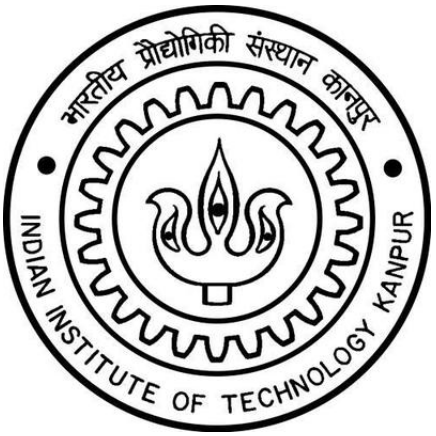

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

CS 690A Semester 2020–2021-I: Assignment 1

Predicting the Drug Resistance in Mycobacterium Tuberculosis

GROUP-3

1 INTRODUCTION

The objective of the assignment is to produce a machine learning/statistical model that improves the predictive performance of Whole Genome Sequencing (WGS) for a set of first and second line drugs. Provided a set of mutations as features, we are to fit a model on the data provided and predict whether a sample is resistant or susceptible to a particular drug. Eleven drugs were taken into consideration which are first-line drugs (rifampicin, isoniazid, pyrazinamide, and ethambutol); streptomycin; second-line injectable drugs (capre- omycin, amikacin, and kanamycin); and fluoroquinolones (ciprofloxacin, moxifloxacin, and ofloxacin). This report provides a description of the steps taken in creating the models, the processing of datasets and the results inferred from the trained models. The report also provides a reasoning and analysis of the models performance over the datasets.

2 METHODOLOGY

This section describes in particular, modifications made to the original dataset and the models considered for the task at hand.

2.1 Data Processing

Data processing involved the conversion of the original data provided into modified datasets which could have possibly helped the models predict better. A separate modified dataset was created after performing the required processing procedures. Multiple processing procedures were performed which are listed below,

- Handling missing values
  - Row deletion
  - Column deletion
- Handling class imbalance problem
  - Simple oversampling
  - Synthetic Minority Oversampling Technique (SMOTE)

2.1.1 Handling missing values

Every well-designed and controlled studies are prone to error and therefore missing data occurs in almost all research. Missing data can reduce the statistical power of a study and can produce biased estimates, leading to invalid conclusions. To prevent the same, different imputation (replacing missing data with substituted values) were pondered over of which two of the most promising methods were considered.
(Note: Irrespective of the processing performed, samples wherein the output label (i.e. resistant/susceptible) was unknown were removed from the dataset)

Row deletion: The method of imputation is as the title describes it to be. Delete all samples which have missing feature values in the training data. Although the imputation method reduces the number of samples available for training, the method allows us to capture true, experimentally generated information provided by every feature in the dataset.

Column deletion: On observing the dataset, it was found that all the missing values fell within three particular features, (SNP_CN_2714366_C967A_V323L_eis, SNP_I_2713795_C329T_inter_Rv24-15c_eis and SNP_I_2713872_C252A_inter_Rv2415c_eis) due to which the three features were removed as the presence of large number of missing values seemed to outweigh the information provided by the three features.

Other imputation techniques such as kNN and mode value were attempted but were not considered due to contradictions produced in the modified dataset, i.e. due to the imputation techniques redundant samples were generated and scenarios where samples with the same input features would have contradicting output labels would arise.

2.1.2 Handling class imbalance problem

class imbalance problem typically refers to a problem with classification where the classes are not represented equally. Eg: A binary classification problem with 100 instances (rows) where 80 instances are labeled with Class-1 and the remaining 20 instances are labeled with Class-2. This is an imbalanced dataset and the ratio of Class-1 to Class-2 instances is 80:20 or more concisely 4:1. To solve the problem, we chose two sampling methods as described below,

Simple oversampling: Copies of the samples of the minority class were made to prevent the class from being neglected while the model is trained. The point of choosing the simple oversampling technique was to check if a simpler solution would work better than a more complex solution.

Synthetic Minority Oversampling Technique (SMOTE):From [1], SMOTE selects samples that are close in the feature space, draws a line between the samples in the feature space and draws a new sample at a point along that line, i.e. a random sample from the minority class is first chosen. Then k of the nearest neighbors for that sample are found. A randomly selected neighbor is chosen and a synthetic example is created at a randomly selected point between the two samples in the feature space. The method works as it causes the classifier to build greater decision regions that contain nearby minority class points. A con of this approach is that the samples are created without considering the majority class which could result in contradictory samples if there is a strong overlap for the classes. Smote was applied using the python library imblearn.

