reported a compliant amount of hours, with binarization based on the full compliance benchmark (10), the mean report (34), and median report (30). On the other hand, the distribution of the stated income is bimodal, with peaks at the full compliance benchmark (10) and the full non-compliance benchmark (60). A comparison of Fig. 3 with Fig. 1 reveals that participants in the new experiment reported in a more compliant way than those in Fochmann et al. (2019). To perform the classification task, we binarized the stated income based on the mean of the reported surplus hours.

[6] https://www.coll.mpg.de/124252/decision-lab

[7] The experiment and payment processes were both conducted online. Personally identifiable information was excluded ex-post from the decision data and never used for data analysis.

Table 4
Out-of-context performance of the pretrained classifier.

	F1 score	Precision	Recall	AUC	Accuracy
> mean	71.4	55.9	100.0	55.8	55.6
> median	60.5	44.2	97.8	56.6	45.7
> max	46.8	32.4	86.7	56.5	41.0

Note: Numbers in %. Classification was based on a stacking classifier with tf-idf weighted bag of words as input trained on data from the tax evasion experiment. F1 score, precision, and recall are reported for the minority label (= 1) "compliance".

Do the two experiments exhibit structural differences in the way participants communicate? The answer appears to be yes based on simple word counts: the text data in Fochmann et al. (2019) counts 2547 unique words, while the text data in the new experiment counts 975 unique words with only 540 tokens in common.

Comparing AUC. To evaluate the model's generalizability, we tested its out-of-context performance on the dataset obtained from the new experimental setup.

Specifically, we extend our analysis to compare the classifier's performance on a second, distinct dataset with the performance on the test set from the first experiment. Given that this is an inter-dataset comparison, the AUC is a more reliable performance indicator than the F1 score due to its focus on model discrimination. For the chosen thresholds, binarization at the median yields the highest AUC (56.6%, Table 4). A slight decrease in AUC is noted when using the maximum compliance threshold (56.5%) and a further reduction when applying the mean compliance threshold (55.8%). When binarizing at the mean threshold, the model's recall hits 100%, marking full compliance detection. Fig. 3 reveals a balanced label distribution at this threshold, which typically benefits classifier efficacy. Nonetheless, a recall of 100% warrants investigation for potential overfitting within the model.

The classifier's performance in the first experiment, with an AUC of 60.8%, hints at better-than-chance predictive capabilities. When applying the classifier to the second experiment, the AUC drops to 56.6%. Consequently, while the classifier predicts compliance moderately well in contexts similar to its training environment, transferring it to an entirely different dataset poses a challenge. The observations from the data obtained by the new experimental setup are consistent with the idea that the model's poor generalization performance is due to structural differences between the two experiments. Specifically, participants in the new experiment reported a higher number of compliant instances than in the study by Fochmann et al. (2019), and the relationship between compliance and chat length was the opposite of what was observed in the earlier study. In addition, the small number of common words in both datasets suggests that the language used by the participants was significantly different. These differences likely account for the poor performance of the already weak model, which was unable to generalize to a new context.

3.3. How does language reflect lying intentions?

The previous subsection addressed the generalizability of the classifier. This subsection addresses its robustness concerning two major components: Is text an independent predictor in laboratory experiments? Moreover, does the text reflect concepts that previous literature found to predict lying?

Chat text is an unbiased independent variable. In addressing the potential for omitted variable bias, we scrutinized participant chat behavior alterations due to experimenter demand effects, which could concurrently affect chat interactions and subsequent reports. Our findings reveal a minimal impact, with only 4% of participants (13 in total) acknowledging a change in their chat behavior due to the experimenter's presence, underscoring the stability of chat text as a likely unbiased independent variable in laboratory experiments.

Analysis of 13 open-ended responses, annotated by the first author, revealed diverse behavior changes among participants. Six participants

Table 5
Linear Regression.

	Dependent variable: hours stated
Joy	1.090***
	(0.367)
Risk Attitude	0.870**
	(0.397)
Belief	0.438***
	(0.035)
Lab Experience	0.512
	(0.744)
Total Words	0.095***
	(0.031)
Lying Attitude	−0.122
	(0.466)
Political Orientation	−0.951
	(0.624)
Econ Classes	−2.262
	(1.975)
Constant	−2.036
	(6.544)
Observations	324
R^2	0.443
Adjusted R^2	0.428
Residual Std. Error	16.170 (df = 315)
F Statistic	31.254*** (df = 8; 315)

Note: $^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$. Hours stated $H \in [10, 60]$, where 10 denotes full compliance. Joy experienced $J \in [1, 10]$, where 1 denotes no joy. Beliefs $B \in [0, 100]$, where 0 means no overstatement. Risk attitude $R \in [1, 11]$, where 1 denotes risk aversion. Political orientation $P \in [1, 9]$, where 1 denotes left-wing. Econ classes $E \in [1, 2]$, where 1 denotes fewer than one class. Lying attitude $L \in [1, 10]$, where 1 denotes opposition to lying. Lab experience Lab $\in [1, 5]$, where 1 denotes no lab participation.

reduced their text length, potentially due to privacy concerns; three used more formal language, and one increased text length. Crucially, only two participants showed increased compliance, and one sought to be more agreeable. We consider only these last three responses critical, although they represent a small fraction of the sample. The behavior changes noted in the first ten responses appear to be uncorrelated with our dependent variable of compliance.

