The Academic College of Tel Aviv-Yaffo

THE SCHOOL OF COMPUTER SCIENCE

Drug repurposing for the SARS-CoV-2 pandemic - predicting whether an existing drug will be tested in clinical trials

September 2024

Thesis submitted in partial fulfilment of the requirements for the M.Sc. degree in the School of Computer Science of the Academic College of Tel Aviv-Yaffo

By

Kfir Avlas

The research work for the thesis has been carried out under the supervision of

Dr. Sarel Cohen

Abstract

The SARS-CoV-2 pandemic caused more than 769 million cases and 6.9 million deaths worldwide [18]. The development of completely new drugs for such a novel disease is a challenging, time-intensive process. This emphasizes the importance of drug repurposing, where treatments are found among existing drugs meant for different diseases. A promising approach to this is based on combining knowledge graphs with state-of-the-art results from graph neural networks. So far, such approaches only considered the unsupervised setting. However, since the outbreak of SARS-CoV-2 a few years ago, several clinical trials have already been conducted on multiple drugs. In this work, we revisit the established DR-COVID model and add supervision of the lists of clinical trials that were conducted. In the first phase, we used Machine Learning Grid Search techniques and trained several binary classifiers manually. We found that decision trees outperformed the other models with an AUC of 54% for predicting if a drug is likely to be in clinical trials for SARS-CoV-2. In the second phase we trained several classifiers using Automated Machine Learning techniques and improved the results to 59%. The conclusion is that now we can predict with a success of 59% whether an existing drug will be tested in COVID-19-related clinical trials. **Our preliminary results were accepted as an extended abstract in the ComplexNetworks 2023 conference. We attach first our accepted extended abstract, and afterwards we continue with the full description of this research.**

Acknowledgements

I am grateful to all of those with whom I have had the pleasure to work during this project.

- Dr. Sarel Cohen
- Mr. George Kour

Also thank to Orna Avlas for her full support along the way and excellent project review.

Drug Repurposing Using Link Prediction on Knowledge Graphs With Supervision (Extended Abstract That Was Accepted To ComplexNetworks 2023 Conference)

Kfir Avlas[1], Sarel Cohen[1], Tobias Friedrich[2], and Martin S. Krejca[3]

[1] School of Computer Science, Tel-Aviv-Yaffo Academic College, Tel-Aviv, Israel
{kfiravla, sarelco}@mta.ac.il
[2] Hasso Plattner Institute, University of Potsdam, Potsdam, Germany
tobias.friedrich@hpi.de
[3] Ecole Polytechnique, IP Paris, Palaiseau, France
martin.krejca@polytechnique.edu

Abstract. The SARS-CoV-2 pandemic caused more than 769 million cases and 6.9 million deaths worldwide. The development of completely new drugs for such a novel disease is a challenging, time-intensive process. This emphasizes the importance of *drug repurposing*, where treatments are found among existing drugs meant for different diseases. A promising approach to this is based on combining *knowledge graphs* with state-of-the-art results from graph neural networks. So far, such approaches only considered the unsupervised setting. However, since the outbreak of SARS-CoV-2 a few years ago, several clinical trials have already been conducted on multiple drugs. In this work, we revisit the established DR-COVID model and add supervision of the lists of clinical trials that were conducted. We trained several binary classifiers and found that decision trees outperformed the other models with an AUC of 54 % for predicting if a drug is likely to be in clinical trials for SARS-CoV-2.

Keywords: drug repurposing, knowledge graphs, link prediction, supervised learning

1 Introduction

Developing drugs for novel diseases is a challenging, expensive, and time-consuming task. A more cost-efficient and faster way is to repurpose existing drugs for the new disease. However, since a huge variety of approved drugs exists, it is infeasible to test all of them. A feasible alternative is to predict which existing drugs could be potentially repurposed and then run clinical trials on these candidates.

Doshi and Chepuri [4] proposed such a predictive model for the SARS-CoV-2 virus, called DR-COVID. They combine *knowledge graphs*—which condense relationships between entities like drugs, diseases, and genes—with state-of-the-art research from graph neural networks. Via link prediction in a knowledge graph for drug repurposing [5], DR-COVID proposes candidates that could be repurposed for SARS-CoV-2. This approach was later improved by Cohen et al. [3], adding more output interpretation strategies.

DR-COVID works in an unsupervised manner, which was useful when SARS-CoV-2 emerged, since no results for repurposed drug were available at that point in time. However, by now, data on clinical trials for repurposing drugs for SARS-CoV-2 exist. Taking this data into consideration can improve the predictability performance of the model.

Our Contribution. We integrate existing data on repurposed drugs for SARS-CoV-2 into DR-COVID, turning the link prediction task into a supervised setting [1]. To this end, we analyze various predictors. We find that using a decision tree works best. The resulting model achieves an area-under-the-ROC-curve (AUC) score of 54 % with respect to predicting if an existing drug is likely to be in clinical trials for SARS-CoV-2.

2 Dataset

We consider the drug repurposing knowledge graph (DRKG [5]). We restrict DRKG to 5 228 drug entities (labeled *compound*) and 33 SARS-CoV-2 variants (labeled *disease*) as well as edges of the type *compound-treats-disease*. The drug entities are a subset of all 8 070 compound entries, eliminating all entries whose name is longer than 30 characters. The reason for this choice is that some entries do not reflect the name of a drug but the name of the actual chemical compound, which are typically very long. In order to restrict to existing, commercially available drugs, we filtered the data as specified.

The existing edges in the remaining graph act as positive samples, missing edges as negative ones.

3 Model Architecture

We use traditional machine-learning techniques for supervised learning, utilizing feature vectors. In order to incorporate the relations of DRKG into such feature vectors, we preprocess the DRKG data with DR-COVID [3, 4] in the following way: DR-COVID first uses TRANSE [2] in order to embed graph data into a vector space, retrieving feature vectors. Afterward, DR-COVID reduces the dimensionality of these feature vectors further, using a graph neural network. The resulting low-dimensional feature vectors are then used as input for our supervised binary classification task of whether a drug is going to be utilized in a clinical trial for SARS-CoV-2 or not.

4 Experiments

We evaluate various supervised machine-learning models on parts of our DRKG subset. Afterward, we compute the AUC score of each model and determine which model makes the best predictions.

4.1 Setup

We consider the following models: k-nearest neighbors (k-NN), decision tree, random forest, and multi-layer perceptron (MLP).

