Percorso di Eccellenza

Gianluca De Carlo (mat. 1805894)

A.Y. 2021/2022

Contents

1 Introduction

Complex diseases are caused by a group of genes known as disease genes. Identifying the associations between diseases and their causal genes is a critical problem concerning human health; indeed, understanding the mechanisms of diseases, can result in their better diagnosis, treatment, and prevention.
During the recent years, many powerful techniques have been developed to find disease-gene associations using different approaches, like text mining of a specific disease's literature or by employing Machine Learning (ML) algorithms, exploiting a wide variety of models, training datasets and feature extraction mechanisms.
Many efforts are devoted to the discovery of genes involved with specific diseases, and powerful techniques are used to detect numerous candidate genes. However, the experimental validation of candidate genes is an expensive and resource-consuming task. Therefore, it is important to develop computational approaches able to focus expensive laboratory tests only on the most promising candidates.
The results of a ML method applied to the **Gene-Disease Association** (GDA) problem will be hand in to scientist working in laboratories where the associations are validated, and usually these scientist are not experts in machine learning. So that costly experiments are performed on the results of a ML algorithm, it is necessary that the scientists are able to trust the model. To achieve such goal, it is necessary to understand how the model reached a certain decision, but most deep learning models are designed without considering later interpretability of their results; this leads into treating them as black boxes.
Without understanding the relationships behind the inputs and the predictions, ML models cannot be understood and fully trusted, which prevents their applications in critical areas [1].
Explainable AI is a recent research area, where the goal is to dive deep into a non-interpretable model, to understand why it reached a determined prediction. Employing explainability have two major benefits: explain the reasons behind a model's prediction even to non-ML experts, making the model more trustworthy, debug the learning process of the model.
The use of explainability in this work is **two-folded**:

- Provide an insight on the model's predictions. By exploiting the explanation's results, it can be possible to analyze the features that are most important for a particular disease; moreover, explainability will be used to understand how different models learn on the same dataset.

- Explainability methods are then used as a tool to perform **dimensionality reduction**. **Curse of dimensionality** is a widespread problem in Data Science. By using the explanations, it is possible to uncover if there are some features that are not relevant for the purpose of classification, hence, that can be removed, reducing the size of each vector in the dataset, and consequently, bringing the dataset to a lower-dimensional space.

2 NeDBIT Features

The experiments in this work are carried out on the **NeDBIT** features, introduced in [2]. The authors of NIAPU treated GDA as a Positive-Unlabeled (PU) learning problem, where the known positive genes are labeled as such, while the remaining part of the dataset is not labeled, since in the set of non-labeled genes new positive candidates can be identified.
One investigated approach is a **two-step** technique, that aims at relabelling the unlabeled instances into five classes: Positive (P), Likely Positive (LP), Weakly Negative (WN), Likely Negative (LN) and Reliable Negative (RN).
The technique is based on the separability and smoothness assumptions, which require that the features should be able to distinguish between positive and negative instances, and that instances which have similar features are more likely to have the same label.
The study has been performed on two datasets: Human PPI from BioGRID and GDAs from DisGeNET. With a focus on five pathologies: malignant neoplasm of breast (disease ID C0006142, 1074 genes), schizophrenia (C0036341, 883 genes), liver cirrhosis (C0023893, 774 genes), colorectal carcinoma (C0009402, 702 genes) and malignant neoplasm of prostate (C0376358, 616 genes).
Positive instances are the known disease genes, Reliable Negative instances represent the genes whose features are the furthest from the average features in the P set, while the remaining labels are assigned through a Markov process with restart, introduced to obtain the stationary distribution of the propagating labels process, described by the equation:

$$G_r = (1 - \alpha)\mathcal{W}_n^t G_{r-1} + \alpha G_0 \tag{1}$$

Where α is the restart probability, $\mathcal{W}$ is the weight matrix and G_0 is the starting vector, containing the initial weights. When a stationary distribution is considered reached, the vector G_∞ is used to assign the remaining labels (LP, WN, LN) by a division in quantiles.
The label propagation process is based on the NeDBIT features, that are: degree, ring, NetRank, NetShort, HeatDiff, InfoDiff.
Where HeatDiff and InfoDiff are two network diffusion-based features, and NetRank and NetShort are two biology-informed topological metrics.
HeatDiff and **InfoDiff** are obtained by a heat diffusion process over networks, that is among the most used process for disease gene prioritisation and prediction. These processes are based on the heat diffusion equation in normal environments, where the heat tends to spread from the warmer area to the colder one, and in the same way, they want to represent how the information spreads in the network, from important (warmer) nodes, to less important (colder) nodes.
The **NetRank** measure, introduced for the first time in this work, is based on the concept of ring structure generalized to a set of seed nodes. Starting from seed nodes, a partition of the graph in sub-graphs, or rings, is introduced. In this way the seed nodes that have many other seed nodes in the ring have the highest ranks.
The **NetShort** measure is based on the idea that a generic node is topologically

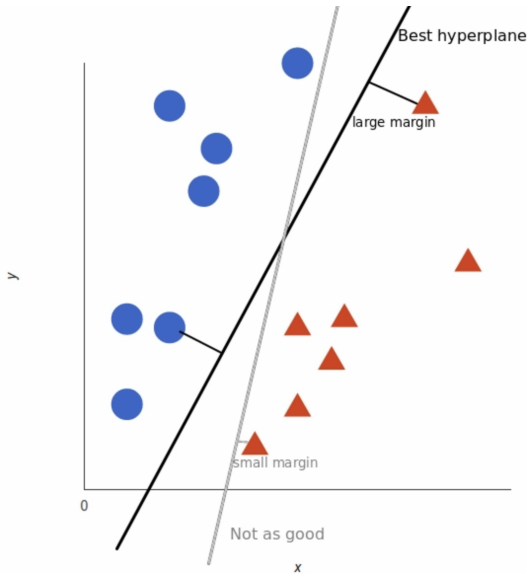
important for a disease if a large number of seed nodes must be traversed to reach it. Therefore, the measure is implemented computing the weighted average shortest paths reaching the node under consideration, where each link is weighted favouring links connecting seed nodes and penalising links connecting non-seed nodes.

3 Models

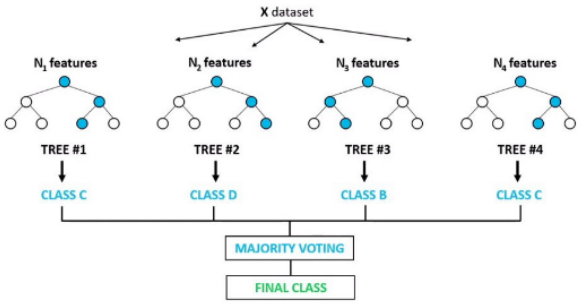
On the **NeDBIT** features have been trained three different classifiers:

- Support Vector Machine (SVM) [3]
- Random Forest Classifier (RFC) [4]
- Multi-layer Perceptron (MLP) [5]

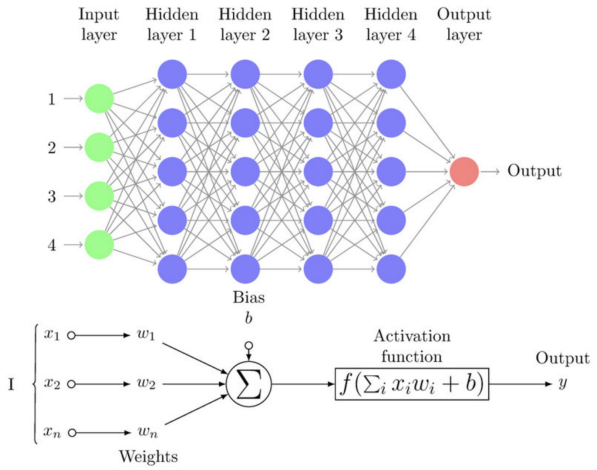
to analyze how each model performs and to select the best one.
SVM is a supervised learning model for solving classification and regression problems. The main idea behind SVM is to create a hyperplane which separates the dataset into classes.



