Explainable Artificial Intelligence using Counterfactual Explanations

Asher Caley(21333994), Nicholas Cannon (22241579), Lucy Conran (22571813) and Willian Herreira Lima (21840411)

CITS4404 AI and Adaptive Systems
The University of Western Australia
Faculty of Engineering and Mathematical Sciences
Perth, Australia

Increased computing power allows development of increasingly accurate machine learning models. This increase in accuracy is generally accompanied by an increase in model complexity. We often have a 'blackbox model', a model which receives inputs and makes outputs, or predictions, with little visibility possible into the mechanics of the model, or reasons a prediction was made. 'Explainable Artificial Intelligence' or 'XAI' is an active research area, seeking to develop methods to explain the reasons for model output to users. One method of explanation is to use 'Counterfactual Explanations'. Herein we examine some currently available python modules that cater for Counterfactual Explanations, and select some for further analysis and implementation.

1 Introduction

Artificial intelligence methods are being increasingly used to make predictions about future events. Many of these future events have real life implications for the subject of the prediction (i.e. the patient whose disease is being diagnosed or the credit card customer whose credit is being assessed). New and more complex models can achieve high prediction accuracy, but may not be easily understood by the user. It is important (i) in order to support good decision making processes, (ii) to comply with legal and regulatory requirements, and (iii) given the real life and important impact that a prediction can have on the subject of the prediction, that the users of a machine learning model understand why the model is making the predictions that it is. Explainable Artificial Intelligence or XAI, is an active research area which seeks to develop methods to 'explain' the predictions of the machine learning method. Further, understanding the mechanics of how a complex model makes predictions can uncover patterns and correlations in the features. For example, a study shows that a model that was predicting risk of death for pneumonia patients was showing 'asthma' as a positive indicator. The model was correctly identifying that having asthma reduces the risk of death, but this was because of the different and more intensive medical treatment that pneumonia patients with asthma receive. Not understanding the reasons behind the predictions can lead to bad decision making. Research in the explainable artificial intelligence, or XAI, area is leading to a growing number of available methods and interfaces that can be used to explain complex models. These explanation methods are actively being used in fields such as finance, science, medicine, law and criminal justice. There various different bases to the existing XAI methods available, which include (i) 'Counterfactual explanations' that describe how much that instance would need to change by to yield a different prediction (ii) TCAV - testing with concept activation - which outputs a predictions sensitivity to some high level concept (e.g. the presence of stripes in image prediction of 'zebra') (iii) various 'Feature Importance' based methods. We focus on the first method,'Counterfactual Explanations' herein.

Aim Our aim is to explore three of the available methods and identify some of the differences between them.

2 Counterfactual Explanation

A counterfactual is essentially a statement of how the world would have to change in order to achieve a different outcome. Comparable with cause-and-effect relationships, for example: "If X had not occurred, Y would not have occurred" or "If I had not eaten peanut, I would not have had an allergic reaction". Event Y is that I had an allergic reaction; the cause is X, that I ate a peanut. Interpretation of counterfactual explanation requires imagining an alternative reality that contradicts the current one, hence the name "counterfactual". Multiple counterfactuals are often possible, as there can be many ways in which a situation could change to bring about a different outcome. In his book Christoph Molnar elegantly describes it as:

"A counterfactual explanation of a prediction describes the smallest change to the feature values that changes the prediction to a predefined output." [1]

In explainable artificial intelligence, counterfactual explanations can be used as a method to explain the individual predictions of a complex machine learning model. The predicted value of an instance is calculated from the feature values by a model, and the values of those features influence the prediction. Counterfactual explanations 'interrogate' a model to show how much individual feature values would have to change by in order to flip the overall prediction. According to Rubin, the investigator should think hard about the variables besides the treatment that may causally affect Y and plan in advance. [2]. In our example, we are investigating by using counterfactual explanations the hypothesis of changing variables, such the radius of the tumour, that would change the diagnostics from benign to malignant. Although a researcher can use counterfactual methods to explain a black box model, there is an important drawback in describing the smallest changes in feature values to change the prediction, this is known as "Rashomon effect" [1] or different ways to explain the same output. For instance, when Miller [3] assess how humans select and evaluate explanations he uses Hanson's explanation [4]:

"There are as many causes of x as there are explanations of x. Consider how the cause of death might have been set out by the physician as 'multiple haemorrhage', by the barrister as 'negligence on the part of the driver', by the carriage-builder as 'a defect in the brakelock construction', by a civic planner as 'the presence of tall shrubbery at that turning'. None is more true than any of the others, but the particular context of the question makes some explanations more relevant than others."