Each of these models is trained on 20 % of our DRKG subset via 10-fold cross-validation. During this process, for each model, we search for optimal parameters via a grid search. Afterward, the model is tested on the remaining 80 % of the dataset. Last, we compute the area under the receiver-operating characteristic (ROC) curve for each model.

4.2 Results and Discussion

The AUC score of all our tested approaches are stated in Table 1. We see that out of all models, the decision tree works best in terms of AUC score on the testing data. The random forest and MLP both come in second.

Interestingly, the AUC score of the decision tree on the testing data is higher than the AUC score of all other models on the training data, except for the random forest. This shows that most models predict rather poorly. Further, while the AUC score of the random forest on the training data is 1, its score on the testing data is far lower, showing that it overfits too strongly to the data.

Last, even the AUC score of the random forest, which is the best among all models, is still not too far from 0.5. This means that its predictive power is close to random and that there is substantial room for improvement. The models we analyzed are most likely not well suited for the task at hand.

5 Conclusion

None of the models we tested are particularly well suited for predicting whether an existing drug will be repurposed for SARS-CoV-2 treatment, with the resulting AUC scores being close to a fully random classifier (which has a score of 0.5). It would be interesting to see why this performance is so poor. Similarly, future research should investigate why the random forest overfits so severely to the data. Last, an important next step is to compare the results to the prediction quality of the unsupervised setting, which requires a setup that allows for a fair comparison of both approaches.

References

1. Avlas, K.: Drug repurposing for the SARS-CoV-2 pandemic. https://github.com/KfirAvlas/MTA (2023)
2. Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Neural Information Processing Systems (NeurIPS). pp. 2787–2795 (2013)
3. Cohen, S., Hershcovitch, M., Taraz, M., Kißig, O., Issac, D., Wood, A., Waddington, D., Chin, P., Friedrich, T.: Improved and optimized drug repurposing for the SARS-CoV-2 pandemic. Plos One (2023)

Table 1. The best AUC score of each model for the test and the training data, as explained in Section 4.1. The scores are rounded to four decimal places. The scores were achieved with the parameters mentioned in the second column. Please also refer to Section 4.2.

Model	**Parameters**	**AUC Score**	
k-NN	algorithm: ball_tree, n_neighbors: 12, weights: uniform	train test	0.5075 0.5000
decision tree	criterion: entropy, max_features: sqrt, min_samples_split: 10, splitter: random	train test	0.6879 0.5485
random forest	criterion: log_loss, max_features: sqrt, n_estimators: 1500	train test	1.0000 0.5200
MLP	activation: tanh, hidden_layer_sizes: 20, learning_rate: invscaling, solver: adam	train test	0.5324 0.5200

4. Doshi, S., Chepuri, S.P.: Dr-COVID: Graph neural networks for SARS-CoV-2 drug repurposing. CoRR (2020)
5. Ioannidis, V.N., Song, X., Manchanda, S., Li, M., Pan, X., Zheng, D., Ning, X., Zeng, X., Karypis, G.: DRKG - drug repurposing knowledge graph for Covid-19. https://github.com/gnn4dr/DRKG/ (2020)

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

With the novel coronavirus, a global pandemic with serious socio-economic implications for most parts of our daily lives is active [4]. The limited ability to take precautions for an unsuspected event like this and the rapid spread make finding an effective treatment as necessary as difficult, since the disease-specific knowledge is limited at the beginning and human lives are lost every day. Known and approved drugs happen to be well-studied, thus, they pose a good starting point for swift development of treatments, and an emerging tactic in fighting the pandemic [17]. DrugBank, an extensive database compiling information about drugs approved by the US Food and Drug Administration as well as experimental drugs, contained more than 2300 approved drugs and over 4500 experimental drugs as of 2018; both with a strong upward trend [1]. This emphasizes the need for computer aided development of treatments. Drug repurposing with knowledge graphs, as first described by [2] is the current state-of-the-art approach for finding possible treatments for novel diseases among known drugs using machine learning. Applying drug repurposing allows for a better way to maneuver through the pandemic. It can lead to better treatments for patients infected with one of the COVID-19 strains and a better understanding of the characteristics of the individual strains. Today, we approach the problem of drug repurposing using machine learning, focusing on supervised classifier. The idea of predicting unknown links between entities in a knowledge graph is traditionally known as Collaborative Filtering, as described by [14]. Despite researchers around the world working on this task, no effective treatments have been developed at the time. This emphasizes the importance of drug repurposing, where treatments are found among existing drugs that are meant for different diseases. A common approach to this is based on knowledge graphs that condense relationships between entities like drugs, diseases and genes. Graph neural networks (GNNs) can then be used by predicting links in such knowledge graphs. Expanding on state-of-the-art GNN research, Doshi et al. developed these Dr-Covid model [7]. Extension of their work presented by Sarel Cohen et al. by using additional output interpretation strategies [5]. The research uses the concept of graph embeddings, which map fixed-size feature vectors to graph nodes and relations based on deep neural networks (DNNs). The best aggregation strategy derives a top-100 ranking of 8,070 candidate drugs, 32 of which are currently being tested in COVID-19-related clinical trials. So far, such approaches only considered the unsupervised setting. However, since the outbreak of SARS-CoV-2 a few years ago, several clinical trials have already been conducted on multiple drugs. In this work, we revisit the established DR-COVID model and add supervision of the lists of clinical trials that were conducted.

1.2 Aims and Objectives

DR-COVID works in an unsupervised manner, which was useful when SARS-CoV- 2 emerged, since no results for repurposed drug were available at that point in time. However, by now, data on clinical trials for repurposing drugs for SARS-CoV-2 exist. Taking this data into consideration can improve the predictability performance of the model. We integrate existing data on repurposed drugs for SARS-CoV-2 into DR-COVID, turning the link prediction task into a supervised setting. To this end, we analyze various predictors. The goal was to build a good machine learning model which can then be used to **predict whether an existing drug will be tested in COVID-19-related clinical trials.**

1.3 Solution Approach

1.3.1 First Phase - Grid Search Cross Validation

Exhaustive search over specified parameter values for an estimator. Grid Search Cross-Validation is a powerful technique for fine-tuning the hyperparameters of machine learning models. It allows us to systematically search through a predefined set of hyperparameter values to find the combination that results in the best model performance.