As SVM, **Random Forest Classifier** is a supervised learning algorithm that can be used for classification and regression problems. Random Forests can be regarded as an **ensemble method**; indeed, they are comprised of decision trees, that are the building blocks. The actual prediction of a random forest is chosen via majority voting between the predictions of the decision trees.



Unlike SVM and Random Forest, **Multi-Layer Perceptron** is a Deep-Learning algorithm, meaning that it is built using different layers: the input/output layers plus the hidden layers.
The input layer receives the input signal to be processed, the output layer performs the required task, such as regression or classification, and an arbitrary number of hidden layers are placed in between the input and output layers, forming the true computational engine of the MLP.
The layers are composed by neurons, whose operation is regulated by an activation function. If the weighted sum of the inputs coming into a neuron exceed a certain threshold, then the neuron is activated. The values of the weights associated to each feature are trained by a back propagation learning algorithm.



4 Explainability Methods

On the models described in the precedent chapter, two different explainability methods have been used: Local Interpretable Model-agnostic Explanations (**Lime**) [6] and SHapley Additive exPlanation (**Shap**) [7].

4.1 LIME

The overall goal of LIME is to identify an interpretable model over the interpretable representation that is locally faithful to the classifier. The explanation produced by LIME is obtained by solving the optimization problem described by the following equation:

$$\xi(x) = argmin_{g \in G}\ \mathcal{L}(f, g, \pi_x) + \Omega(x) \tag{2}$$

Where $x \in \mathbb{R}^d$ is the original representation of an instance being explained, and $x' \in \{0,1\}^{d'}$ is its interpretable representation. G is the class of interpretable models, and the domain of g is $\{0,1\}^{d'}$, hence, g works over the presence/absence of interpretable components. $f : \mathbb{R}^d \to \mathbb{R}$ is the model to be explained and π_x is a proximity measure between an instance z to x, to define a locality around x. $\mathcal{L}(f, g, \pi_x)$ is a loss functions that measures how unfaithful g is in approximating f in the locality defined by π_x. Since not every model is interpretable in the same way, it is used $\Omega(g)$ as a metric for the complexity of the explanations of g.
Since a desired property of an explainer is to be **model-agnostic**, the loss function $\mathcal{L}$ is minimized without making any assumption on f.
In order to understand the local behavior of f, nonzero elements of x' are draw uniformly at random, generating **perturbed samples**. Given a perturbed sample $z' \in \{0,1\}^{d'}$, it is recovered its original representation $z \in \mathbb{R}^d$ and obtained $f(z)$, which is used as label for the explanation model. Given the dataset $\mathcal{Z}$ of perturbed samples, Eq. 2 is optimized to compute the explanation $\xi(x)$.

4.2 SHAP

SHAP is a game theoretic approach to explain the output of any Machine Learning model [8]. In game theory are implied a 'game' and some 'players'; when applying Shapley Values to ML, the 'game' is reproducing the outcome of the model, while the 'players' are the features learned by the model [9]. To generate the explanations, SHAP computes the average marginal contribution that each feature bring to the prediction made by the model.
The base idea behind Shapley values is that the outcome of each possible combination of features $f \in F$, where F is the set of all features, should be considered in order to understand the importance of each feature.
The number of combinations is $2^{|F|}$, and for each combination SHAP must train a distinct model. Each model is equivalent to the other, differing only on the set of features they are trained on. Since the exact computation of SHAP values is challenging, one solution is to compute the contributions for only a few samples of all the possible coalitions.

