Date: 18th April 2021

THE TOXICITY PREDICTION CHALLENGE

Jaskiran Kaur, Shubham Deshmukh, Sohini Sarkar
(Maple Squad)

(CSCI555-DATA MINING AND MACHINE LEARNING)

Supervised by: Dr. Othman Soufan
Department of Computer Science
ST. FRANCIS XAVIER UNIVERSITY

1 INTRODUCTION:

Evaluation of toxicity of chemicals is of great importance in process of new drug development and approval. Undoubtedly the trials of drugs are always associated with the risk to humans. The Unsafety of these drugs has always been a concerning issue among scientists, but this process is time-consuming and laborious to carry out. This can be an obstacle in finding the toxicity of drugs and making drugs safer for individuals to consume. Nowadays to address these challenges, the computer-based toxicity prediction field is growing rapidly. Prediction is made on basis of some sophisticated machine learning (a branch of artificial intelligence) algorithms that learn from the data. Machine learning has numerous benefits such as it is rapid in dealing with big data, cost-effective in predictions, shows a high rate of accuracy. Therefore, more researchers are moving towards this technology. Different algorithms of machine learning can illustrate the different levels of performance. The level of performance depends on the factors such as datasets, patterns, computational representation, and much more.

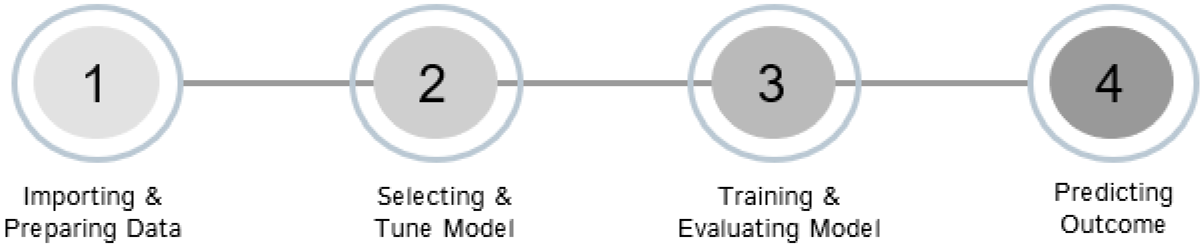


Fig 1: Project Outline.

2 DATA PREPARATION:

It is the process of collecting data, merging data, structuring, and transforming data which can be used for data visualization and data modeling. The raw data is being available for research to do further processing and it has some missing values, inaccuracies, and other errors. To resolve all these issues, data preparation and preprocessing are done.

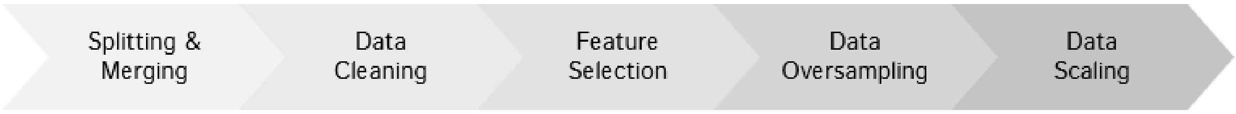


Fig 2: Data Preparation.

2.1 DATA SPLITTING & MERGING

In this competition, the datasets are available on Kaggle (largest data science community) in four different files named *train.csv, test.csv, feamat.csv, features_id_name_mapping.csv, train.csv* file contains the *"id"* and *"prediction"* column whereas the *test.csv* includes only the "*id*" column, *feamat.csv* has the features for the columns of training and testing dataset. Feamat id mapping file is only used for the information about the chemicals. The train and test files are being parsed along with this, the delimiters(_) and *"NOCAS"* string constant have been removed and replaced with a constant respectively. After parsing we split *"id"* into *"assay_id"* and *"chemical_id"*. Finally, a new dataset is prepared by merging the training and testing datasets with the *feamat.csv* file. Different models are then on these feature matrices.

2.2 DATA CLEANING

In the preprocessing part, the dataset containing the infinite values are replaced with **NaN** intermediately and were finally replaced using KNN imputer. We use this step to convert the data type of entire columns in the dataset to float. Two columns that were not significant are removed from the dataset to avoid the overfitting of the model. The columns which were of object and string datatype are converted into float.

