Research on Interpretability of Stacking Ensemble Model

Applied to Mental Illnesses Prediction

ÁLVARO ORGAZ EXPÓSITO HEEJE LEE
alvarooe | heeje@kth.se

February 9, 2019

Abstract

Ensembling models is a widely used technique to improve the predictive performance of machine learning algorithms. Stacking, one of the possible methods to do ensembling, can be very useful for combining different types of models. However, the trade-off between predictive performance and model interpretability is one of the key issues in machine learning, and stacking tends to be a black box model with a reduced model interpretability. It is a relatively recent ensembling technique and there are many types of research going on about how to solve this issue. The research question of this project is: can the interpretability of stacking ensemble model predictions be improved in spite of submodels non-transparency? An empirical method has been used to conduct the research by exploring diverse statistical methods (punctual estimations, confidence intervals and probability distributions) to be applied to stacked models. Then, a new statistical evaluation method for stacking ensemble model interpretability is proposed and, in more detail, it includes indicators at a population level (submodels importance, feature importance and model instability to data noise) as well as at an instance level (confidence interval of predictions exploiting stacking architecture, feature importance and marginal effects). Finally, the empirical research has been applied to a stacking model (with eight different submodels and one final meta-learner) trained on medical data from *OSMI Mental Health in Tech Survey*, which is a topic that requires complex and accurate solutions but at the same time diligent method justifications and interpretability. We provide empirical results on this data that can lead to an effective contribution to the topic, and the main answer to the research question is that it is feasible to get a better understanding of the stacking models behaviour and feature importance, without worrying about submodels non-transparency, by using the proposed statistical evaluation method in this research.

Keywords: Machine Learning, Ensembling, Stacking, Model Interpretability, Feature importance

Table of contents

1. Introduction 2
2. Research question and hypotheses 3
3. Research method 3
4. Method application 4
5. Results and analysis 7
6. Discussion 10

1. Introduction

Statistics, machine learning[1] and artificial intelligence[2] are playing an important role as solutions for many tasks in this digital era. Some years ago, simple statistical inference using samples of data was enough for helping to make decisions. But nowadays the focus is more on achieving the highest predictive performance and boost the capabilities of statistical learning by creating state of the art machine learning models. In some scenarios where the task is complex, such as self-driving cars, some simple statistical techniques are not suitable because they are not powerful enough to find real patterns and correlations in the data.

For boosting the predictive performance and achieving higher levels of accuracy is very common to ensemble models. Ensembling means combining properly the predictions of various models, instead of choosing only the best one, when validating the generalization of candidate models. Why should we rely only on the decision of only one expert having more available? Then, when predicting new data, the final prediction is a combination of the predictions of all optimal submodels for the task in question. Ensembling includes a wide range of techniques such as averaging, weighted averaging, conditional averaging, bagging, boosting and stacking, but this project focuses on stacking.

Stacking, unlike other ensembling techniques, accepts very diverse types of models as submodels and its training process is very flexible. The main idea, which will be developed in more detail in the following section, is to predict the original target variable by training a final model or meta-learner using as features the predictions of other submodels for the same task. This gives you better predictive performance than simple averaging because it connects the predictions of submodels in a more complex way, and that is why the final model is also called meta-learner. As you can deduce, the possibilities of this technique are endless since the user can define many different stacking architectures and choosing diverse predictive models. Previously, there have been attempts that tried to study the optimal way of applying stacking in the diverse research area, some good references in [1] and [2].

However, the trade-off between predictive performance and model interpretability is one of the key issues in machine learning. As the increase of predictive performance requires more complexity, the stacking method tends to lose the interpretability that single submodels offer. That is why sometimes it is not possible to use stacking in many tasks that require transparency and clear justification of predictions. This topic opens a knowledge gap and is deeply discussed in [3].

To solve this interpretability issue, several researches were conducted with the aim of improving the transparency of ensembling techniques. As you can see in [4], there are ways to approach the trustfulness and explain the prediction of complex predictive models. In [5] the authors examined various methods for explaining black box models. A completely different way to approach the problem is introduced in [6] and also in [7]. However, there is no meaningful trial for enhancing the transparency of stacking models, so it is currently unknown how to actually approach it successfully.

[1] Field of computer science that uses statistical techniques to give computers the ability to learn with data.
[2] Theory and development of computer systems able to perform tasks that require human intelligence.

Feature importance is one of the important metrics or indicators for the interpretability of models since it can give powerful information to understand the predictions based on the features. Different ways to calculate feature importance have been researched and one of the first important contributions was for the model *Random Forest* in [8]. After that, in [9] the authors developed a model-independent version of the feature importance based on error rate after permuting features. But none of them is actually suitable for stacking ensembling architecture since it contains heterogeneous models in terms of the learning process.

In short, this research aims to contribute in this knowledge gap of stacking model interpretability by exploring a set of statistical methods (punctual estimations, confidence intervals and probability distributions) to be applied to a trained stacked model for providing indicators (at population and instance level) of feature importance, submodels importance, model instability to data noise, as well as marginal effects of features. By using this approach, insight is gained about the predictive behaviour of stacking ensemble models and its predictions interpretability.

2. Research question and hypotheses

The main aim of this research is motivated by the knowledge gap in the model transparency or interpretability of the stacking ensembling technique. Then the research question of this project is: can the interpretability of stacking models predictions be improved in spite of submodels non-transparency? Understanding interpretability as justifying the predicted value for a new instance based on its features values, getting an indicator of feature importance at population and instance level, knowing the importance and instability to data noise in the submodels of the stacking architecture, or knowing the marginal effects of features at instance level prediction. Providing an answer to the stated question would contribute to fill the knowledge gap and would enable to choose the stacking technique in more tasks.