5 Results

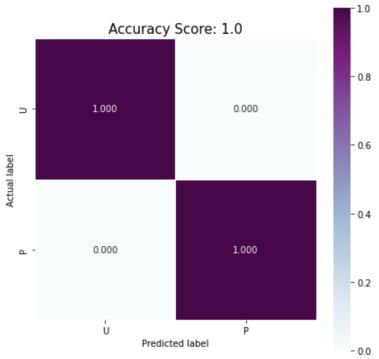
In this chapter are shown the results obtained by the three different models (SVM, RFC, MLP) on the diseases 'malignant neoplasm of breast' (disease ID C0006142) and 'schizophrenia' (C0036341), with the binary classification task, where the models are asked to classify if a gene is positive (P) or unlabelled (U); then, the two explainability methods (Lime, Shap) are applied to the trained models to analyze the differences in the learning processes on the same dataset. Finally, the explanations are used to find features that are not relevant for the classification process, hence that can be removed, and the models are tested on the smaller dataset.

5.1 Classification results

- Malignant neoplasm of breast

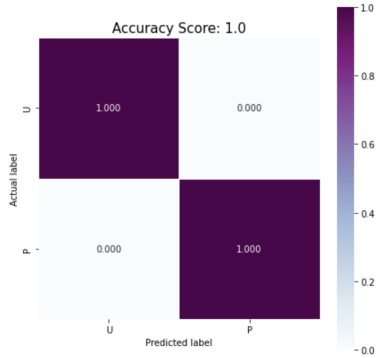
SVM

	precision	recall	f1-score	support
0	1.00	1.00	1.00	5621
1	1.00	1.00	1.00	308
accuracy			1.00	5929
macro avg	1.00	1.00	1.00	5929
weighted avg	1.00	1.00	1.00	5929



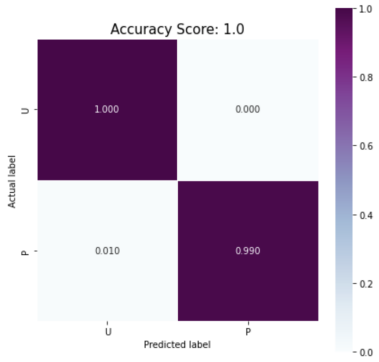
RFC

	precision	recall	f1-score	support
0	1.00	1.00	1.00	5621
1	1.00	1.00	1.00	308
accuracy			1.00	5929
macro avg	1.00	1.00	1.00	5929
weighted avg	1.00	1.00	1.00	5929



MLP

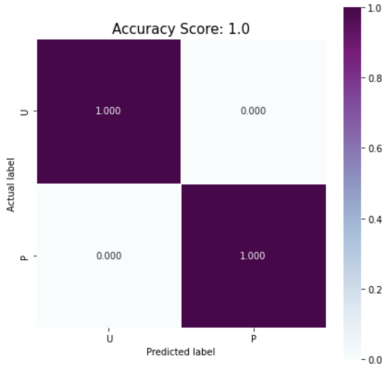
	precision	recall	f1-score	support
0	1.00	1.00	1.00	5621
1	1.00	0.99	1.00	308
accuracy			1.00	5929
macro avg	1.00	1.00	1.00	5929
weighted avg	1.00	1.00	1.00	5929



- Schizophrenia

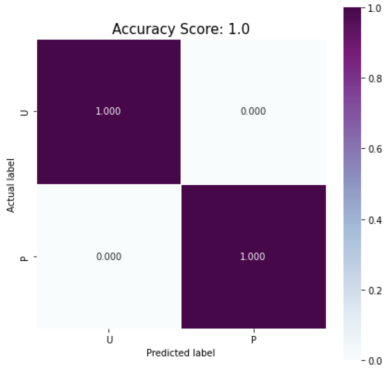
SVM

	precision	recall	f1-score	support
0	1.00	1.00	1.00	5679
1	1.00	1.00	1.00	250
accuracy			1.00	5929
macro avg	1.00	1.00	1.00	5929
weighted avg	1.00	1.00	1.00	5929