Our goal in this phase was to test several well-known machine learning models with different hyperparameters to select the best model suited to our problem. We compared the results using various machine learning scoring metrics.

1.3.2 Second Phase - Automated Machine Learning

In the second phase, we utilized Automated Machine Learning (AutoML) techniques to obtain a more accurate model and improve prediction results. AutoML is the process of automating the tasks of applying machine learning to real-world problems. It is the combination of automation and machine learning. AutoML potentially includes every stage from beginning with a raw dataset to building a machine learning model ready for deployment. AutoML was proposed as an artificial intelligence-based solution to the growing challenge of applying machine learning. Automating the process of applying machine learning end-to-end additionally offers the advantages of producing simpler solutions, faster creation of those solutions, and models that often outperform hand-designed models [8].

1.4 Summary of Contributions and Achievements

Using Grid Search Cross-Validation, the best model performance on the test set was achieved with a decision tree classifier. Decision trees are a non-parametric, supervised learning method used for classification [6]. This classifier reached a 54% success rate on the test set when using the hyperparameters returned from the grid search. An improved model was achieved by applying Automated Machine Learning (AutoML) techniques, resulting in a success rate of 59% on

the test set. We can now conclude that the model can predict with 59% accuracy whether an existing drug will be tested in COVID-19-related clinical trials. These results demonstrate a significant improvement compared to the unsupervised approach presented earlier by [17]

2 Methodology

2.1 Supervised Learning

Supervised learning is a type of machine learning where a model is trained on labeled data. This means the training dataset contains both the input features and the corresponding correct output (target). The goal of supervised learning is to learn a mapping or relationship between the input and the output so that the model can accurately predict the target for unseen data. Algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs [11]. The data consists of a set of training examples. Each training example has one or more inputs and the desired output. Through iterative optimization of an objective function, supervised learning algorithms learn a function that can be used to predict the output associated with new inputs.

2.1.1 Key Concepts

Training Data: A set of examples where each example consists of an input and a known output. The model learns from these examples. *Input*: Features (variables) used for prediction. *Output*: Label or target value (what the model is supposed to predict). *Model*: A mathematical function or algorithm that maps input data to the output. The model uses the input data to learn patterns and relationships between the inputs and the outputs. *Objective*: Minimize the difference between the predicted output and the actual output. This is done by iteratively adjusting the model's parameters using optimization techniques. *Prediction*: Once trained, the model is used to predict outputs for new, unseen data where the correct output is not known.

2.1.2 Supervised Learning Process

Data Collection: Collect labeled data that consists of input-output pairs. *Training*: The model is fed the training data, and it adjusts its internal parameters to minimize the error between its predictions and the actual outputs. *Validation*: A portion of the data (validation set) is used to tune the model parameters (like hyperparameters) and avoid overfitting. *Testing*: The model is tested on new, unseen data (test set) to evaluate its performance. *Prediction*: After training, the model can predict outputs for new inputs.

2.1.3 Types of Supervised Learning tasks

Classification: The goal is to predict a discrete label or category. *Regression*: The goal is to predict a continuous value.

In order to predict whether an existing drug will be tested in clinical trials we used binary classification algorithms.

2.2 Knowledge Graphs

A knowledge graph is a type of database that stores information in a way that allows for connections and relationships to be easily identified and analysed. It is a structured representation of knowledge that captures the relationships between different entities, concepts, and ideas [9]. Knowledge graphs are often used in artificial intelligence and machine learning applications to help computers understand and interpret complex information. In the realm of machine learning, a knowledge graph is a graphical representation that captures the connections between different entities. It consists of nodes, which represent entities or concepts, and edges, which represent the relationships between those entities. The basic elements of a knowledge graph are entities, relations, and attributes. It is a unique type of graph database, and the relation is often single-directional. It is the relation that distinguishes Knowledge Graphs from the traditional structured data formats as the relation provides proximity and remoteness, the path to reach a final target, and similarities between entities not only based on attributes/features but also on their surrounding neighbours. Another important difference lies in the attributes associated with the entities. These relations and attributes make a Knowledge Graph a graph: the entities are nodes, and relations are the edges connecting the nodes. Drug repurposing with knowledge graphs, as first described by [2], is the current state-of-the-art approach for finding possible treatments for novel diseases among known drugs using machine learning. The idea of predicting unknown links between entities in a knowledge graph is traditionally known as Collaborative Filtering, as described by [5].

2.2.1 Knowledge Graph Embedding

In representation learning, knowledge graph embedding (KGE) is a machine learning task of learning a low-dimensional representation of a knowledge graph's entities and relations while preserving their semantic meaning. Leveraging their embedded representation, knowledge graphs (KGs) can be used for various applications such as link prediction, triple classification, entity recognition, clustering, and relation extraction [10].

Our data is based on the Drug Repurposing Knowledge Graph (DRKG), which compiles data from different biomedical databases and creates neighborhood graph embedding which maps fixed-size feature vectors to graph nodes and relations.

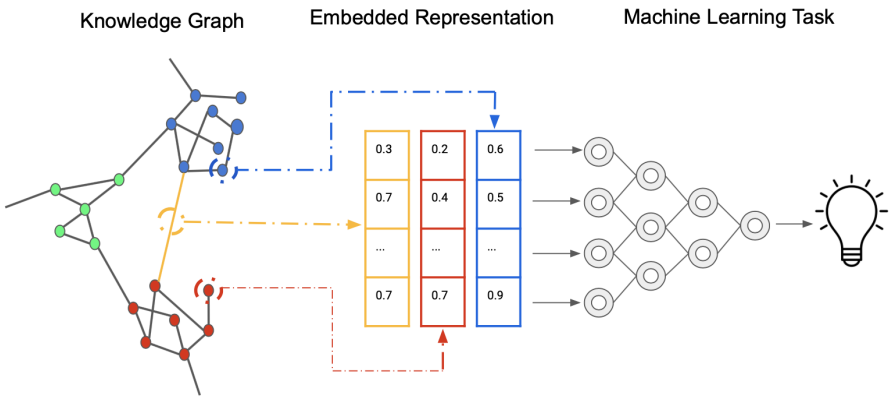

Figure 1: The embedding of a knowledge graph translates each entity and relation of a knowledge graph into a vector of a given dimension, called embedding dimension [10].