At this point of the research three main hypotheses have been performed:

- Using the stacking technique will improve the predictive performance of the individual submodels when applying it empirically to the data of the project.
- The trade-off between interpretability and predictive performance in machine learning techniques is difficult to beat, but it could be possible using an appropriate set of statistical techniques to create an evaluation method for the stacking model.
- For answering the research question it is not necessary to assure transparency in the submodels of the stacking architecture.

3. Research method

Although a more analytical or theoretical method could be chosen, the research method selected with respect to the research question has been the empirical [10] since we have conducted the experiment based on observed medical data from *OSMI Mental Health in Tech Survey*. Another way to conduct the experiment would be to collect data by ourselves but researching the topic of stacking requires a large number of samples and many validated datasets can be found online. Then, providing an answer to the question is obtained empirically by trial and error of theoretical ideas using the collected data.

4. Method application

In this section, more detailed information is provided about the technical process applied in the experiment.

On the one hand, the input data used to perform this empirical research has been the medical dataset from *OSMI Mental Health in Tech Survey* [11] during the years 2014 and 2016. It consists of 2692 responses in total and contains information from asked questions (27 in 2014 and 63 in 2016) that aimed to measure attitudes to mental health in the tech workplace and examine the frequency of mental disorders among tech workers. One of the questions is "Have you ever sought treatment for a mental health issue from a mental health professional?" and this is the binary target variable to predict.

On the other hand, for developing the entire research a predictive performance metric has been chosen. According to our target variable, the metric chosen for binary classification is the area under the receiver operating characteristic curve (ROC) or AUC.

Then we proceed to the technical process of the experiment in the following order:

1) *Preprocessing of data.* As two survey datasets from 2014 and 2016 are used, we firstly merge them by sorting out 23 common questions (in Appendix II) as features and match them into a single dataset. Then the following data preprocessing has been done:
    - Grouping answers of questions to create the new levels of each feature. Since it is a survey with non-delimited answers, there are many different answers in some questions becoming too many levels for the models. The final levels are shown in Appendix III.
    - Missing values treatment imputing them by the mean in numerical features and by a new level "Missing" in categorical features.
    - Z-normalization of numerical features.
    - One-hot encoding of categorical features.

    Once the data has been preprocessed we split the 2692 total observations into three datasets (using balanced splitting for getting the same original target distribution in each split) with the following purposes:

    - 50% as the *train* set for training 1st stacking layer models.
    - 25% as the *validation* set for validating 1st stacking layer models and training 2nd stacking layer using as features its predictions with 1st stacking layer models.
    - 25% as the *test* set for validating 2nd stacking layer models.

2) *Building the stacking architecture.* For training a stacking model it is necessary to specify an architecture, and we have conducted the experiment with two stacking layers. The 1st contains 8 different models or submodels (*Logistic Regression, Random Forest, K-Nearest Neighbours, ExtraTrees, XGBoost, Light Gradient*

Boosted Machine, Decision Tree, AdaBoost) and the 2nd layer contains only one model or meta-learner. This structure is represented in Figure 1.

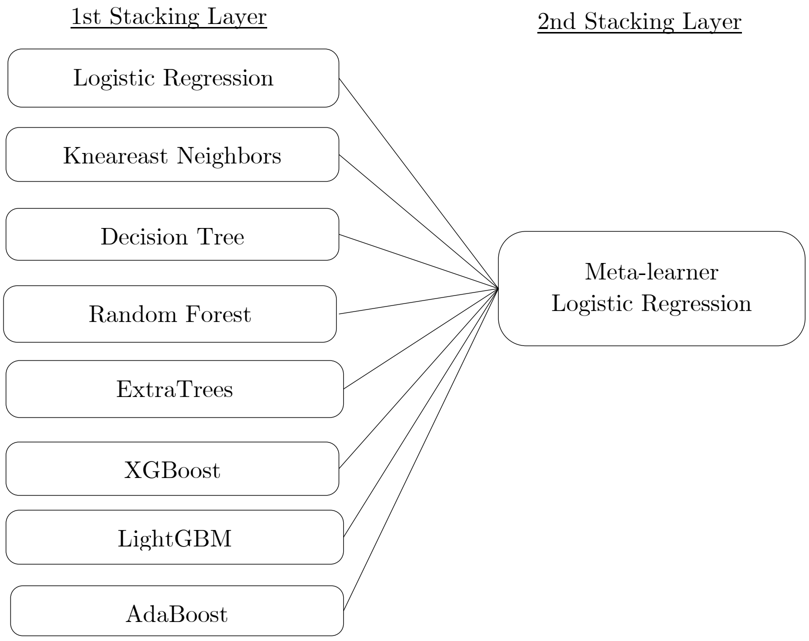


Figure 1: Stacking architecture of the experiment.

Once the architecture is defined, it is time to train all the models, layer by layer. For 1st stacking layer models, as mentioned in the data preprocessing section, we have trained the 8 submodels using the *train* set and validated them with the *validation* set. Thus, we have selected the parametrization of each candidate with higher AUC in the *validation* set as submodels. For 2nd stacking layer model, we have trained the same 8 models using as features the predictions of submodels for the *validation* set and validated them with the *test* set. Then, we have selected the model and parametrization with higher AUC in the *test* set as the meta-learner of our stacking architecture, which has been a *Logistic Regression*. Technically, for finding the best parametrization a grid search has been performed with many combinations of parameters by models which can be found in Appendix I together with the optimal parametrization selected for submodels and meta-learner.