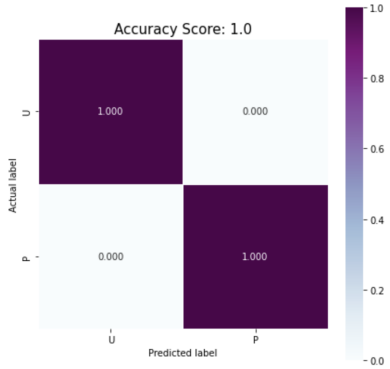
RFC

	precision	recall	f1-score	support
0	1.00	1.00	1.00	5679
1	1.00	1.00	1.00	250
accuracy			1.00	5929
macro avg	1.00	1.00	1.00	5929
weighted avg	1.00	1.00	1.00	5929



MLP

	precision	recall	f1-score	support
0	1.00	1.00	1.00	5679
1	1.00	1.00	1.00	250
accuracy			1.00	5929
macro avg	1.00	1.00	1.00	5929
weighted avg	1.00	1.00	1.00	5929



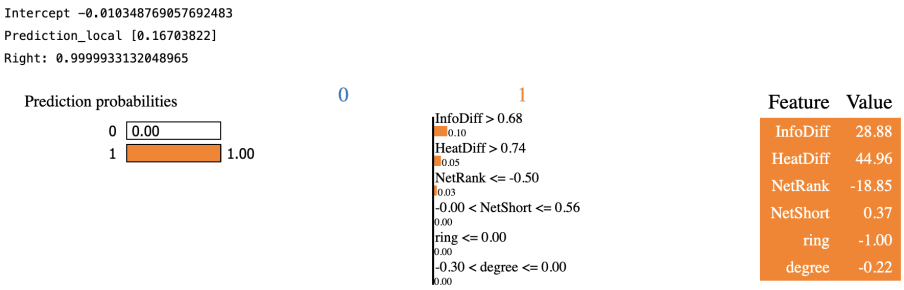
5.2 Explanations

As we can see from the results, the models learned with very high accuracies the classification task. In this case it makes sense to investigate the results of the explanations to compare the learning processes.

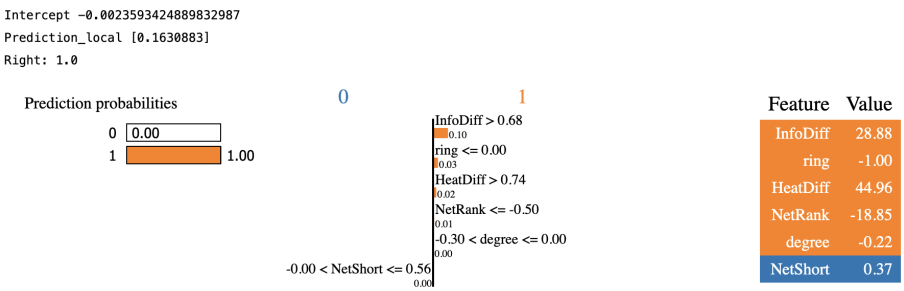
5.2.1 Lime

The local explanations conducted by Lime refer to the same sample for each disease. The following plots allow to understand how the different models learned on the same sample.