With multiple truths, different counterfactual explanations' methods might show distinct outcomes. For example, one counterfactual method explored here might show that a reduction of a cell radius would change the diagnostics, while the same attribute might not be detected by its counterparts. As each method explains distinct truths by either reporting all counterfactual explanations or having a criterion to evaluate counterfactuals, it is helpful to create explanations that change the minimum number of features [1].

3 Counterfactual Explanation and Data Protection Regulation

After providing a glimpse of counterfactual explanations are, it is interesting to highlight that implementing the method in a AI project not only provide valuable information to change a prediction or classification but also decrease some risks. According to the Australian Privacy Act 1988, all companies operating in the country shall adopt transparent information handling practices. The same best practice is also required by The European Union General Data Protection Regulation (the GDPR). Therefore, it might be required for a company using a classification / prediction systems based on Artificial Neural Networks (CNN, RNN, FNN or other) to explain automated-decision making. [5]. For example, in the international arena recital 71 of the GDPR states that suitable safeguards against automated decision-making should be implemented [6]. While the Australian Privacy Principle's Guideline (12 and 13) says individuals should have misleading personal information corrected, adding that privacy risks can be reduced by increasing transparency about the algorithm used in an AI solution could be addressed through increased openness and accountability [7]. If a financial institution using a Neural Network to approve loans is considered to have a racial bias, the might suffer dire consequences if not finding a logical explanation in a black-box model. Therefore, by having its human readable characteristics adding Counterfactual Explanations as part of an Artificial Intelligence project adds an additional layer of protection while attending international and local guide lines.

4 Methods

4.1 Datasets used

Wisconsin breast cancer 'Breast cancer wisconsin dataset (classification)' available to load from the scikit-learn datasets [8] module in python. The dataset contains 569 instances of 30 numerical features which split into two classes, benign and malignant.

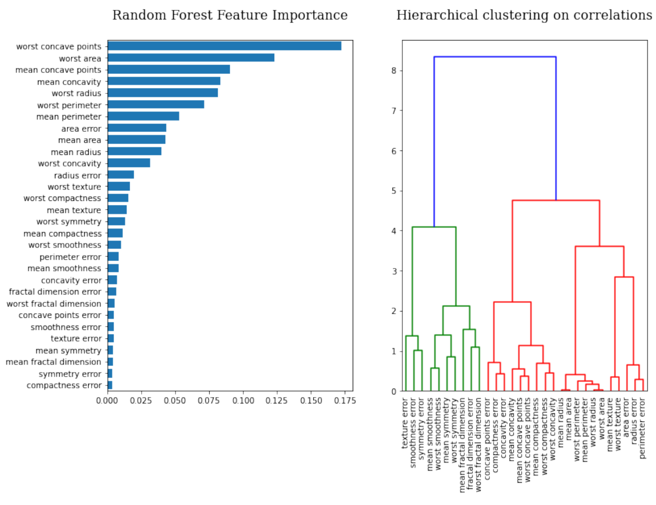


Fig. 1: The bar plot shows a tree based feature importance while the dendrogram shows grouped features into clusters

Findings from the Exploratory Data Analysis (EDA) outputs shows that after training a Random Forest Classifier using feature permutation [9], on figure 1, while the bar plot shows the feature importance in a decreasing order, whereas, the dendrogram depicts the result of a hierarchical clustering on the Spearman rank-order correlations. These results are interesting for our analysis because while "worst concave point" is shown as the most important feature, dendrogram shows that the same feature is in a different cluster when compared with "worst area" which is the second most important, thus exposing its low correlation give an idea of how complex it is to find counterfactual explanations in this dataset.

Adult income The well known 'Adult dataset' available from UCI Machine Learning repository. The dataset can be used for classification tasks has eight categorical and six continuous features, including 'age', 'occupation', 'hrs per week' and a label to indicate whether that Adult's income above or below USD50k.