3) *Proposing a statistical evaluation method for stacking interpretability.* In this research diverse statistical methods (punctual estimations, confidence intervals and probability distributions) has been studied to conduct the experiment. But we propose a new semi-supervised evaluation method that combines different statistical techniques (standard deviation, permutation-based out-of-bag indicators, and probability distributions) which is composed of several measures or indicators. We have designed it for extracting information from the stacking architecture and to be applied to an already trained stacking model. In detail, our evaluation method is divided into two parts:

   3.1) *At population level.* It means that we use the whole dataset and its corresponding predictions of the stacking model, instead of considering the information of only one instance. Composed by:

- *Submodels importance.* For evaluating the 1st stacking layer models importance we will compute the AUC of the meta-learner predictions in the *test* set (not used for training neither submodels or meta-learner) after randomly permuting the predictions of each submodel (model by model without modifying the rest of submodels predictions) several times with different random seeds. Then, we will create a model importance indicator using the standard deviation of all permutations AUC for each submodel. The higher standard deviation the more contribution to the stacking architecture because it means that changing the predictions of this submodel (which is one of the features for the meta-learner) affect more to the final prediction.

- *Feature importance.* For evaluating the feature importance (at the population level) we compute the AUC of the meta-learner predictions in the *test* split after randomly permuting all features (feature by feature leaving the rest of features as original data) several times with different seeds. Then, we create a feature importance indicator using the standard deviation of all permutations AUC for each feature. The higher standard deviation the more important because it means that changing the values of this feature affects more to the final prediction. Note that it enables us to get the feature importance indicators for the meta-learner but also for the submodels.

- *Model instability to data noise.* For evaluating the instability to data noise of the submodels and meta-learner we will create an instability indicator using the sum of all features importance. Basically, we sum model by model (submodels and meta-learner) the standard deviations of all permutations AUC for all features. Then, the higher the sum of standard deviations the more instability to data noise for the model.

3.2) *At instance level.* It means that we use the information of only one instance and its corresponding prediction of the stacking model. Composed by:

- *Confidence interval of prediction.* This part of the evaluation method provides a confidence interval to the stacking model prediction exploiting the stacking architecture. The final prediction (meta-learner or 2nd stacking layer model) is the centre of the confidence interval (since it is the most accurate punctual estimation of expected probability) and the upper/lower bounds are calculated using the standard deviation of submodels or 1st stacking layer models predictions.

- *Marginal effects of features.* For studying the marginal effects of features we will predict the instance after changing, feature by feature, its feature value for all possible feature levels. This indicator studies how changing the values of the features affect the predicted value for the instance.

- *Feature importance.* For studying the feature importance (at instance level) we will use, feature by feature, the standard deviation of all possible final predicted values for the instance after changing its feature value for all possible feature levels. The higher standard deviation the more important because it means that changing the feature values affect more to the predictions. It helps to evaluate the level of effect by features to the stacking model for this particular instance.

- *Prediction distribution by population features compared to the instance.* For evaluating the prediction of a particular instance respect to the predictions distribution by population features, we have used all original survey data (used for training and validating the whole stacking model) to create for each feature a visualization of the prediction distribution by its levels. Then, knowing the predicted value of the instance and its feature value, the illustration provides information for understanding why the instance has this predicted probability based on the feature.

Moreover, note that it is not necessary to assume transparency or low complexity in submodels since all the components of this evaluation method for enhancing the interpretability of the stacking ensemble models are only based on outputs of the architecture components or submodels.

5. Results and analysis

Results of building the stacking architecture

The result of predictive performance for the trained stacking model is shown in Figure 2, where the ROC curve and AUC for the *test* set (out of the bag or not used in all training and validation process) is shown for the optimal parametrization of each candidate model in both 1st and 2nd stacking layer models.

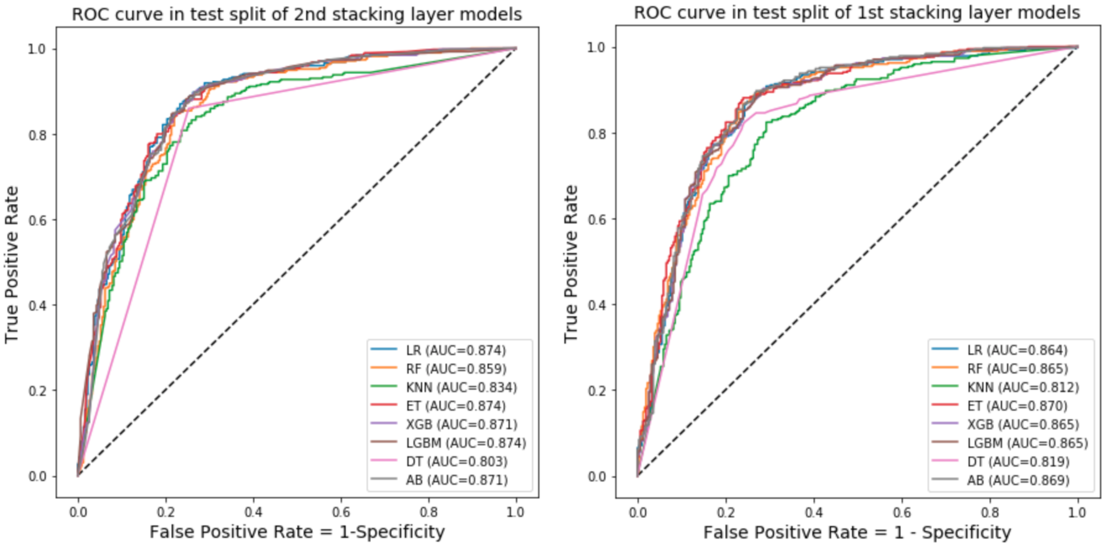


Figure 2: ROC curves of the test set for 1st (left) and 2nd (right) stacking layer models.