2.3 The Data-Set

Sarel Cohen et al. work [5] relies on the Drug Repurposing Knowledge Graph (DRKG), which compiles data from different biomedical databases and uses 98 edge types between 4 entity types, namely gene, compound, anatomy and disease. In particular, it contains drugs and substances as compound entities, as well as different COVID-19 variants as disease entities. Finally there are 8070 drug entities and 33 different COVID-19 entities. The edge types include compound-treats-disease edges, which is the kind of edge the model predicts. We restrict DRKG to 5228 drug entities (labeled compound) and 33 SARS-CoV-2 variants (labeled disease) as well as edges of the type compound-treats-disease. The drug entities are a subset of all 8070 compound entries, eliminating all entries whose name is longer than 30 characters. The reason for this choice is that some entries do not reflect the name of a drug but the name of the actual chemical compound, which are typically very long. In order to restrict to existing, commercially available drugs, we filtered the data as specified.

2.3.1 The input

The input graph is represented by a two-dimensional matrix. different COVID-19 variants as disease vs drugs approved by the US Food and Drug Administration as drugs. Every cell i,j in the matrix has a value which is the prediction score of a disease j to be treated by a drug i. In addition, as the disease has already been studied, the data-set contains an output target variable showing whether a disease was tested in COVID-19-related clinical trials. Finally the input consist of multiple labeled input features (of disease , drug, score) and a target binary variable.

2.3.2 Output Target

Since the disease has already been studied, we have partial knowledge about effective drug treatments and drugs tested in clinical trials. For this reason, we investigated which drugs have been found effective or tested in clinical trials. To achieve this, we utilized an additional data source. The clinical trials dataset includes columns for study types, conditions, and interventions. We used this information to map drugs involved in COVID-19-related clinical trials by searching for drug names within the intervention study types.

2.3.3 The Data-Set Structure

Table 1: Example of a data-set structure

Drug 1	COVID-1-score	...COVID-33-score	Is in clinical trials
Drug 2	COVID-1-score	...COVID-33-score	Is in clinical trials
Drug 5228	COVID-1-score	...COVID-33-score	Is in clinical trials

2.4 Train Test Split

Learning the parameters of a prediction function and testing it on the same data is a methodological mistake: a model that would just repeat the labels of the samples that it has just seen would have a perfect score but would fail to predict anything useful on yet-unseen data. This situation is called overfitting. To avoid it, it is common practice when performing a (supervised) machine learning experiment to hold out part of the available data as a test set [16].

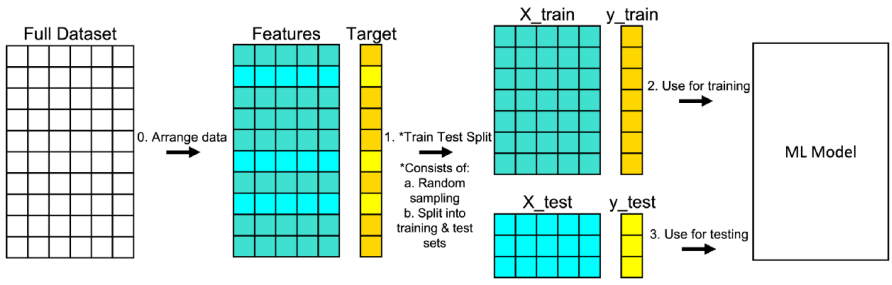

Figure 2: common practice when performing a (supervised) machine learning experiment to hold out part of the available data as a test set [16].

2.5 Grid Search Cross Validation

Finding the optimal tuning parameters for a machine learning problem can often be very difficult. We may encounter overfitting, which means our machine learning model trains too specifically on our training dataset and causes higher levels of error when applied to our test/holdout datasets. Or, we may run into underfitting, which means our model doesn't train specifically enough to our training dataset. This also leads to higher levels of error when applied to test/holdout datasets. When conducting a normal train/validation/test split for model training and testing, the model trains on a specific randomly selected portion of the data, validates on a separate set of data, then finally tests on a holdout dataset. In practice this could lead to some issues, especially when the size of the dataset is relatively small, because you could be eliminating a portion of observations that would be key to training an optimal model. Keeping a percentage of data out of the training phase, even if its 15–25% still holds plenty of information that would otherwise help our model train more effectively. In comes a solution to our problem — Cross Validation. Cross validation works by splitting our dataset into random groups, holding one group out as the test, and training the model on the remaining groups. This process is repeated for each group being held as the test group, then the average of the models is used for the resulting model. One of the most common types of cross validation is k-fold cross validation, where 'k' is the number of folds within the dataset. Using k 5 is a common first step and easy for demonstrations of this principle [15].

2.5.1 GridSearchCV

The module we utilized is sklearn's GridSearchCV, which allow us to pass our specific estimator, our grid of parameters, and our chosen number of cross validation folds. Some of the main parameters are below:

estimator — this parameter allows us to select the specific model we are choosing to run, in our case Decision Trees Classification. *param grid* — this parameter allows us to pass the grid of parameters we are searching. This grid must be formatted as a dictionary with the key corresponding to the specific estimators parameter names, and the values corresponding to a list of values to pass for the specific parameters. *cv* — this parameter allows us to change the number of folds for the cross validation.

2.5.2 Imbalanced Data

Typically refers to a problem with classification models where the classes are not represented equally - classification data set with skewed class proportions.

Our final data-set consists of 250 "true" labeled drugs out of 5228(5% positive rate) which means that the data-set is imbalanced . In order to deal with it we used the following metrics in the training process as described below:

2.5.3 Stratifying

K-Folds cross-validation divides all the samples in K groups of samples, called folds of equal sizes. The prediction function is learned using K-1 folds , and the fold left out is used for testing. Stratifying is a variation of k-fold which returns stratified folds: each set contains approximately the same percentage of samples of each target class as the complete set [15].