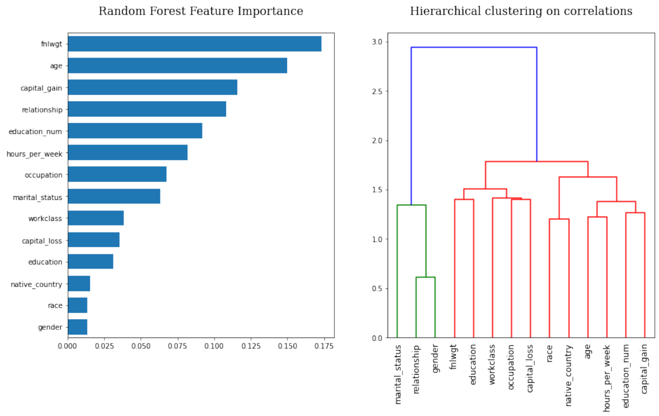


Fig. 2: The bar plot shows a tree based feature importance while the dendrogram shows grouped features into clusters

Figure 2, depicts the output of the same EDA method used on the previous dataset, note that while 'fnlwgt' and age are important predictors using a Random Forrest Classifier, they are less correlated then 'marital status' and 'gender'. These characteristics makes of this dataset very interesting example for counterfactual explanations, if one of the hypothesis is that marital status influences earnings. As broadly known in statistics, correlation does not imply causation, in the results session lets see how the method explains it.

Fashion MNIST an MNIST-like dataset as described in [10]. This dataset contains greyscale images of clothing articles, with 60,000 training examples and 10,000 testing examples. These images are labeled into 10 classes e.g. 'sneaker', 'T-shirt/top'. Each image is 28 pixels high by 28 pixels wide. The pixel values are integers ranging from 0-255 indicating the level of lightness/darkness of each pixel. It is designed to be a drop-in replacement for the overused original MNIST dataset, to be used for evaluating machine learning models.

4.2 Counterfactual Explanation Methods

We have examined a number of currently available methods of producing Counterfactual explanations:

4.2.1 Alibi

Alibi is an open source Python library aimed at machine learning model inspection and interpretation [11], it has a number of examples from well known datasets including MNIST and Boston Housing. When applied to the Wisconsin Breast Cancer dataset the library outputs a counterfactual explanation in which it finds the proportion of change of an instance, i.e.: cell shape uniformity, would collaborate with its predicted label (benign, malignant). In other words, the basic method suggests X as an instance of interest and the task is to find a complement or X' that in some way relate to the original X changing the label prediction. A better definition from Alibi Explains states that, for this method the library's logic is restricted to a particular definition, in which by asserting that X' should be as close as possible to X without violating the first condition. The authors also highlight that the main issue with this definition of "interpretability" is that the difference between X' and X required to change the model prediction might be so small as to be uninterpretable to the human eye, therefore, a sophisticated approach is recommended for these cases. [11]. The model implements its loss function as follows:

$$L(X'|X) = (f_t(X') - p_t)^2 + \lambda L_1(X', X) \qquad (1)$$

Where t is the target class for X', p_t is the target probability of the class, f_t is the model prediction on class t and L_1 is the distance between the proposed counterfactual instance X' and the instance to be explained X, for more details check Alibi GitHub repository.

Counterfactuals guided by prototypes: In addition to the method described above, Alibi can generate counterfactuals guided by prototypes, which is detailed in [12]. This method builds a prototype for each prediction class, for a given instance X, using either an autoencoder or k-d trees. The closest prototype in a class other than the instance is used to guide the counterfactual search. For example, in the Wisconsin Breast Cancer dataset, with a benign instance the closest (and only) other class is malignant. The loss function between a proposed counterfactual, and a malignant prototype will be minimised. The loss function to be optimised is as follows [11]:

$$Loss = cL_{pred} + \beta L_1 + L_2 + L_{AE} + L_{proto} \qquad (2)$$

Where the first loss term refers to the models prediction function. L_1 and L_2 define elastic net regularisation. L_{AE} aims to penalise counterfactuals outside the distribution, and L_{proto} guides the counterfactual to a prototype. The last two terms are optional. The addition of L_{proto} to the loss function aims to reduce the computational complexity of the counterfactual search process. The L_{proto} term allows the removal of the prediction loss L_{pred} yet still arrive at a plausible counterfactual. This is advantageous as when working with a black box model and only have access to the models prediction function, L_{pred} is typically required to be called many times while updating the gradients. However, when working with prototypes, only a single prediction is required at the end of each gradient update in order to check whether the counterfactual class proposed is different to the original class. It is important for machine learning tools to be able to manage both categorical and numerical data. Categorical data can be challenging for counterfactual explanations, as perturbations must be meaningful and fit with the underlying structure of the data. Badly perturbed instances might result in misleading or implausible explanations. In it's current form Alibi only has a method to handle categorical data for it's counterfactual explanation method guided by prototypes.