As mentioned before, the selected model as meta-learner or 2nd stacking layer model is the *Logistic Regression* because it achieves the higher AUC (0.874 together with *ExtraTrees* and *Light Gradient Boosted Machine*) and it is the simplest model.

Looking at the plots, the higher AUC for the *test* split in 1st stacking layer models is 0.870 with *ExtraTrees* model and the lower is 0.812 with *K-Nearest Neighbours*. However, the AUC for the *test* set of the meta-learner or 2nd stacking layer model is 0.874. It means that we have improved the predictive performance with respect to all the 1st stacking layer models.

Results of the proposed statistical evaluation method for stacking interpretability

At the population level, we can see the submodels importance indicator in Figure 3, and the feature importance and the indicator of model instability to data noise in Figure 5.

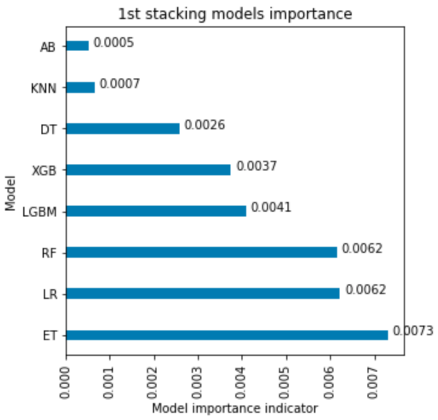


Figure 3: Submodels importance indicator.

Model	Coefficients Meta-learner Logistic Regression
AB	-0.85514709
KNN	-0.16424269
DT	0.44319322
XGB	0.74589132
LGBM	0.81482275
LR	1.14259399
RF	1.31537341
ET	1.49235045

Figure 4: Coefficients of the meta-learner Logistic Regression.

From Figure 3 we can observe that the contribution of the submodels to the final prediction or 2nd stacking layer is very different. In the empirically trained stacking architecture on the medical data of the experiment, the most important models are *ExtraTrees*, *Logistic Regression*, and *Random Forest*, and the less important *AdaBoost* and *K-Nearest Neighbours*. Moreover, a very interesting result is shown in Figure 4, where we can observe how the order of the coefficients of the meta-learner Logistic Regression matches the order of submodels importance indicator in our evaluation protocol. Then, it is a good achievement since we have checked empirically that evaluating the submodels importance with the proposed indicator achieves similar results than using a *Logistic Regression* as meta-learner. It provides transparency and trustful interpretability to the stacking architecture even when using a more complex type of model as meta-learner instead of the *Logistic Regression*.

Comparing the feature importance of the meta-learner in Figure 5 and the feature importance of each submodel in Appendix IV, we can see that for many submodels half of the features are not contributing to its prediction but for the meta-learner, which ensembles all the submodels, the list of features importance is more distributed. Moreover, another important result in Figure 5 is that the indicator value of instability to data noise for the meta-learner is ranked just in the middle of submodels, which makes sense since the meta-learner ensembles all of them becoming more powerful in terms of predictivity but also more robust than most of them.

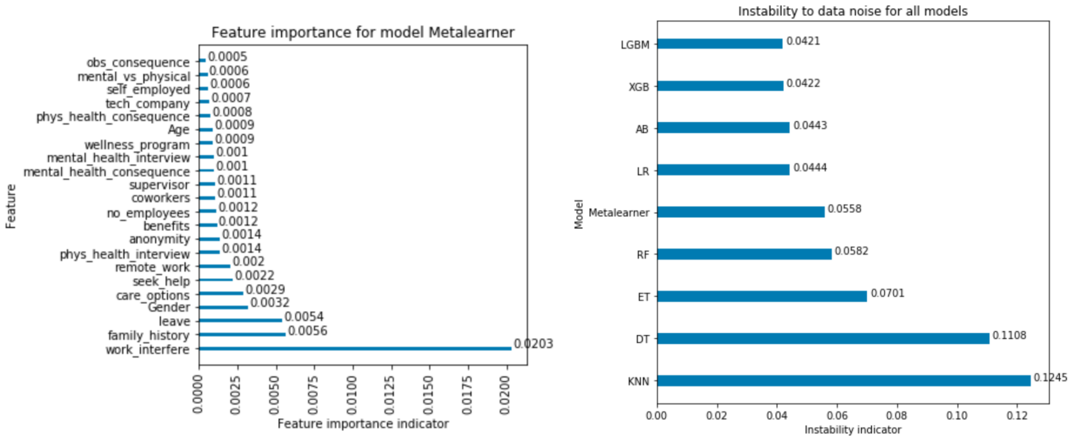


Figure 5: Feature importance at the population level (left) and submodels instability to data noise (right).

At instance level, we have chosen one instance or survey answer that received treatment (binary target variable equal to 1) out of all 2692 observations in the described dataset and its features values are specified in Appendix III.

We can see the confidence interval of its prediction and its feature importance in Figure 6. Also, its marginal effects and the predicted probability distribution by population feature *work_interfere* (most important feature at the population level as example) compared this instance in Figure 7.