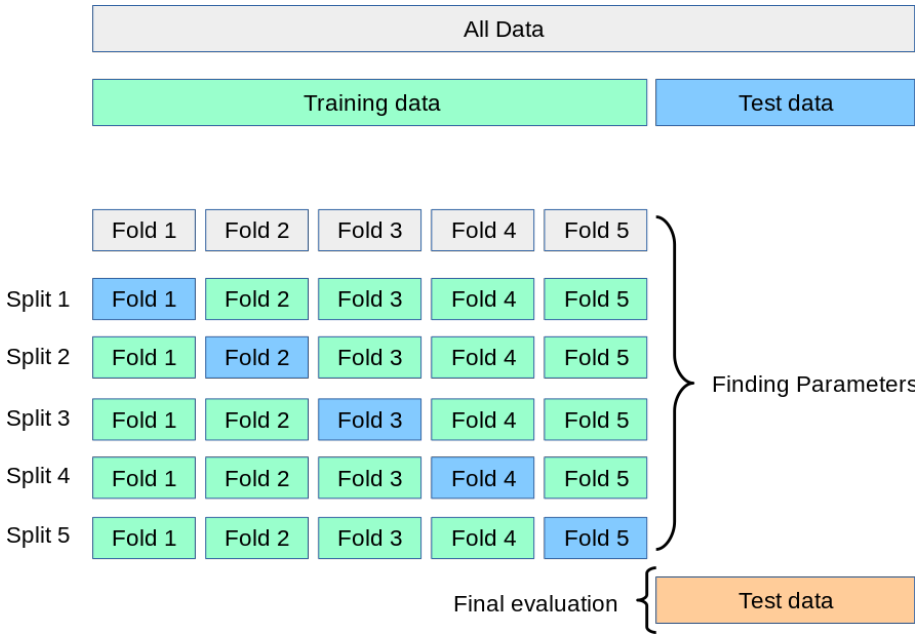

Figure 3: K-Folds cross-validation.

2.5.4 Area Under The Receiver-Operating Characteristic Curve

AUC-ROC curve is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes [3].

True Positive Rate

The true positive rate (**TPR**), or the proportion of all actual positives that were classified correctly as positives, is also known as recall.

$$\text{Recall (or TPR)} = \frac{\text{correctly classified actual positives}}{\text{all actual positives}} = \frac{TP}{TP + FN}$$

Figure 4: True positive rate.

False Positive Rate

The false positive rate (**FPR**) is the proportion of all actual negatives that were classified incorrectly as positives, also known as the probability of false alarm. It is mathematically defined as:

$$\text{FPR} = \frac{\text{incorrectly classified actual negatives}}{\text{all actual negatives}} = \frac{FP}{FP + TN}$$

Figure 5: False positive rate.

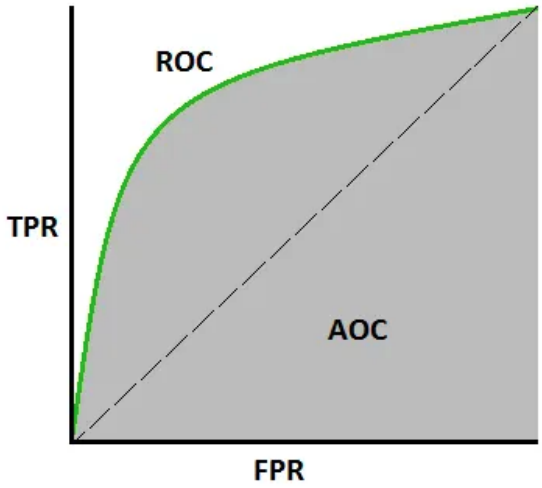

Figure 6: The ROC curve is plotted with TPR against the FPR where TPR is on the y-axis and FPR is on the x-axis [3]

2.6 Ensemble Modeling

Ensemble modeling is a process where multiple diverse base models are used to predict an outcome. The motivation for using ensemble models is to reduce the generalization error of the prediction. As long as the base models are diverse and independent, the prediction error decreases when the ensemble approach is used. If the predicted value is a category, each constituent modeling algorithm contributes a "vote" for a category, and the category with the most votes wins for a given record in the list (majority rule). Some analytic tools provide other heuristics to replace the majority rule. If the predicted value is a number, the mean value for the predictions (or some other heuristic value) is calculated across all model predictions.

2.7 Setup

2.7.1 Phase 1

We consider the following models: k-nearest neighbors (k-NN), decision trees, random forest, and multi-layer perceptron (MLP). Each of these models was trained on 20% of our DRKG subset via 10-fold crossvalidation. During this process, for each model, we search for optimal hyper parameters via a grid search. Afterward, the model was tested on the remaining 80% of the dataset. Last, we compute the area under the receiver-operating characteristic (ROC) curve for each model.

2.7.2 Phase 2

We executed Automated Machine Learning by using *mljar-supervised* Automated Machine Learning Python package. It abstracts the common way to preprocess the data, construct the machine learning models, and perform hyper-parameters tuning to find the best model. The *mljar-supervised* help us with: explaining and understanding our data, trying many different machine learning models, creating Markdown reports from analysis with details about all models, saving, re-running and loading the analysis and Machine Learning models [13].

3 Results

3.1 Phase 1

The AUC score of all our tested approaches are stated in the following table. We see that out of all models, the decision tree works best in terms of AUC score on the testing data. The random forest and MLP both come in second. Interestingly, the AUC score of the decision tree on the testing data is higher than the AUC score of all other models on the training data, except for the random forest. This shows that most models predict rather poorly. Further, while the AUC score of the random forest on the training data is 1, its score on the testing data is far lower, showing that it overfits too strongly to the data.

Table 2: The best AUC score of each model for the test and the training data

Model	**Parameters**	**AUC Score**
k-NN	algorithm: ball tree, n neighbors: 12, weights: uniform	train 0.5075 test 0.5000
decision tree	criterion: entropy, max features: sqrt, min samples split: 10, splitter: random	train 0.6879 test 0.5485
random forest	criterion: log loss, max features: sqrt, n estimators: 1500	train 1.0000 test 0.5200
MLP	activation: tanh, hidden layer sizes: 20, learning rate: invscaling, solver: adam	train 0.5324 test 0.5200

3.1.1 Decision Tree Classifier

The parameters passed to grid search:

- criterion: function to measure the quality of a split
- max features: number of features to consider when looking for the best split
- min samples split: minimum number of samples required to split an internal node
- splitter: strategy used to choose the split at each node

Table 3: Grid Search Results

Model	Parameter	Best Value
sklearn.tree.DecisionTreeClassifier	criterion	entropy
sklearn.tree.DecisionTreeClassifier	max features	sqrt
sklearn.tree.DecisionTreeClassifier	min samples split	10
sklearn.tree.DecisionTreeClassifier	splitter	random

3.1.2 Final Score

Table 4: Score Results

Score metric	Source	Score
roc-auc score	grid search	0.67
roc-auc score	train set	0.69
roc-auc score	test set	0.54

3.1.3 Phase Summary

The best model score on the test set achieved when running a decision tree classifier model [6] with a success of 54%. The conclusion is that now we can predict with a success of 54% whether an existing drug will be tested in COVID-19-related clinical trials using this model.