Consequently, we assert that chat text is likely a valid and unbiased independent variable in laboratory experiments, ensuring the integrity of our predictive analyses.

Various concepts influence lying. Following the surplus hours report, participants responded to a series of control questions designed to capture concepts identified in previous research as relevant to lying behavior. These questions probed feelings and emotions experienced during the experiment, attitudes towards risk, and participants' field of study, among other factors. We visualized the distributions of the answers to these questions in S1 and S2. It is unlikely that concepts with little variation in responses would strongly predict lying behavior in our experimental setting.

To assess the relationships between the control variables and the reported surplus hours, we estimated a linear regression (Table 5). We found four variables to significantly increase the number of surplus hours stated: Participants who experienced more joy during the experiment also tended to report more surplus hours (Joy = 1.090, $p < 0.001$). This result aligns with (Siniver, 2021), who found that happiness was positively correlated with dishonest behavior. The more risk-prone a participant stated to be, the higher the reported surplus hours (Risk Attitude = 0.870, $p < 0.005$), which is in line with previous literature (Dulleck et al., 2016; Fochmann et al., 2019; Fochmann & Wolf, 2019) suggesting that a higher willingness to take risks is associated with lower compliance. Participants who believed that more people in their group were non-compliant tended to report higher surplus hours (Belief = 0.438, $p < 0.001$). This finding is consistent with previous research (Fochmann, Kölle, Mohr, & Rockenbach, 2020) which suggests that non-compliant individuals expect more non-compliance from others than compliant individuals do. Furthermore, we found that the more words participants wrote in the chat the more surplus hours they stated (Total Words = 0.095, $p < 0.001$). This observation might be partly explained by anchoring and default effects. The experiment instructions provided the true amount of surplus hours as a salient piece of information, which could serve as an anchor for participants' responses (Tversky & Kahneman, 1974). Sticking with the truth (the anchor) would require less communication, as no alternative number needs to be agreed upon.

Furthermore, the coefficients for *Lab Experience*, *Lying Attitude*, *Political Orientation*, and *Econ Classes* were not statistically significant at the conventional levels. The model accounts for 44.3% of the variance in the dependent variable and is highly significant (F-statistic=31.3, $p < 0.01$).

4. Discussion and conclusion

This paper aimed to identify the best configuration for text classification of behavioral experimental data and to investigate the generalizability of the classifier. The results showed that the individual's text messages were the most predictive of their decisions, and the mean report as binarization threshold provided the most balanced precision–recall tradeoff in mapping the decisions to a broader concept. A tf-idf weighted bag of words was the most effective feature representation. A stacking classifier outperformed all individual models tested, yielding an F1 score of 56.3% and an AUC of 60.8%, which, although low, was expected due to the small and heavily skewed dataset. The generalizability of the classifier was tested by assessing whether it could be applied to another experimental behavioral setting. The results indicated that the classifier did not generalize well to a new dataset, as the AUC dropped to 56.6%.

Overall, while the predictive quality of the classifier was low, the study provided important insights into the best configuration for text classification of behavioral experimental data. By varying the configuration, the study demonstrated that a considerable gain in performance can be achieved. The recommendations in this study can be applied to a considerably larger dataset, which is necessary to build a more accurate predictive model. In that way, this study provides a valuable toolbox for the community.

These findings have practical implications for future research in the field. Researchers with larger and more diverse datasets can readily adopt the recommended configuration proposed in this study without requiring expertise in natural language processing or programming. By providing specific recommendations and easy-to-use tools, we hope to facilitate the development of predictive classifiers in the behavioral community.

Additionally, our behavioral experiment sheds light on several interesting concepts related to participants' lying behavior. Our findings show that beliefs about others' compliance behavior, risk attitudes, joy experienced, and the total number of words written highly influence a participant's decision to comply. These indices suggest an alternative avenue to predict compliance.

More generally, this paper introduces a strategy to enhance intervention effectiveness based on participants' (non) compliance. Considering the diverse responses to interventions (Engel, 2019) and potential counterproductive effects (Bruno, 1997; Fehr & Rockenbach, 2003; Gneezy & Rustichini, 2000), we propose real-time predictive models to discern group attitudes and intentions, utilizing group chats as a valuable data source. This enables precise, tailored interventions. For example, groups identified as potentially non-compliant can be redirected to compliance-enhancing treatments.