2.3 FEATURE SELECTION

Data can be marked as food for a machine learning model, but sometimes a vast amount of data hinders the performance of the model due to overfitting. Feature selection plays an imperative part in any machine learning application. It refers to the process of selection of the most significant features of the data. In the real world, the dataset may contain irrelevant data which degrades the performance of the model while training.

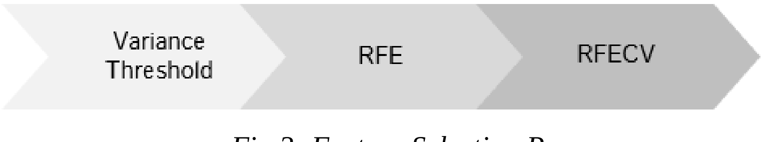


Fig 3: Feature Selection Process.

The first step of feature selection here is done using the Variance Threshold. It is a simple pre-processing step which is used here to remove feature whose variance does not satisfy the mentioned threshold value. A threshold value of *0.02* is being used, out of the initial *1077* columns, 468 columns are removed from the dataset leaving behind *609* columns to be used for further processing and training. The testing dataset is being transformed into the same dimension as the training dataset.

The second step of feature selection is done using a more complex feature selection method called Recursive feature elimination (RFE). It is one of the most popular feature selection technique and is a wrapper style method which selects feature by recursively eliminating features out of the set and making the subset smaller after each recursion, this is configured using "***step"*** parameter which is given as *0.30* (30%) and eliminates 30% features of the subset after each recursion. The RFE uses an external estimator to assign weight to each feature, ***XGB classifier*** is used as the estimator with default parameters. Another important configuration is the number of features to select (***n_features_to_select***), it's given as *200* which states the number of features to be selected using the RFE, the number is chosen by one-time hyperparameter tuning while testing, using scoring (F1-macro) of numbers in a given range of values, however, it's not mentioned in the code. Both the training and the testing data are transformed into the same dimension using the RFE.

The third and final step of feature selection is done using an extension of RFE called Recursive Feature Elimination Cross-validation (RFECV) which performs cross-validation of the different number of features while scoring them and selects the number of features automatically. Much like the RFE, RFECV also takes an external estimator and a step value, here ***Gradient Boosting Classifier*** is used as an estimator and step count is given as *0.1* (10%), other configuration like ***"min_features_to_select"***, and "***scoring"*** are set to *30* and '***f1_macro***' respectively. 40 columns are selected for the further process after this step of feature selection.

About 95% of the features are removed after the entire feature selection process.

2.4 DATA RESAMPLING

Another step in the data processing stage is resampling of the data, which is done to solve the problem of imbalanced/unequal data, Random Oversampling has been applied to the data. Random oversampling increases the size of the training data to make the number of majority and minority data equal and make the training dataset balanced. It copies data from majority samples randomly into the minority class and does not create any synthetic data. Oversampling is not applied to the testing dataset.

2.5 DATA SCALING

Feature engineering is used in machine learning to create new features/columns out of the existing ones by performing operations on to them, to ultimately improve a machine learning model's outcome performance. It is the way of changing the input into things in which the algorithms can understand better. Feature scaling is a technique that is used to scale the unordered data into a standardized and normalized way within a particular range, we have applied the standard scaling technique which removes mean and scales the data into unit variances.

3 MODEL TRAINING/TESTING:

The machine learning model is the heart of a machine learning application, using the correct machine learning model can play a significant part in any classification problem. For prediction of toxicity, choosing a model can be a cumbersome process and can give underwhelming output than expected, we are using internal validation [3.2], to validate the output of the prediction, we are using Extreme Bosting Classifier (*XGBoost Classifier*) as our final machine learning model after using and comparing with several other classification models as show in submission history [Fig 6].