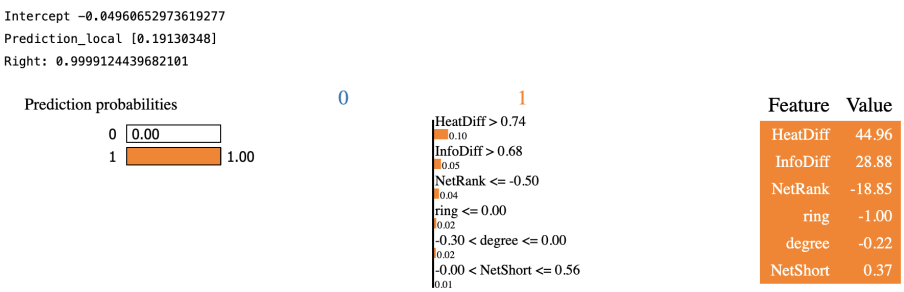
SVM



RFC



MLP

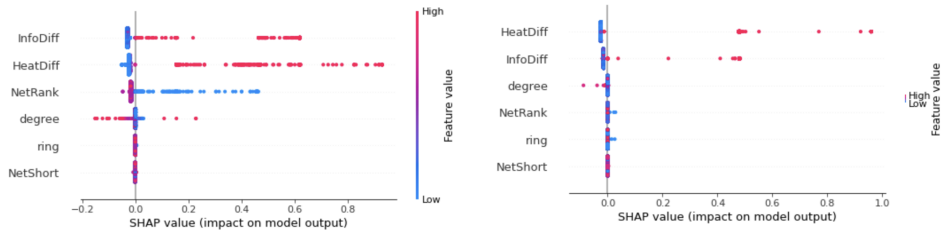


By analyzing the local explanations on the same sample, it can be noted how the models learned the features with the same importance.

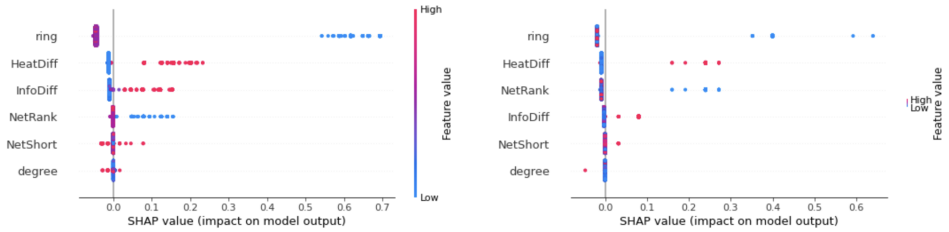
5.2.2 Shap

Shap allows to gain an insight on the global behavior of the model. By looking at the **beeswarm** plot [10], it is possible to understand the relation between a feature's value and its impact on the model prediction.
For each model type, the plot on the left refers to the Malignant neoplasm of breast while the plots on the right to the Schizophrenia. The following plots allow to understand, for each model, the features' relevance for different disease, mirroring how genes associated to different diseases may behave differently.