Please find details at the following GitHub repository: https://github.com/SeldonIO/alibi

4.2.2 DiCE

Diverse Counterfactual Explanations (DiCE) for Machine Learning Classifiers is a open source python project (originated by Microsoft Research) to develop a Python module that generates counterfactual examples. It uses an explicit loss-based method and a Variational AutoEncoder to compute the counterfactuals. As a result, it is currently limited to 'differentiable' or 'smooth' models, such as specific neural networks. Work is underway to produce a tree based model. According to the authors Mothilal et al. the method provide a general solution to an optimization problem that considers a diversity of examples and the proximity to the original input to avoiding the "Rashomon effect" [13] [1], their solution considers the tradeoff between diversity and proximity, and the tradeoff between continuous and categorical features which may differ in their relative scale and permutability.

This method uses a counterfactual generation engine where f is the Machine Learning Model, x is the instance and k is a set of generated counterfactual examples $(c_1, c_2, ..., c_k,)$ which lead to a different decision than x.

Capturing diversity via Determinantal Point Processes (DPP), [14], using metric a based on the determinant of the kernel matrix given the counterfactuals:

$$dpp\_diversity = det(K), \tag{3}$$

where $K_{i,j} = \frac{1}{1+dist(c_i,c_j}$ and $dist(c_i, c_j$ denotes a distance between two counterfactual examples. In addition, DiCE quantifies proximity as the negative vector between the original input and its counterfactual example's feature, specified by l_1-distance, calculating the proximity of a set of cf examples as the mean proximity over the set:

$$Proximity := -\frac{1}{K}\sum_{i=1}^{k} dist(c_i, X) \tag{4}$$

This paper will not covers the whole mathematical method developed by Mochilal et al. [13], but provide an introductory view to the authors' work.

Please find details at the following GitHub repository: https://github.com/interpretml/DiCE.

4.2.3 CEML

CEML is an extensible Python counterfactual toolkit that allows counterfactual explanations to be calculated on models built using popular Python machine learning packages such as Scikit-Learn, Tensorflow and Pytorch [15]. The CEML library computes counterfactuals using techniques outlined in a paper published by the author [16]. The paper explores the computation of counterfactuals in general and for specific models, in which CEML implements the general technique for computing counterfactuals.

The authors define the finding of counterfactuals using Equation (5) below which is essentially an optimization problem. Assuming $h : \mathcal{X} \mapsto \mathcal{Y}$ is a prediction function, and the counterfactual to compute is $\vec{x}'$ with input $\vec{x}$.

$$\underset{\vec{x}'}{\arg\min} \ell\left(h\left(\vec{x}'\right), y'\right) + C \cdot \theta\left(\vec{x}', \vec{x}\right) \tag{5}$$

ℓ defines the loss function that calculates the difference between the prediction $h\left(\vec{x}'\right)$ and the desired prediction y'. θ defines a regularization technique and C defines the amount of regularization strength to be applied. The common regularizations used are weighted Manhattan distance and a generalized L2 distance.

Given an arbitrary model, a counterfactual can be computed by substituting the model with h and providing a loss and regularization function. However, depending on the type of model and the selected loss and regularization, the optimization function may or may not be differentiable. There does exist separate ways of optimizing the function depending on whether or not the function is differentiable. For example, if the combination of h and the loss and regularization function are differentiable, the problem can be solved using a conjugate gradients or (L-)BFGS methods. Otherwise, gradient free methods such as the Downhill simplex method or evolutionary algorithms such as CMA-ES or CERTIFAI could be used to solve the optimization problem, if the optimization problem is not differentiable. The paper then goes on to explain model specific methods for computing counterfactuals, however these are not utilised in the CEML library.

Please find details at the following GitHub repository: https://github.com/andreArtelt/ceml

5 Results

5.1 Alibi model

We have trained a convolutional neural network ('CNN') on the Fashion MNIST classification dataset and applied the Alibi package. We have used both the non-prototype and prototype options as described above in section 4.2.1 to create counterfactual explanations. We will use the instance in Figure 3 to illustrate. The CNN predicts class 'Shirt' for our illustrative instance.

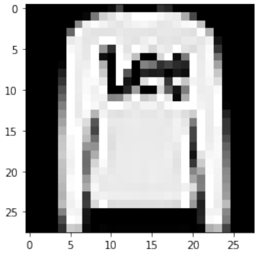


Fig. 3: Sample instance - CNN predicts class 'Shirt'

The non-prototype Alibi counterfactual explanation is show in Figure 4. It highlights the pixels that would need to change intensity in order to flip the predicted class. It is not easy to see what action needs to be taken in order to change the prediction.