3.1 HYPERPARAMETER TUNING

Machine Learning model has several features which are configured using parameters of the function, the parameters by default can produce good results but doesn't reflect on the actual capabilities of the model, Hyperparameter tuning is the process to tune the parameter of the machine learning model according to the available data and parameter combination to achieve the best performance, Randomized SearchCV is used for the XGBoost classifier in the application, the process takes sliced training data and range of different parameter values which are select randomly, it also uses cross-validation for a robust outcome of the hyperparameter. Hyperparameter tuning is a time-consuming process depending upon the hardware, it was only run once in the application (*commented in the code*) to save the processing time and the same parameters are used for further builds and submission.

3.2 INTERNAL VALIDATION

Internal validation is a practice that is used before the actual prediction by the machine learning model to validate the model's prediction based on scoring, a strong validation system reflects the actual prediction accuracy of the model, lesser the deviation between the internal score and final score the more accurate it is, it is very important to have a robust internal validation scheme for the model before making predictions since the model can overperform internally and underperform for the actual training data. We have used ***F1 macro*** scoring to check the accuracy of the model, the validation process follows the same path as the original prediction process only with lesser data. The merged training dataset is split into an 80:20 ratio with 80% training data and 20% testing data.

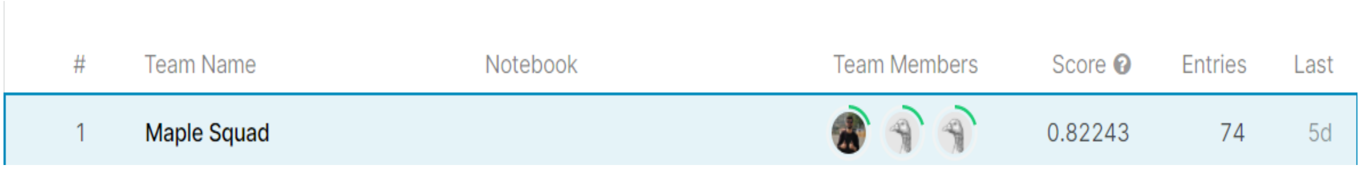
3.3 FINAL PREDICTION

After applying several data processing, feature selection, and model selection & hyperparameter tuning, the tuned model (*XGBoost Classifier*) is trained with the processed data and the prediction is made, the result is then exported to a CSV file and submitted to Kaggle for scoring, a brief scoring history can be seen in the image below [Fig 6].

4 LEADERBOARD SCORE:

We achieved 1st rank in the public leaderboard with a score of **0.82243** and 3rd in the private leaderboard with a score of **0.80294,** below are the final snapshot of both the Public [Fig 4] and Private [Fig 5] Kaggle leaderboard.

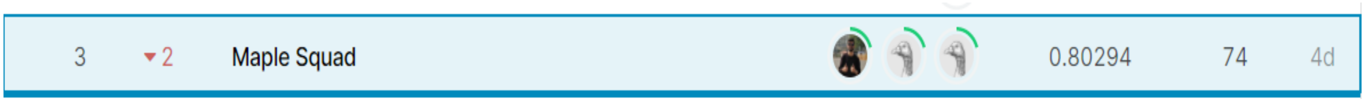
PUBLIC SCORE:



#	Team Name	Notebook	Team Members	Score	Entries	Last
1	Maple Squad			0.82243	74	5d

Fig 4: Kaggle public leaderboard.

PRIVATE SCORE:



3	▾2	Maple Squad		0.80294	74	4d

Fig 5: Kaggle private leaderboard.

SUBMISSION HISTORY:

MODEL	F1-MACRO SCORE	LEADERBOARD SCORE	COMMENTS
XGBoost	0.8085	0.8029 (*private*) 0.8208 (*public*)	Variance Threshold, RFE, RFECV, Sampling, Scaling
XGBoost	0.814	0.81722	RFE, Sampling, Scaling
XGBoost	0.794	0.80619	Hyperparameter tuning
Decision tree & Random Forest	0.745 – 0.759	0.74803 – 0.76021	Manually selected features
KNN, SVM, Naïve Bayes, SGD, AdaBoost, LDA, QDA, Neural Network, LightGBM, GBC	0.47 – 0.73	LGBM – 0.74300 No submissions for the rest of the models.	Used 30 columns, Hyperparameter tuning

Fig 6: Submission history.

COMPETITION LINK:

The Toxicity Prediction Challenge | Kaggle