3.2 Phase 2

3.2.1 Steps of AutoML

The training of mljar-supervised AutoML is divided into steps. Each step represents the actions that are common in the process of searching for the best performing Machine Learning model in the ML pipeline. Below are described steps of AutoML.

Simple algorithms

The first step in the AutoML training is to check the simplest algorithms to get quick insights:

- Baseline - provides the baseline result without complex ML involved. If further results are much better than Baseline results, it justify the use of Machine Learning.
- Decision Tree- provides results for simple tree with maximum depth up to 4.
- Linear- provides simple Machine Learning model. The models in this step should be quickly trained, so we will get fast intuition about our data and the solved problem.

Default algorithms:

In this step, the models are trained with default hyperparameters. Regardless of the data, the hyperparameter values used are always the same for each algorithm. In this step, we can compare the results of default models from other datasets and get intuition about our problem complexity. The following algorithms can be fitted in this step:

- Random Forest
- Extra Trees
- Xgboost
- LightGBM
- CatBoost
- Neural Network
- Nearest Neighbors

There is exactly one model fitted for each algorithm in this step. (Each algorithm has one set of default hyperparameter values for each ML task).

Not so random

This step performs Random Search over defined set of hyperparameters (hence the name).

Golden features

Golden Features are new features constructed from original data which have great predictive power. They can be constructed based on the original features. The common way to construct them is to try features differences or ratios. The procedure to find Golden Features: Generate all possible unique pairs of original features, for each pair of features construct a new feature with substract or division operators, then there is a computed score on test samples for each feature. The score is logloss metric. Newly generated features are sorted based on the score and are inserted into the training data.

Insert random feature

This step is a first part of **Feature Selection** procedure. During this step:

- The random feature is added to the original data.
- The best model is selected from previous steps and its hyperparameters are used to train model with newly inserted feature.
- After the training, the feature importance (permutation-based) is computed.
- Features with lower importance than random feature are saved.

Feature Selection

This step is a second part of Feature Selection procedure. In this step:

- The best model for each algorithm is selected.
- Its hyperparameters are reused and it is trained using only selected features.
- If all features are important, this step will be skipped.

Hill climbing

In the hill climbing step the fine tuning of models is done. There can be several hill climbing steps. In each hill climbing step, the top performing models from each algorithm are tuned further. If a model is selected for further tuning, then only one randomly selected hyperparameter from its setting is changed. The selected hyperparameter will be changed in two directions.

Ensemble

During ensemble step all models from previous steps are ensembled. After all Level-0 models are trained step by step (like simple algorithms, default algorithms, not so random, golden features, insert random feature and feature selection), it will start to process the Ensemble step. In this process, the weight values according to the all Level-0 model will be calculated and become Ensemble model.

Stack

In this step models are stacked. The original input data is used to train several models of Level-0, and the prediction results of the models are added to the original data then generate new input data, which will be used to train the Stacked Model of Level-1 and obtain the final prediction result.

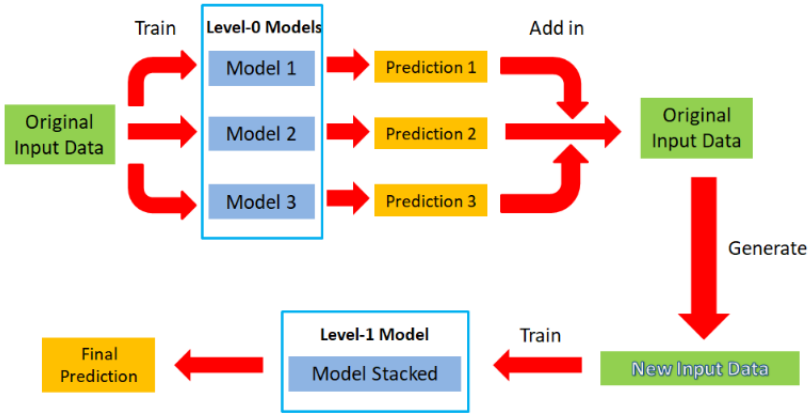

Figure 7: Stack process.

Ensemble stacked

Ensemble stacking means to ensemble the above-mentioned Level-0 and Level-1 models with different weight values and predict final results.

3.2.2 The Selected Model

The AutoML process is using a many algorithms described at [12].

The ensemble model achieved the best score on the training set in terms of AOC metric, and therefore it is the model selected by the Automated Machine Learning process.

Table 5: Ensemble structure

Model	Weight
1_Linear_GoldenFeatures	2
42_CatBoost	1
4_Default_CatBoost	1
7_Xgboost	1