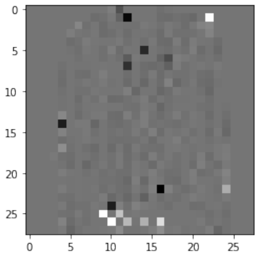


Fig. 4: Alibi counterfactual explanation without prototype

The Alibi counterfactual explanation with prototype first creates a 'prototype' for each class using an autoencoder. It then compares the original instance to each prototype (except the prototype from the predicted class) and selects the prototype and class that is most 'similar' to our instance. The counterfactual explanation outputs the pixels that need to change in intensity in order to flip the prediction to the class of the nearest prototype. Taking our illustrative instance which is predicted as 'Shirt', Alibi assigns the most similar class as 'Coat'. The counterfactual shows the pixels that need to change intensity in order to flip the prediction from 'Shirt' to 'Coat'.

The prototype selected by the model as the most similar to the 'Shirt' instance was the prototype for the 'Sandal' class. The counterfactual explanation with prototype for our example instance is shown in Figure 5 and highlights the pixels within our sample instance that would need to change in order to flip its predicted class from 'Shirt' to 'Coat'.

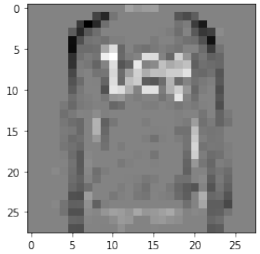
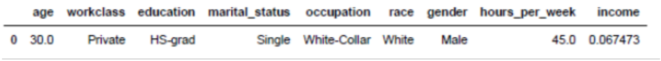


Fig. 5: Alibi counterfactual explanation with prototype

It is clear that the prototype improves the quality of the explanation.

5.2 DiCE model

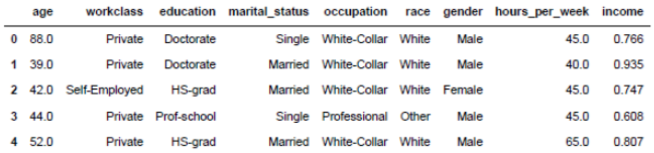
We have used the DiCE explainer to produce counterfactual explanations for the predictions of pre-trained model that has been trained on the Adult Income dataset. We 'query' a specific instance in the dataset. The model predicts income less than USD50k for our queried instance.



	age	workclass	education	marital_status	occupation	race	gender	hours_per_week	income
0	30.0	Private	HS-grad	Single	White-Collar	White	Male	45.0	0.067473

Fig. 6: Instance that has been 'queried'

It generates five counterfactual examples, each showing which feature can be adjusted, and by how much, in order to achieve a different prediction.



	age	workclass	education	marital_status	occupation	race	gender	hours_per_week	income
0	88.0	Private	Doctorate	Single	White-Collar	White	Male	45.0	0.766
1	39.0	Private	Doctorate	Married	White-Collar	White	Male	40.0	0.935
2	42.0	Self-Employed	HS-grad	Married	White-Collar	White	Female	45.0	0.747
3	44.0	Private	Prof-school	Single	Professional	Other	Male	45.0	0.608
4	52.0	Private	HS-grad	Married	White-Collar	White	Male	65.0	0.807

Fig. 7: 5 counterfactual explanations for the prediction

DiCE suggests some counterfactuals that are impossible to achieve in our example. For example, to change a person's 'Age' in order to flip a prediction. In these cases, the explanation gives insight into how the predictions on the data would change and is not proposing a new course of action. By specifying the features which we do not want to, or cannot, change we can make the output more meaningful.

5.3 CEML model

The CEML library worked very well for simple models such as Decision Trees and Random Forests with a small number of estimators on the breast cancer dataset. We attempted to use the CEML model to generate counterfactuals for a Neural Network model trained on the breast cancer dataset, but were unable to find any counterfactuals. We adjusted the default loss and optimization algorithms for the Neural Network, using a negative log likelihood loss function and a gradient descent optimization algorithm. Even adjusting the hyperparameters of the optimization algorithm, running for more iterations with a lower learning rate did not improve the counterfactual results.

We found a relationship between computation time of finding the counterfactuals, the model complexity and whether or not counterfactuals were actually found. Figure 8 shows this relationship when increasing the complexity of a Random Forest model on the breast cancer dataset. From the graph, there is almost an exponential trend between the Random Forest model complexity and the time taken to compute a counter factual result for the same instance. Increasing the complexity of the Random Forest model beyond 20 estimators, the CEML library was unable to find any counterfactuals for the same instance. This suggests a trade off between the complexity of the underlying model and the probability of finding a counterfactual. Of course, other factors exist when trying to compute counterfactuals, but from this experiment, it is clear that the underlying model complexity is an important factor.