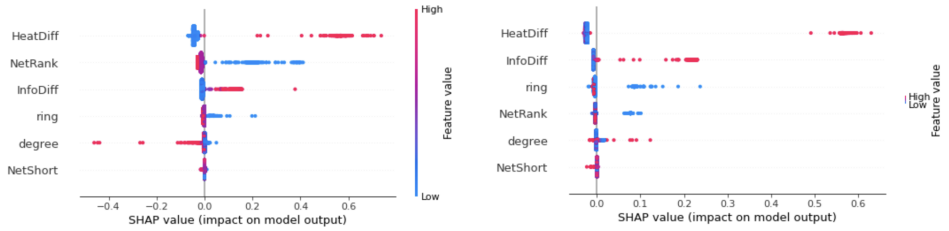
- SVM



- RFC

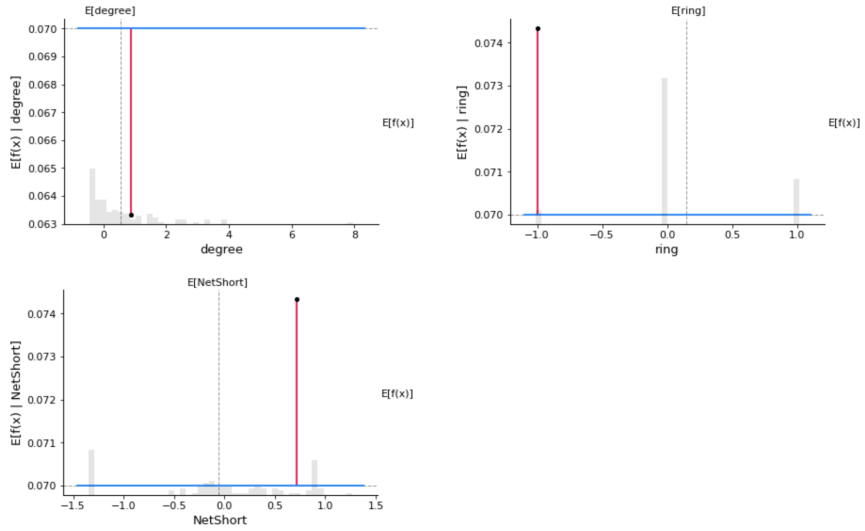


- MLP

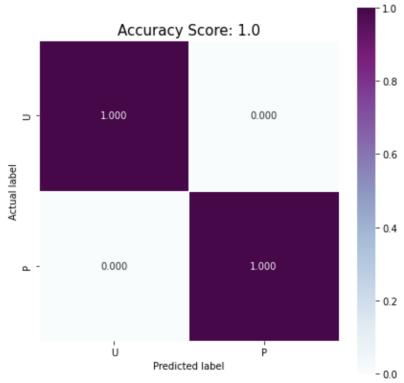


5.3 Dimensionality Reduction

In this chapter, it is investigated a possible solution of the **Curse of Dimensionality** through the use of explainability.
To understand a feature's importance in a model it is necessary to understand both how changing that feature impacts the model's output, and also the distribution of that feature's values. To visualize this, the **partial dependence plots** from Shap [11] can be used. The gray horizontal line in the plots represents the expected value of the model, the vertical gray line represents the average value of a feature, while the blue line represents the partial dependence plot (which is the average value of the model output when a feature has a fixed value).
The basic idea is that if a changing in values of a feature does not affect the expected output of the model, then the feature is not influential in the learning process, hence, can be removed from the dataset.
For example, by analyzing the partial dependence plots of the SVM model trained on the Malignant neoplasm of breast disease, can be noted three features whose trend does not affect the model output:



From the plots above, it is observable how the expected value of the prediction function does not change when the features' values vary in all the possible range. Indeed, by removing 'degree', 'ring' and 'NetShort' from the dataset the performance of the model do not get worse, neither of a small drop.



Moreover, by removing the three features from the dataset, the memory usage of the dataset is dropped by 50% (926.4 KB original dataset, 463.3 KB reduced dataset). [1]

[1] GitHub of the project source code: https://github.com/GiDeCarlo/excellencepath2022

References

[1] Hao Yuan et al. "On Explainability of Graph Neural Networks via Subgraph Explorations". In: (2021). URL: http://arxiv.org/abs/2102.05152.

[2] Paola Stolfi et al. "Adaptive Positive-Unlabelled Learning via Markov Diffusion". In: *CoRR* abs/2108.06158 (2021). arXiv: 2108.06158. URL: https://arxiv.org/abs/2108.06158.

[3] *Scikit Learn SVM documentation*. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html.

[4] *Scikit Learn RFC documentation*. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.

[5] *Scikit Learn MLP documentation*. https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html.

[6] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: *CoRR* abs/1602.04938 (2016). arXiv: 1602.04938. URL: http://arxiv.org/abs/1602.04938.

[7] Scott M. Lundberg and Su-In Lee. "A unified approach to interpreting model predictions". In: *CoRR* abs/1705.07874 (2017). arXiv: 1705.07874. URL: http://arxiv.org/abs/1705.07874.

[8] *SHAP documentation*. https://shap.readthedocs.io/en/latest/index.html.

[9] *SHAP Values Explained Exactly How You Wished Someone Explained to You*. https://towardsdatascience.com/shap-explained-the-way-i-wish-someone-explained-it-to-me-ab81cc69ef30.

[10] *SHAP Beeswarm plot documentation*. https://shap.readthedocs.io/en/latest/example_notebooks/api_examples/plots/beeswarm.html.

[11] *SHAP Beeswarm plot documentation*. https://shap.readthedocs.io/en/latest/example_notebooks/overviews/An%20introduction%20to%20explainable%20AI%20with%20Shapley%20values.html#A-more-complete-picture-using-partial-dependence-plots.