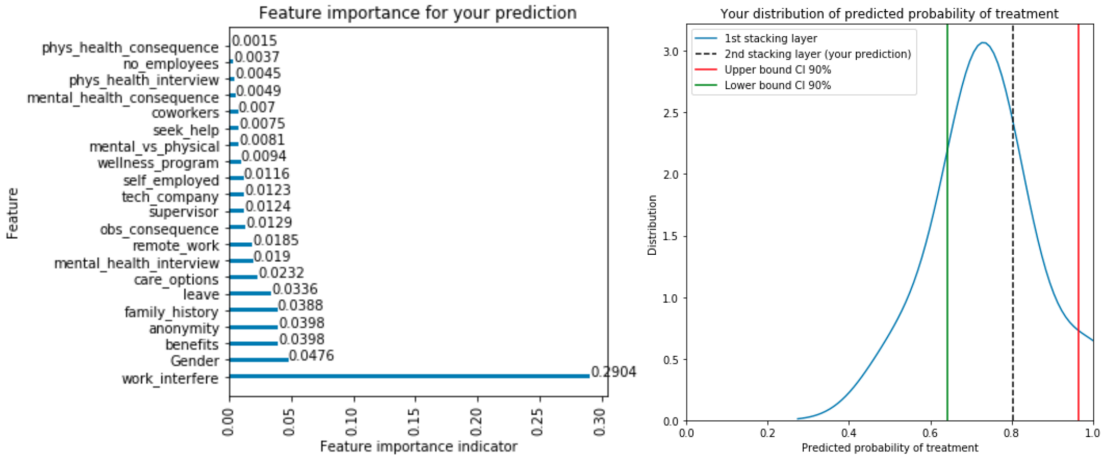


Figure 6: Confidence interval of prediction (right) and feature importance at the instance level (left).

From Figure 6 we can observe not only the final prediction of the meta-learner or 2nd stacking layer model but also the confidence interval for its prediction. Moreover, from Figure 7 we can observe that the feature importance for this selected instance is very different than feature importance at the population level, which provides specific information about the contribution of features to a particular instance prediction.

From the prediction distribution for population feature work_interfere in Figure 7, we can observe that the predicted value for the instance in question is an atypical value for the feature levels different than the instance level or answer in the corresponding question to this feature. Moreover, the marginal effects of this feature over the predicted value for the instance in question are very interesting, since its predicted value could be reduced drastically if its answer to this question was changed. Looking feature by feature at these components of the evaluation method (and combining it with the feature importance indicators) provides a detailed explanation of stacking model predictions based on its features.

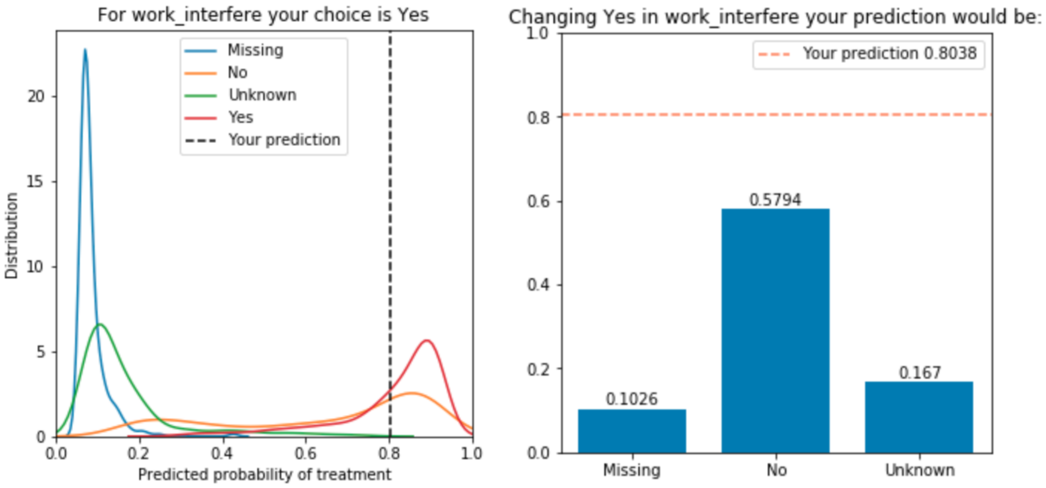


Figure 7: Prediction distribution by population feature work_ interfere (left) and its marginal effects (right) at the instance level.

Note that we have chosen only the most important feature at the population level for showing the last two components of the evaluation method, but we obtain the same for each feature in the dataset.

Results deployed in web prototype

The results of the evaluation method proposed in this research project can be explored in the web prototype[3] developed by the authors. In this web link the reader can fill out the survey and have access to his results or output of the evaluation method proposed in this research. In this prototype, the data is not collected for other reasons apart from sending the survey answer to the trained stacking model of the experiment and computing all indicators at instance level of the evaluation method.

6. Discussion

Discussion about hypothesis

Firstly, in the empirical experiment of this research, it is true that using stacking ensemble model improves the predictive performance of the individual submodels. Secondly, the hypothesis about high complexity and loss of interpretability is accepted, but the statistical evaluation method proposed in this project helps to understand the stacking architecture predictive behaviour. Thirdly, we achieve to create an evaluation method for stacking interpretability without worrying about the non-transparency of submodels, then we corroborate the third hypothesis.

[3] Web prototype at http://141.223.239.241:1234/mentalchecker

Discussion about the research question and previous work

Answering the research question of this project in a quantitative way is complex. Although we give suggestions for doing it in the future work section, we have to answer in a more qualitative way. Basically, considering interpretability issue in stacking models as defined in the research question section, our evaluation method provides a set of statistical indicators about the stacking architecture that do not depend on submodels transparency and are informative enough to answer the research question positively.

However, the research method of this experiment is empirical and the results of this project could be biased, that is why it would be necessary to apply it to different datasets or stacking model architectures, and check if the evaluation method provides informative outputs too. Furthermore, it is important to discuss that the selection of performance metric AUC could affect the results of the experiment and it would be necessary to check this evaluation method choosing another metric such as Log-Likelihood or Log Loss.