Table 6: Leader board

Model	Model Type	AUC Metric Value
Ensemble	Ensemble	0.8223593671521850
1_Linear_GoldenFeatures	Linear	0.8125489703666500
1_Linear_GoldenFeatures_RandomFeature	Linear	0.8110082872928180
1_Linear	Linear	0.8107621798091410
4_Default_CatBoost	CatBoost	0.8030637870416880
35_Xgboost	Xgboost	0.800876443997991
40_Xgboost	Xgboost	0.800876443997991
41_Xgboost	Xgboost	0.800876443997991
19_RandomForest	Random Forest	0.7949177548970370
31_CatBoost	CatBoost	0.7945768458061280
19_RandomForest_SelectedFeatures	Random Forest	0.7932383224510300
32_Xgboost_SelectedFeatures	Xgboost	0.7930719487694630
43_Xgboost_SelectedFeatures	Xgboost	0.7930719487694630
44_Xgboost_SelectedFeatures	Xgboost	0.7930719487694630
17_CatBoost	CatBoost	0.7924654696132600
34_Xgboost	Xgboost	0.7917899296835760
42_CatBoost	CatBoost	0.7913247112004020
16_CatBoost	CatBoost	0.7899692365645400
22_RandomForest	Random Forest	0.7895178302360620
33_Xgboost_SelectedFeatures	Xgboost	0.7894035660472120
30_RandomForest_SelectedFeatures	Random Forest	0.7892139628327470
20_RandomForest	Random Forest	0.7882232546459070
6_Default_RandomForest	Random Forest	0.787940105474636
9_Xgboost_SelectedFeatures	Xgboost	0.7878346308387750
29_RandomForest	Random Forest	0.7858576092415870
19_RandomForest_GoldenFeatures	Random Forest	0.7835145655449520
27_CatBoost	CatBoost	0.7829758915118030
9_Xgboost	Xgboost	0.7821289552988450
18_CatBoost	CatBoost	0.7812788799598190
28_CatBoost	CatBoost	0.7798731793068810
11_LightGBM	LightGBM	0.7798681567051730
46_LightGBM_SelectedFeatures	LightGBM	0.7785792315419390
21_RandomForest	Random Forest	0.7762060522350580
45_LightGBM	LightGBM	0.773360120542441
11_LightGBM_SelectedFeatures	LightGBM	0.7730662983425410
37_LightGBM_SelectedFeatures	LightGBM	0.7698706680060270
15_CatBoost	CatBoost	0.7665381717729780
36_LightGBM	LightGBM	0.7604664741336010
10_Xgboost	Xgboost	0.7524108488196890
7_Xgboost	Xgboost	0.7515927925665500
4_Default_CatBoost_SelectedFeatures	CatBoost	0.7341605976896030
12_LightGBM	LightGBM	0.7337882973380210
3_Default_Xgboost	Xgboost	0.7281165243596180
4_Default_CatBoost_GoldenFeatures	CatBoost	0.7255631592164740
14_LightGBM	LightGBM	0.7168194374686090
5_Default_NeuralNetwork	Neural Network	0.7141857106981420
13_LightGBM	LightGBM	0.7041310899045710
2_Default_LightGBM	LightGBM	0.6993640130587650
8_Xgboost	Xgboost	0.697426544450025
25_NeuralNetwork	Neural Network	0.6291913611250630

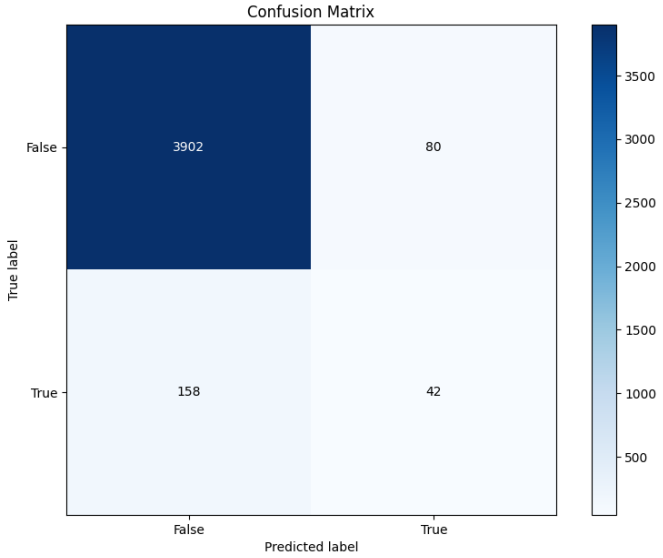

Figure 8: Confusion Matrix

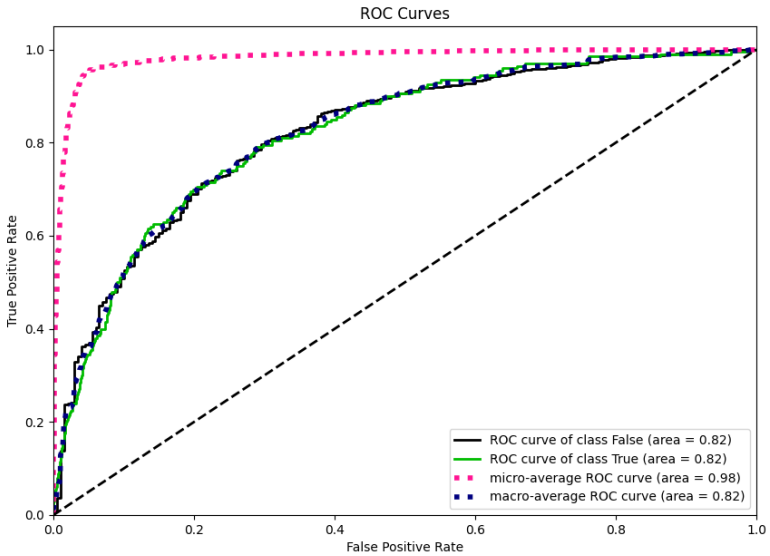

Figure 9: ROC Curve

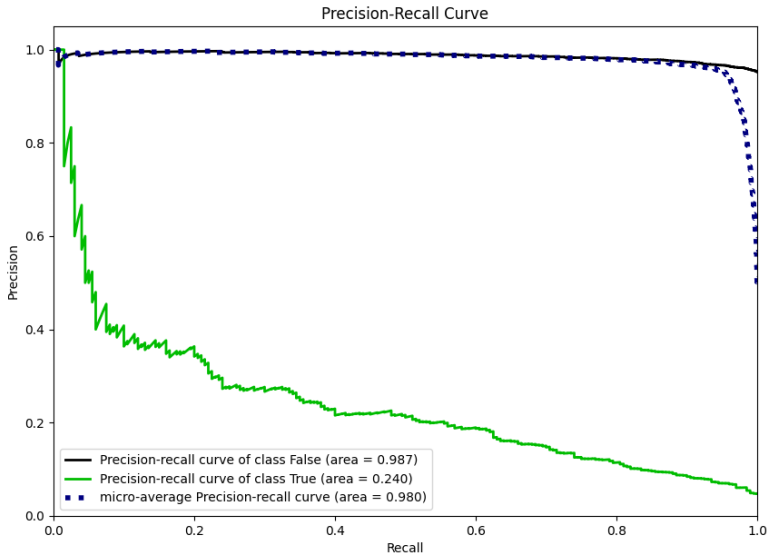

Figure 10: Precision - Recall Curve

Table 7: Golden Features

Feature1	**Feature2**	**Operation**	**Score**
COVID_27	COVID_2	Sum	0.1671561082
COVID_20	COVID_6	Multiply	0.1809569985
COVID_28	COVID_22	Sum	0.1867115977
COVID_25	COVID_13	Ratio	0.1875402709
COVID_29	COVID_11	Multiply	0.188816341
COVID_20	COVID_17	Multiply	0.190802086
COVID_27	COVID_25	Sum	0.1916766833
COVID_20	COVID_1	Multiply	0.192422798
COVID_15	COVID_9	Sum	0.192925978
COVID_18	COVID_15	Multiply	0.1947446547