Apart from the above mentioned techniques, weighted models were also tested over the dataset, which will be explained in detail in section 2.3.

2.2 Feature Sets Developed

A group of feature sets were reproduced taking into consideration data processing, as well as feature selection and feature extraction. The group of feature sets are described below,

- **row deletion**: A modified dataset made by applying row deletion mentioned in section 2.1.1.
- **column modified**: A modified dataset made by applying column deletion mentioned in section 2.1.1.

- **oversampling**: A modified dataset made by applying simple oversampling over the column modified dataset.
- **SMOTE**: A modified dataset made by applying SMOTE over the column modified dataset.
- **logistic PCA**: From [2], An extension over PCA logistic PCA is utilized to extend PCA to a binary dataset. What logistic regression is to linear regression, logistic PCA is to PCA. This is done so by projecting the natural parameters from the Bernoulli saturated model and minimizing the Bernoulli deviance. This is represented as

$$minimize_{W,C}\{\Sigma_{i=1}^{N}\Sigma_{j=1}^{D}log(1+exp(-x_{ij}.\Sigma_{k=1}^{K}W_ikC_jk))\}$$

  where, W = loading matrix, C = top K components, N=no. of samples and D = no. of features. In this feature set, a value of $K = 155$ was chosen. This decision was based over the visualization provided by the scree plot. 155 features accounted for ∼97% of the total variance found in the dataset. The dataset was created using the R library logistic PCA where the parameters of $K = 155$ and $m = 14$ was considered (the value of m is chosen for which the negative log likelihood is the least after applying the method cv.lpca over multiple values of m).

- **Chi-square**: Among several uses of the chi-square test, we used the chi-square test for feature selection. From [3] we defined our Null hypothesis as "the feature is independent (or does not provide sufficient information) for the output variable". Using the formula

$$\chi_c^2 = \frac{\Sigma(O_i - E_i)^2}{E_i}$$

  where, c = degree of freedom, O = observed values, E= expected values, χ = random variable , chi-square value of each of the input features is calculated and using the p-value, the calculated chi-square value is checked (whether it falls in the accepted or rejected region with the help of the chi-square table). Accordingly features were selected or rejected to obtain the final dataset.

2.3 Models Utilized

Various machine learning classifiers were explored, such as logistic regression with L2 regularization, support vector machines with radial basis function kernel, neural network and ensemble learning techniques such as extreme gradient boost, and stacking. For every drug, the problem is a single binary classification problem, i.e to predict if the MTB isolate is resistant or susceptible to that particular drug. The models were applied using popular libraries: sklearn, tensorflow and keras.

Logistic regression: Logistic regression is a statistical model which is used primarily for classification tasks. Logistic regression uses an equation similar to the equation of linear regression, however, it models the dependent (or output) variable as a binary (or categorical) value using a logistic or sigmoid function [4]. A L2 regularized logistic regression model has been used for the experiments.

Support Vector Machine: Support Vector Machines (or SVM) is a supervised machine learning algorithm which can be used for both classification and regression tasks. For our experiments, we will be using Support Vector Classification (or SVC). The intuitive idea behind SVM is to find the optimal decision boundary (or hyper-plane) that separates the two classes. A Gaussian kernel implementation using the radial basis function has been used. [5]

Neural network: Inside the human brain, multiple neurons are interconnected, where each neuron takes a sensory input and produces a response. Similarly in an artificial neural network, each input node is connected to numerous neurons in multiple hidden layers, which are in turn are connected to the output nodes in the output layers. The layers are fully connected, and each interconnection has a weight associated with it, which are learned during the training process. Back propagation and gradient descent algorithm are used to train the neural net. [6]

Extreme gradient boosting (XGBoost): It is an ensemble learning technique based on gradient boosted decision trees, where errors made by the older models are corrected by adding new models. In gradient boosting the models are generated to predict the errors of the previous models, and they are added sequentially to make the final prediction. This process of adding models continues until no further improvements are possible. In extreme gradient boosting, gradient descent algorithm is used to optimize (minimize) the loss function. [7]

Super Learner Ensemble (SLE): From [8] The super learner algorithm is based off of the stacking ensemble technique. It involves pre-defining a k-fold split of the training dataset, then evaluating all algorithms (models in the ensemble) and algorithm configurations on the same split of the data. All out-of-fold predictions are then used to train a meta-model that learns how to best combine the predictions. We stacked SVM and XGBoost as base models and used Logistic regression as the meta-model.

Each model was trained on the different processed datasets: row deletion, column modified, oversampling, SMOTE, logistic PCA, and Chi-square. The results have been studied and compared.