Some components of the proposed evaluation method for stacking interpretability are related to the research in [9] where the authors developed another version of the feature importance based on error rate after permuting features. But both researches are significantly different since either the statistical measure (standard deviation in this project) or the performance metric over permutations are not the same.

Discussion about future work

This section explores a list of possible future actions to continue developing this research. Firstly, according to submodels, a possible future task is to delete submodels with lower predictive performance and try other model candidates with higher capabilities. Then we would reduce the range of prediction confidence interval and make more accurate all components and measures of the evaluation method.

Secondly, according to features, a future task is feature selection for simplifying the interpretability using the proposed evaluation method. Maybe training the stacking architecture with less features does not suppose a decrease of predictive performance but it would simplify the stacking interpretability.

Moreover, according to the proposed evaluation method for stacking interpretability, it would be interesting to explore new statistical methods for proposing multidimensional indicators and measures. It means that now all evaluation method components are unidimensional (feature by feature) without considering interactions between features. Also, it would be interesting to compare the prediction CI 90% of our evaluation method for stacking model with the single submodel *Logistic Regression* and conclude if it is more accurate (lower range).

Finally, for answering the research question, it is possible to provide a more quantitative conclusion about if the interpretability of stacking models predictions can be improved with the proposed statistical evaluation method by launching a survey to a sample of workers in tech workplaces. Then, they could use the web prototype and answer some questions in the survey about the user experience for understanding their prediction.

References

[1] Zhai, Binxu, and Jianguo Chen. "Development of a stacked ensemble model for forecasting and analyzing daily average PM 2.5 concentrations in Beijing, China." Science of The Total Environment 635 (2018): 644-658.

[2] Bhasuran, Balu, et al. "Stacked ensemble combined with fuzzy matching for biomedical named entity recognition of diseases." Journal of biomedical informatics 64 (2016): 1-9.

[3] Džeroski, Saso, and Bernard Ženko. "Is combining classifiers with stacking better than selecting the best one?." Machine learning 54.3 (2004): 255-273.

[4] Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "Why should I trust you?: Explaining the predictions of any classifier." Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. ACM, 2016.

[5] Guidotti, Riccardo, et al. "A survey of methods for explaining black box models." ACM Computing Surveys (CSUR) 51.5 (2018): 93.

[6] Vellido, Alfredo, José David Martín-Guerrero, and Paulo JG Lisboa. "Making machine learning models interpretable." ESANN. Vol. 12. 2012.

[7] Rudin, Cynthia. "Algorithms for interpretable machine learning." Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2014.

[8] Breiman, Leo. "Random forests." Machine learning 45.1 (2001): 5-32.

[9] Fisher, Aaron, Cynthia Rudin, and Francesca Dominici. "Model Class Reliance: Variable Importance Measures for any Machine Learning Model Class, from the" Rashomon" Perspective." arXiv preprint arXiv:1801.01489 (2018).

[10] Bock, Peter. Getting it right: R & D methods for science and engineering. Academic Press, 2001.

[11] Experiment data available at: www.kaggle.com/osmi/mental-health-in-tech-survey

Appendix

Appendix I

List of model parameters used in the grid search to optimize the submodels and meta-learner of the stacking architecture. Also, the optimal value is shown.

Model	Python library - Function	Parameters: Optimal value
Logistic Regression	Sklearn (v0.20.2) - LogisticRegression	C: 0.1, penalty: l1, random_state: 1
Random Forest	Sklearn (v0.20.2) - RandomForestClassifier	n_estimators: 60, criterion: entropy, max_depth: 10, max_features: sqrt, random_state: 1
K-Nearest Neighbors	Sklearn (v0.20.2) - KNeighborsClassifier	weights: distance, algorithm: kd_tree, leaf_size: 10
ExtraTrees	Sklearn (v0.20.2) - ExtraTreesClassifier	n_estimators: 30, criterion: entropy, max_depth: 11, max_features: sqrt, random_state: 1
XGBoost	Xgboost (v0.81) - XGBClassifier	booster: gbtree, eta: 0.1, max_depth: 1, tree_method: auto, seed: 1
Light Gradient Boosted Machine	Lightgbm (v2.2.2) - LGBMClassifier	num_leaves: 10, min_data_in_leaf: 10, max_depth: 1, min_samples_split: 1, seed: 1
Decision Tree	Sklearn (v0.20.2) - DecisionTreeClassifier	criterion: gini, splitter: random, max_depth: 11, min_samples_split: 8, random_state: 1
AdaBoost	Sklearn (v0.20.2) - AdaBoostClassifier	n_estimators: 30, learning_rate: 0.50, random_state: 1
Logistic Regression (Meta-learner)	Sklearn (v0.20.2) - LogisticRegression	C: 1.1, penalty: l2, random_state: 1

Appendix II

List of questions of the *OSMI Mental Health in Tech Survey* used as data for the research experiment.