Our method has practical implications. We quantified the monetary implications of our classifier with the experimental context it was trained on. Our commitment to a no-deception policy in experimental economics necessitates transparency about employing a classifier in our

studies. Participants might alter their behavior to communicate more ambiguously because their communications may affect the likelihood of being audited. Nevertheless, the observed financial benefits from our experiment offer insight into how such classifiers might be used in practice. For instance, previous research attempted to identify tax evasion based on field data obtained from social media; however, depending on manually labeled data (Zhang, Nan, Huang, & Liu, 2020, 2021). Our experimental data, inherently linked between text and outcomes, offers a unique advantage. We propose utilizing this kind of data for semi-supervised learning, initially training models on labeled experimental data and applying these models to assist in labeling real-world data. This approach aims to streamline and enhance the accuracy of the labeling process. Additionally, a two-step transfer learning strategy can be employed. A model pre-trained on a broad corpus can be fine-tuned with our specific experimental data, capturing the linguistic patterns associated with tax evasion. Further fine-tuning on a larger dataset ensures adaptability and robustness, blending domain-specific insights with real-world diversity. This strategy leverages the strengths of both datasets, ensuring comprehensive learning. Nevertheless, the applicability and performance enhancement depends on the congruence between lab and real-world behaviors, underscoring the need for careful implementation and evaluation.

In conclusion, this paper significantly contributes to natural language processing in behavioral experimental data analysis. The study identified the most effective configuration for text classification and provided insights into the best practices for feature representation and model selection. While the predictive performance of the classifier was low, the study demonstrated the potential for natural language processing in extracting insights from behavioral data. The proposed approach for assessing compliant decision-making and the potential application for informing intervention strategies represent novel contributions to the field. We hope this work inspires further research in this area and assists researchers in developing more accurate and reliable predictive models.

CRediT authorship contribution statement

Carina I. Hausladen: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Martin Fochmann:** Resources, Supervision, Writing – review & editing. **Peter Mohr:** Conceptualization, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

All data and code is available via a public Github repository: https://github.com/carinahausladen/PredictingCompliance.

Acknowledgments

We gratefully acknowledge the valuable feedback and suggestions provided by the anonymous reviewers of the SocInfo2020 conference. This project was made possible by the generous support of a research grant from the Center for Social and Economic Behavior at the University of Cologne (Grant Agreement No. Rd09-2019-JSUG-Hausladen).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.socec.2024.102164.

References

Andres, Maximilian, Bruttel, Lisa, & Friedrichsen, Jana (2019). The effect of leniency rule on cartel formation and stability: Experiments with open communication.

Arad, Ayala, & Penczynski, Stefan (2018). Multi-dimensional reasoning in competitive resource allocation games : Evidence from intra-team communication.

Banea, Carmen, Chen, Di, Mihalcea, Rada, Cardie, Claire, & Wiebe, Janyce (2015). SimCompass: Using deep learning word embeddings to assess cross-level similarity.

Breiman, Leo (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140.

Breiman, Leo (1998). Arcing classifiers. *The Annals of Statistics*, *26*(3), 801–849.

Breiman, Leo (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Bruno, S. (1997). Frey and felix oberholzer-gee. The cost of price incentives: An empirical analysis of motivation crowding-out. *American Economic Review*, *87*(4), 746–755.

Burchardi, Konrad B., & Penczynski, Stefan P. (2014). Out of your mind: Eliciting individual reasoning in one shot games. *Games and Economic Behavior*, *84*(3), 39–57.

Chawla, Nitesh V., Bowyer, Kevin W., Hall, Lawrence O., & Philip Kegelmeyer, W. (2002). Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357.

Chen, Tianqi, & Guestrin, Carlos (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd Acm Sigkdd international conference on knowledge discovery and data mining* (pp. 785–794).

Chen, Daniel L., Schonger, Martin, & Wickens, Chris (2016). oTree – An open-source platform for laboratory, online, and field experiments. *Journal of Behavioral and Experimental Finance*, *9*, 88–97.

Dulleck, Uwe, Fooken, Jonas, Newton, Cameron, Ristl, Andrea, Schaffner, Markus, & Torgler, Benno (2016). Tax Compliance and Psychic Costs: Behavioral Experimental Evidence Using a Physiological Marker. *Journal of Public Economics*, *134*, 9–18.

van Elten, Jonas, & Penczynski, Stefan P. (2020). Coordination games with asymmetric payoffs: An experimental study with intra-group communication. *Journal of Economic Behaviour and Organization*, *169*(1), 158–188.

Engel, Christoph (2019). Estimating heterogeneous reactions to experimental treatments. *SSRN Electronic Journal*, 1–30.