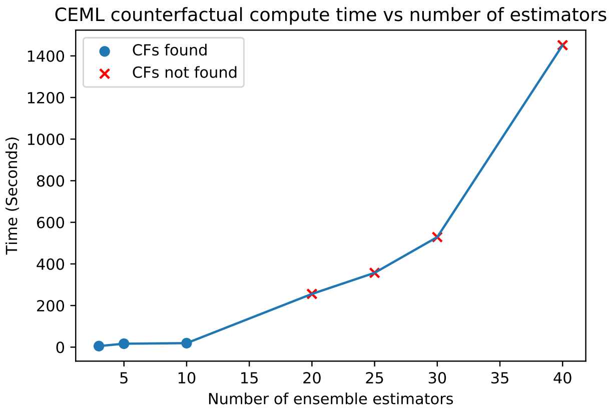


Fig. 8: CEML Random Forest counterfactual computation time and results.

6 Discussion

In this paper, we have explored and applied three of the available methods of generating counterfactuals between the approaches. Each of these approaches had benefits and tradeoffs.

Alibi showed impressive performance with more complex models, including image data. Especially the class prototyping method is effective at accelerating counterfactual search, and in this case also improved the quality and plausibility of predictions. It is also advantageous over the standard counterfactual method provided by Alibi which currently does not have a way to produce meaningful counterfactuals from categorical data. Both methods lack flexibility, such as it is not possible to only allow permutations to specific features to generate a counterfactual. This may be problematic if the feature changes made are implausible.

DiCE produces a simple output of features and by how much each would need to change in order to flip the predictions. It allows for specifying the features that are allow to be changed to generate a counterfactual which helps to ensure explanations are meaningful.

CEML provides a basic, flexible ‘barebones’ method which worked effectively to explain simple models like decision trees, however is not guaranteed to work with more complex, deep neural network models. This method requires more experimentation for the user in order to find suitable loss functions for effective functionality with a given machine learning model.

References

[1] Molnar, C., 2019. *Interpretable Machine Learning*. `https://christophm.github.io/interpretable-ml-book/`.

[2] Rubin, D. B., 1974. “Estimating causal effects of treatments in randomized and nonrandomized studies”. *Journal of Educational Psychology,* **66**, pp. 688–701.

[3] Miller, T., 2017. “Explanation in artificial intelligence: Insights from the social sciences”. *CoRR,* **abs/1706.07269**.

[4] Hanson, N. R., 1958. *Patterns of discovery : an inquiry into the conceptual foundations of science*. International series of monographs on physics. Cambridge : Cambridge University Press.

[5] , 2018. Australian entities and the eu general data protection regulation (gdpr), June.

[6] Wachter, S., Mittelstadt, B. D., and Russell, C., 2017. “Counterfactual explanations without opening the black box: Automated decisions and the GDPR”. *CoRR,* **abs/1711.00399**.

[7] , 2019. Artificial intelligence: Australia’s ethics framework – submission to the department of industry, innovation and science and data 61, June.

[8] Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G., 2013. “API design for machine learning software: experiences from

the scikit-learn project". In ECML PKDD Workshop: Languages for Data Mining and Machine Learning, pp. 108–122.

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

[10] Xiao, H., Rasul, K., and Vollgraf, R., 2017. "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms". *CoRR,* **abs/1708.07747**.

[11] Klaise, J., Van Looveren, A., Vacanti, G., and Coca, A., 2019. Alibi: Algorithms for monitoring and explaining machine learning models.

[12] Van Looveren, A., and Klaise, J., 2019. "Interpretable counterfactual explanations guided by prototypes". *arXiv preprint arXiv:1907.02584*.

[13] Mothilal, R. K., Sharma, A., and Tan, C., 2020. "Explaining machine learning classifiers through diverse counterfactual explanations". *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, Jan.

[14] Kulesza, A., 2012. "Determinantal point processes for machine learning". *Foundations and Trends® in Machine Learning,* **5**(2-3), p. 123–286.

[15] Artelt, A., 2019 - 2020. Ceml: Counterfactuals for explaining machine learning models - a python toolbox. `https://www.github.com/andreArtelt/ceml`.

[16] Artelt, A., and Hammer, B., 2019. On the computation of counterfactual explanations – a survey.