Feature name	**Survey question**
Age	What is your age?
anonymity	Is your anonymity protected if you choose to take advantage of mental health or substance abuse treatment resources provided by your employer?
benefits	Does your employer provide mental health benefits as part of healthcare coverage?
care_options	Do you know the options for mental health care available under your employer-provided coverage?
coworkers	Would you feel comfortable discussing a mental health disorder with your coworkers?
family_history	Do you have a family history of mental illness?
Gender	What is your gender?
leave	If a mental health issue prompted you to request a medical leave from work, asking for that would be:
mental_health_consequence	Do you think that discussing a mental health disorder with your employer would have negative consequences?
mental_health_interview	Would you bring up a mental health issue with a potential employer in an interview?
mental_vs_physical	Do you feel that your employer takes mental health as seriously as physical health?
no_employees	How many employees does your company or organization have?
obs_consequence	Have you heard of or observed negative consequences for co-workers who have been open about mental health issues in your workplace?
phys_health_consequence	Do you think that discussing a physical health issue with your employer would have negative consequences?
phys_health_interview	Would you be willing to bring up a physical health issue with a potential employer in an interview?
remote_work	Do you work remotely?
seek_help	Do you know local or online resources to seek help for a mental health disorder?
self_employed	Are you self-employed?
supervisor	Would you feel comfortable discussing a mental health disorder with your direct supervisor(s)?
tech_company	Is your employer primarily a tech company?
treatment	Have you ever sought treatment for a mental health issue from a mental health professional?
wellness_program	Has your employer ever formally discussed mental health (for example, as part of a wellness campaign or other official communication)?
work_interfere	If you have a mental health issue, do you feel that it interferes with your work when being treated effectively?

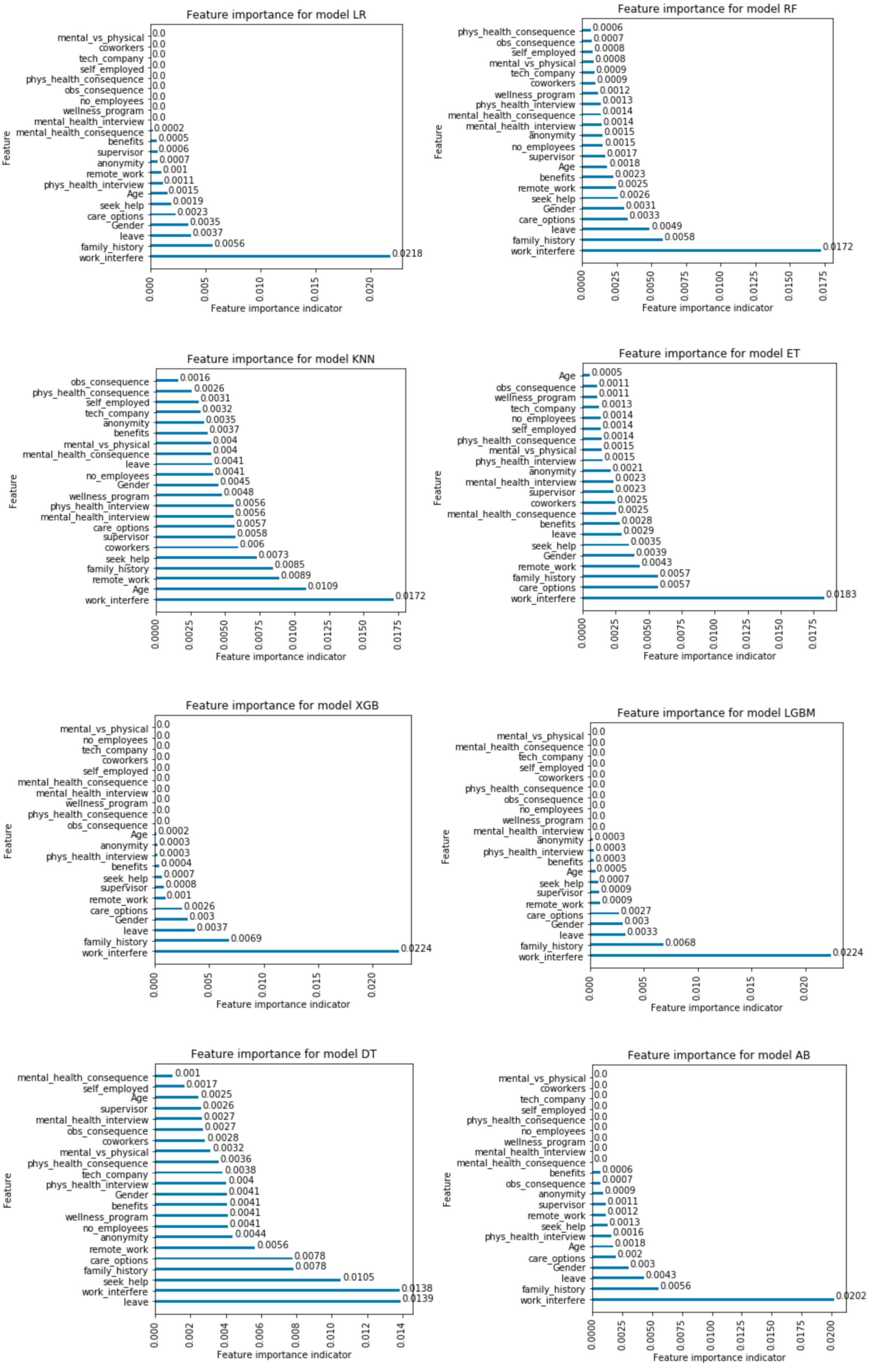
Appendix III

The final grouping of questions answers as features levels of the dataset for the experiment. The features values of the chosen instance or survey answer for the results section are specified.

Feature name	**Feature levels**
Age	15 ≤ Age ≤ 75 (37)
anonymity	Missing, No, Unknown, Yes
benefits	Missing, No, Unknown, Yes
care_options	Missing, No, Unknown, Yes
coworkers	Maybe, Missing, No, Yes
family_history	No, Unknown, Yes
Gender	Female, Homosexual, Male, Missing
leave	Difficult, Easy, Missing, Unknown
mental_health_consequence	Maybe, Missing, No, Yes
mental_health_interview	Maybe, No, Yes
mental_vs_physical	Missing, No, Unknown, Yes
no_employees	1-25, 100-1000, 26-100, Missing, More than 1000
obs_consequence	Missing, No, Yes
phys_health_consequence	Maybe, Missing, No, Yes
phys_health_interview	Maybe, No, Yes
remote_work	No, Yes
seek_help	Missing, No, Unknown, Yes
self_employed	Missing, No, Yes
supervisor	Maybe, Missing, No, Yes
tech_company	Missing, No, Yes
treatment	No, Yes
wellness_program	Missing, No, Unknown, Yes
work_interfere	Missing, No, Unknown, Yes

Appendix IV

Feature importance at the population level of each submodel in the stacking architecture.