Dropped Features:

- COVID_9
- random_feature
- COVID_5
- COVID_25_ratio_COVID_13

- COVID_3, COVID_21
- COVID_28

3.2.3 Final Score

Table 8: Score Results

Score metric	**Source**	**Score**
roc-auc score	train set	0.82
roc-auc score	test set	0.59

3.2.4 Phase Summary

The best model score on the test set was achieved by running an ensemble model consisting of CatBoost, XGBoost, and a linear base model. The ensemble incorporated newly constructed 'golden features' derived from the original data, which had a significant impact on the model's predictive performance. We achieved an overall success rate of 59%. The conclusion is that this model now enables us to predict with 59% accuracy whether an existing drug will be tested in COVID-19-related clinical trials.

4 Conclusions and Future Work

4.1 Conclusions

With the novel coronavirus, a global pandemic with serious socio-economic implications for most parts of our daily lives is active. The limited ability to take precautions for an unsuspected event like this and the rapid spread make finding an effective treatment as necessary as difficult, since the disease-specific knowledge is limited at the beginning and human lives are lost every day. Known and approved drugs happen to be well-studied, thus, they pose a good starting point for swift development of treatments, and an emerging tactic in fighting the pandemic. Drug repurposing with knowledge graphs [9], is the current state-of-the-art approach for finding possible treatments for novel diseases among known drugs using machine learning. Applying drug repurposing allows for a better way to maneuver through the pandemic. It can lead to better treatments for patients infected with one of the COVID-19 strains and a better understanding of the characteristics of the individual strains. Expanding on state-of-the-art GNN research, Doshi et al. developed thes DR-Covid model [7]. Extension of their work presented by Sarel Cohen et al. by using additional output interpretation strategies [5]. The research approaches the problem of drug repurposing using machine learning, focusing on deep learning methods for predicting unknown links between entities in a knowledge graph. DR-COVID works in an unsupervised manner, which was useful when SARS-CoV- 2 emerged, since no results for repurposed drug were available at that point in time. However, by now, data on clinical trials for repurposing drugs for SARS-CoV-2 exist. Taking this data into consideration can improve the predictability performance of the model. *Our Contribution.* We integrate existing data on repurposed drugs for SARS-CoV-2 into DR-COVID, turning the link prediction task into a supervised setting. To this end, we analyze various predictors. We find that using an ensemble model, consist of specific configuration of Linear, CatBoost and Xgboost base models works best. The resulting model achieves an area-under-the-ROC-curve (AUC) score of 59% with respect to predicting if an existing drug is likely to be in clinical trials for SARS-CoV-2. The conclusion is that now we can predict with a success of 59% whether an existing drug will be tested in COVID-19-related clinical trials.

4.2 Future work

There are several actions we can take in the future to improve our model and achieve higher prediction scores. One potential improvement is to refine our data collection pre-process to obtain more useful data. For example, when filtering out drug names longer than 30 characters, we removed 2,842 drugs, which could have enhanced the learning model. In the future, many existing drugs might be tested in COVID-19-related clinical trials, and this new data can be used to further improve the model and its predictive ability. Additionally, this could increase the rate of positive labels (currently around 5%) and address the issue of

imbalanced data, leading to better model performance and prediction accuracy.

5 Source Code

`https://github.com/KfirAvlas/MTA/blob/main/README.md`

6 References

[1] Wishart DS Feunang YD Guo AC Lo EJ Marcu A Grant JR et al. *The DrugBank database*. 2018. URL: https://academic.oup.com/nar/article/46/D1/D1074/4602867.

[2] Thor KB. Ashburn TT. *Drug repositioning: identifying and developing new uses for existing drugs*. 2004. URL: https://www.nature.com/articles/nrd1468.

[3] *AUC-ROC curve*. 2023. URL: https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5.

[4] Nicola M Alsafi Z Sohrabi C Kerwan A Jabir A Losifidies C. *The socio-economic implications of the coronavirus pandemic (COVID-19)*. 2020. URL: https://www.sciencedirect.com/science/article/pii/S1743919120303162.

[5] S. Cohen. *Improved and optimized drug repurposing for the SARS-CoV-2 pandemic*. 2023. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0266572.

[6] *Decision tree learning*. 2023. URL: https://en.wikipedia.org/wiki/Decision_tree_learning.

[7] Siddhant Doshi and Sundeep Prabhakar Chepuri. *Dr-COVID: Graph Neural Networks for SARS-CoV-2 Drug Repurposing*. 2020. arXiv: 2012.02151 [cs.LG].

[8] From Wikipedia the free encyclopedia. *Automated machine learning*. URL: https://en.wikipedia.org/wiki/Automated_machine_learning.

[9] *Knowledge Graph*. URL: https://en.wikipedia.org/wiki/Knowledge_graph.

[10] *Knowledge Graph Embedding*. 2023. URL: https://en.wikipedia.org/wiki/Knowledge_graph_embedding.

[11] *Machine learning*. 2023. URL: https://en.wikipedia.org/wiki/Machine_learning.

[12] *mljar-algorithms*. URL: https://supervised.mljar.com/.

[13] *mljar-supervised*. URL: https://supervised.mljar.com/.

[14] Badrul Sarwar George Karypis Joseph Konstan John Riedi. *Item-based collaborative filtering recommendation algorithms*. 2001. URL: https://dl.acm.org/doi/abs/10.1145/371920.372071.

[15] *SKlearn Cross Validation*. 2023. URL: https://scikit-learn.org/stable/modules/cross_validation.html.

[16] *SKlearn Train Test Split*. 2023. URL: https://github.com/mGalarnyk/Python_Tutorials/blob/master/Sklearn/Train_Test_Split/TrainTestSplitScikitLearn.ipynb.

[17] Shah B Modi P Sagar SR. *In silico studies on therapeutic agents for COVID-19: Drug repurposing approach*. 2020. URL: https://www.sciencedirect.com/science/article/pii/S0024320520304008.

[18] World-Health-Organization. *WHO Coronavirus (COVID-19) Dashboard*. 2023. URL: https://covid19.who.int/.