2.4 Prediction Accuracies

Listed in table 1 are the best two predictions for each drug. Each row lists the drug for which the prediction is performed, the model used for the prediction, the feature set used to train the model and the accuracy of each model wrt AUCROC metric.

3 OBSERVATION AND ANALYSIS

The following observations were made whilst testing the models trained over datasets previously mentioned,

- The models performed very well on the column modified dataset. The removal of columns containing almost all missing values helped improve the prediction accuracies.
- Out of all of the models utilised, XGBoost and SVM gave the best accuracy for many of the drugs across the different feature sets that we developed.
- The performance of the models on the logistic PCA and chi-squared datasets were not up to par with the column modified dataset. Due to this, we concluded that almost all of the features play an important role in the model's output predictions.
- The oversampling and SMOTE feature sets that we developed to tackle the class imbalance problem produced better results than the logistic PCA and chi-square feature sets but results were not better than the column modified feature set.

Drug	Model	Feature Set	Accuracies
RIF	XGBoost	SMOTE	0.98929
	XGBoost	column modified	0.98923
AMK	SLE	column modified	0.93199
	SVM	column modified	0.91949
INH	SLE	column modified	0.97451
	SVM	column modified	0.97285
PZA	SLE	column modified	0.89919
	SVM	column modified	0.89803
EMB	SVM	SMOTE	0.94120
	SVM(weighted)	column modified	0.93834
STR	XGBoost	column modified	0.95304
	SLE	column modified	0.94946
CAP	Neural Network	Logistic PCA	0.82711
	Logistic Regression	SMOTE	0.82312
KAN	XGBoost	column modified	0.92269
	SLE	column modified	0.90716
MOXI	SLE	column modified	0.96881
	XGBoost	column modified	0.96670
OFLX	XGBoost	column modified	0.89278
	SVM	SMOTE	0.85618

Table 1: Training Accuracy Of Different Models Used

- SLE produced acccuracies which were either better (by small margins) or close to the best accuracies of the individual models in the ensemble. The implications that this observation holds is considerable, as models which perform well for a particular drug can be stacked using the SLE algorithm with models that perform well on other drugs thus allowing us to produce an ensemble that predicts quite accurately irrespective the drug presented. Note that sometimes individual models outperformed the SLE model. This is due to the utilization of a different, better dataset for the individual model but not for the model in SLE.

4 MEMBER CONTRIBUTIONS

- **Debanjan Chatterjee**:
  Roll no.: 20111016
  Programme: MTech CSE

  - Developed Neural Network model for every drug.
  - Developed Logistic Regression model for every drug.
  - Developed XGBoost model for every drug.
  - Developed Random Forest, Naive Bayes, SVM with L2 regularization model for specific drugs.

- **Mayank Bansal**:
  Roll no.: 20111032

Programme: MTech CSE

  - Developed the column modified feature set.
  - Performed feature selection and developed the chi-square feature set.
  - Developed the oversampling feature set.
  - Developed SLE model for each drug.

- **Narein Rao**:
  Roll no.: 20111414
  Programme: MS(R) CSE

  - Developed row modified dataset.
  - Applied logistic PCA over column modified dataset.
  - Developed SMOTE dataset.
  - Developed SVM weighted and SVM models for every drug.

- **Harshvardhan Pratap Singh**:
  Roll no.: 20111410
  Programme: MS(R) CSE

  - Developed XGBoost model for every drug
  - Developed Neural network model for every drug
  - Developed SVM model for every drug.
  - Developed model for chi square, pca, smote and column modified datasets.

References

[1] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, Jun 2002.

[2] Andrew J. Landgraf and Yoonkyung Lee. Dimensionality reduction for binary data through the projection of natural parameters. *Journal of Multivariate Analysis*, 180:104668, Nov 2020.

[3] Sampath Kumar Gajawada. towardsdatascience, chi-square test for feature selection in machine learning, oct 2019.

[4] David G Kleinbaum, K Dietz, M Gail, Mitchel Klein, and Mitchell Klein. *Logistic regression*. Springer, 2002.

[5] Bernhard Scholkopf, Kah-Kay Sung, Christopher JC Burges, Federico Girosi, Partha Niyogi, Tomaso Poggio, and Vladimir Vapnik. Comparing support vector machines with gaussian kernels to radial basis function classifiers. *IEEE transactions on Signal Processing*, 45(11):2758–2765, 1997.

[6] Bayya Yegnanarayana. *Artificial neural networks*. PHI Learning Pvt. Ltd., 2009.

[7] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[8] Jason Brownlee. Machine learning mastery, how to develop super learner ensembles in python, dec 2019.