Feature importance for model LR
mental_vs_physical 0.0
coworkers 0.0
tech_company 0.0
self_employed 0.0
phys_health_consequence 0.0
obs_consequence 0.0
no_employees 0.0
wellness_program 0.0
mental_health_interview 0.0
mental_health_consequence 0.0002
benefits 0.0005
supervisor 0.0006
anonymity 0.0007
remote_work 0.001
phys_health_interview 0.0011
Age 0.0015
seek_help 0.0019
care_options 0.0023
Gender 0.0035
leave 0.0037
family_history 0.0056
work_interfere 0.0218
Feature
Feature importance indicator
Feature importance for model RF
phys_health_consequence 0.0006
obs_consequence 0.0007
self_employed 0.0008
mental_vs_physical 0.0008
tech_company 0.0009
coworkers 0.0009
wellness_program 0.0012
phys_health_interview 0.0013
mental_health_consequence 0.0014
mental_health_interview 0.0014
anonymity 0.0015
no_employees 0.0015
supervisor 0.0017
Age 0.0018
benefits 0.0023
remote_work 0.0025
seek_help 0.0026
Gender 0.0031
care_options 0.0033
leave 0.0049
family_history 0.0058
work_interfere 0.0172
Feature
Feature importance indicator
Feature importance for model KNN
obs_consequence 0.0016
phys_health_consequence 0.0026
self_employed 0.0031
tech_company 0.0032
anonymity 0.0035
benefits 0.0037
mental_vs_physical 0.004
mental_health_consequence 0.004
leave 0.0041
no_employees 0.0041
Gender 0.0045
wellness_program 0.0048
phys_health_interview 0.0056
mental_health_interview 0.0056
care_options 0.0057
supervisor 0.0058
coworkers 0.006
seek_help 0.0073
family_history 0.0085
remote_work 0.0089
Age 0.0109
work_interfere 0.0172
Feature
Feature importance indicator
Feature importance for model ET
Age 0.0005
obs_consequence 0.0011
wellness_program 0.0011
tech_company 0.0013
no_employees 0.0014
self_employed 0.0014
phys_health_consequence 0.0014
mental_vs_physical 0.0015
phys_health_interview 0.0015
anonymity 0.0021
mental_health_interview 0.0023
supervisor 0.0023
coworkers 0.0025
mental_health_consequence 0.0025
benefits 0.0028
leave 0.0029
seek_help 0.0035
Gender 0.0039
remote_work 0.0043
family_history 0.0057
care_options 0.0057
work_interfere 0.0183
Feature
Feature importance indicator
Feature importance for model XGB
mental_vs_physical 0.0
no_employees 0.0
tech_company 0.0
coworkers 0.0
self_employed 0.0
mental_health_consequence 0.0
mental_health_interview 0.0
wellness_program 0.0
phys_health_consequence 0.0
obs_consequence 0.0
Age 0.0002
anonymity 0.0003
phys_health_interview 0.0003
benefits 0.0004
seek_help 0.0007
supervisor 0.0008
remote_work 0.001
care_options 0.0026
Gender 0.003
leave 0.0037
family_history 0.0069
work_interfere 0.0224
Feature
Feature importance indicator
Feature importance for model LGBM
mental_vs_physical 0.0
mental_health_consequence 0.0
tech_company 0.0
self_employed 0.0
coworkers 0.0
phys_health_consequence 0.0
obs_consequence 0.0
no_employees 0.0
wellness_program 0.0
mental_health_interview 0.0
anonymity 0.0003
phys_health_interview 0.0003
benefits 0.0003
Age 0.0005
seek_help 0.0007
supervisor 0.0009
remote_work 0.0009
care_options 0.0027
Gender 0.003
leave 0.0033
family_history 0.0068
work_interfere 0.0224
Feature
Feature importance indicator
Feature importance for model DT
mental_health_consequence 0.001
self_employed 0.0017
Age 0.0025
supervisor 0.0026
mental_health_interview 0.0027
obs_consequence 0.0027
coworkers 0.0028
mental_vs_physical 0.0032
phys_health_consequence 0.0036
tech_company 0.0038
phys_health_interview 0.004
Gender 0.0041
benefits 0.0041
wellness_program 0.0041
no_employees 0.0041
anonymity 0.0044
remote_work 0.0056
care_options 0.0078
family_history 0.0078
seek_help 0.0105
work_interfere 0.0138
leave 0.0139
Feature
Feature importance indicator
Feature importance for model AB
mental_vs_physical 0.0
coworkers 0.0
tech_company 0.0
self_employed 0.0
phys_health_consequence 0.0
no_employees 0.0
wellness_program 0.0
mental_health_interview 0.0
mental_health_consequence 0.0
benefits 0.0006
obs_consequence 0.0007
anonymity 0.0009
supervisor 0.0011
remote_work 0.0012
seek_help 0.0013
phys_health_interview 0.0016
Age 0.0018
care_options 0.002
Gender 0.003
leave 0.0043
family_history 0.0056
work_interfere 0.0202
Feature
Feature importance indicator