Fehr, Ernst, & Rockenbach, Bettina (2003). Detrimental effects of sanctions on human altruism. *Nature*, *422*(6928), 137–140.

Fochmann, Martin, Kocher, Martin, Müller, Nadja, & Wolf, Nadja (2019). Dishonesty and risk-taking: Compliance decisions of individuals and groups. Available at SSRN 3436157.

Fochmann, Martin, Kölle, Tobias, Mohr, Peter, & Rockenbach, Bettina (2020). Trust them, threaten them, or lure them? Effective audit systems to promote compliance. In *Effective audit systems to promote compliance (July 7 2020)*.

Fochmann, Martin, & Wolf, Nadja (2019). Framing and salience effects in tax evasion decisions–An experiment on underreporting and overdeducting. *Journal of Economic Psychology*, *72*, 260–277.

Georgalos, Konstantinos, & Hey, John (2019). Testing for the emergence of spontaneous order. *Experimental Economics*, 1–21.

Gneezy, Uri, & Rustichini, Aldo (2000). Pay enough or don't pay at all. *Quarterly Journal of Economics*, *115*(3), 791–810.

Harris, Zellig S. (1954). Distributional structure. *WORD*, *10*(2–3), 146–162.

Hu, Baotian, Lu, Zhengdong, Li, Hang, & Chen, Qingcai (2014). Convolutional neural network architectures for matching natural language sentences. In *Advances in neural information processing systems, volume 3* (pp. 2042–2050).

Joachims, Thorsten. (1998). *Lecture notes in computer science*: *vol. 1398*, *Text categorization with support vector machines: Learning with many relevant features* (pp. 137–142). Springer.

Joulin, Armand, Grave, Edouard, Bojanowski, Piotr, & Mikolov, Tomas (2017). Bag of tricks for efficient text classification. *2*, (pp. 427–431). arXiv preprint arXiv:1607.01759.

Kenter, Tom, & De Rijke, Maarten (2015). Short text similarity with word embeddings. In *Proceedings of the 24th ACM international on conference on information and knowledge management, volume october* (pp. 1411–1420).

Kocher, Martin G., Schudy, Simeon, & Spantig, Lisa (2018). I lie? We lie! Why? Experimental evidence on a dishonesty shift in groups. *Management Science*, *64*(9), 3995–4008.

Le, Quoc, & Mikolov, Tomas (2014). Distributed representations of sentences and documents. In *31st International conference on machine learning, volume 4* (pp. 2931–2939).

Lipton, Zachary C., Elkan, Charles, & Naryanaswamy, Balakrishnan (2014). Optimal thresholding of classifiers to maximize F1 measure. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 225–239). Springer.

Mikolov, Tomas, Sutskever, Ilya, Chen, Kai, Corrado, Greg, & Dean, Jeffrey (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Mónica Capra, C. (2019). Understanding decision processes in guessing games: A protocol analysis approach. *Journal of the Economic Science Association*, *5*(1), 123–135.

Naber, Daniel, et al. (2003). A rule-based style and grammar checker.

Penczynski, Stefan P. (2019). Using machine learning for communication classification. *Experimental Economics*, *22*(4), 1002–1029.

Pennington, Jeffrey, Socher, Richard, & Manning, Christopher D. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing* (pp. 1532–1543).

Rehurek, Radim, & Sojka, Petr (2010). Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 workshop on new challenges for NLP frameworks* (pp. 45–50).

Siniver, Erez (2021). Do happy people cheat less ? A field experiment on dishonesty. *Journal of Behavioral and Experimental Economics*, *91*(2020), Article 101658.

Socher, Richard, Chen, Danqi, Manning, Christopher D., & Ng, Andrew Y. (2013). Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems* (pp. 926–934).

Sun, Shiliang, & Chen, Qiaona (2011). Hierarchical distance metric learning for large margin nearest neighbor classification. *International Journal of Pattern Recognition and Artificial Intelligence*, *25*(7), 1073–1087.

Tversky, Amos, & Kahneman, Daniel (1974). Judgment under uncertainty: Heuristics and biases. *science*, *185*(4157), 1124–1131.

Wolpert, David H. (1992). Stacked generalization. *Neural Networks*, *5*(2), 241–259.

Zhang, Harry (2004). The Optimality of Naive Bayes. In *Proceedings of the seventeenth international florida artificial intelligence research society conference, FLAIRS volume 2004* (pp. 562–567). 2.

Zhang, Lelin, Nan, Xi, Huang, Eva, & Liu, Sidong (2020). Detecting transaction-based tax evasion activities on social media platforms using multi-modal deep neural networks. arXiv preprint arXiv:2007.13525.

Zhang, Lelin, Nan, Xi, Huang, Eva, & Liu, Sidong (2021). Social e-commerce tax evasion detection using multi-modal deep neural networks. In *2021 Digital image computing: Techniques and applications* (pp. 01–06). IEEE.