Binary Classification of Cell Communication in genes of E. coli bacterium

Lalith Veerabhadrappa Badiger
School of Information Technology and Electrical Engineering
The University of Queensland, Qld., 4072, Australia

Introduction

The primary aim of this project is to build a binary classifier using supervised machine learning algorithms that classifies if a gene of Escherichia coli or E.coli bacterium has the function "Cell communication" or not. E.coli is a common bacterium in the gut of humans and warm-blooded animals. Most E. coli strains have beneficial functions such as preventing the establishment of harmful bacteria in our intestine. However, there are pathogenic E. coli i.e., they can cause human illnesses like diarrhoea and even kidney failure. E. coli communicate by means of chemical signal molecules called autoinducers. This process, called "quorum sensing", allows bacteria to count the members in the community and to change gene expression in the population synchronously. Successful bacterial–host interactions, both symbiotic and pathogenic, are frequently dependent on quorum-sensing-controlled processes [1].

Data Specification

The data is given in CSV (Comma Separated Values) format with 1500 data points, one for each gene. The dataset has 116 features in total and one target column indicating whether the gene has the function "Cell communication" represented with binary labels 0 (Negative class) or 1 (Positive class). The first 103 columns describe the expression level of genes which are continuous (or numerical) values and the following 13 columns are nominal (or categorical) features describing gene functional.

Data Pre-processing

In the data pre-processing stage, the data is cleaned (handle outliers and missing values), transformed (standardization or normalization) and reduced (if required, using PCA). The first step of cleaning the data is to identify and rectify outliers (or anomalies) which are data points that are distant from all other observations. Having an outlier will adversely impact statistical analysis, thereby unfavourably affecting the correctness of the classification model. Here, I will explore five techniques to detect outliers namely, density-based, model-based, distance-based, cluster-based, and isolation-based techniques. The data point is considered as an outlier if:

- For density-based technique, the Local Outlier Factor (LOF) is significantly greater than 1.
- For model-based technique, an outlier is a point that is greater than 2 or more standard deviations away from the mean, assuming that the dataset is in Gaussian distribution.
- For distance-based technique, the greater their distance to their kth nearest neighbour, the higher is the chance that this point is an outlier.
- For cluster-based technique, if a point does not belong to any cluster, it is an outlier.
- For isolation-based technique, if a forest of random trees collectively produces shorter path lengths for some points, then these points are highly likely to be outliers.

The entire row for the detected outlier will be dropped.

The second step of cleaning the data is to handle missing values. For the numerical columns, the missing values are imputed by the average value of that column and for the nominal columns, the missing values are imputed by that column's most frequent value. For a particular row, if there are multiple missing values across columns', then that entire row will be dropped.

After data cleaning, the data is transformed (or scaled) since different features have different degrees of magnitudes. This non-uniformity in data ranges can adversely impact classification performance. Hence, the data is normalized by Max-min normalization if data does not follow a Gaussian distribution or Standardized by z-score normalization if data follows a Gaussian distribution.

The effective pre-processing techniques are then selected by using k-fold cross-validation. The given dataset will be first divided into the train (80% data), and test set (20% data). Then, cross-validation will be performed on a train set where it will be randomly divided into k-folders (k set to 3 or 5) with an almost equal number of instances in each folder for training and then validation set is chosen from one of those folders (1/3rd or 1/5th of data). This process is repeated for different combinations of training and validation sets. The best pre-processing technique which yields the highest average accuracy score for the given classification model will be chosen.

Classification

This project experiments with primarily four classification techniques (and their ensemble model) namely: decision tree, random forest, K-nearest neighbour and naïve Bayes classifier. The input to these supervised classification models will be the pre-processed data and the final output will be binary labels i.e., 0 (Negative class) or 1 (Positive class) representing cell communication in genes of E. coli bacterium.

The first classifier is a decision tree that follows recursive partitioning where data is continuously split with increasing purity until stopping criteria is met. The purity is measured by: Information gain (select the largest), Gain ratio (select the largest) and Gini index (select the lowest). The hyperparameters will be tuned using GridSearchCV which performs an exhaustive search over specified parameter values for an estimator with cross-validation and generates a model with the chosen best parameters. The parameters that could be tuned for decision tree are max_depth = [2, 3, 5, 10, 20], min_samples_leaf = [5, 10, 20, 50, 100] and criterion = ["gini", "entropy"]. Also, important features can be extracted using decision trees.

The second classifier is the random forest which is an ensemble of decision trees whose input is the random sample of k-features and output is based on majority voting. The hyperparameters of random forest are the same as the decision tree with an additional hyperparameter i.e., n-estimators = [10, 50, 100, 200] which are the number of trees in the forest.

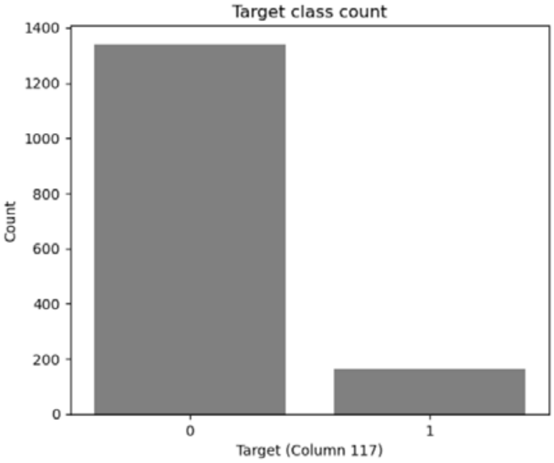
The third classifier is the K-nearest neighbour classifier which is based on lazy learning approach where training data is "memorized" and the labels of "K" nearest neighbours are used to determine the label of test example using a certain distance metric such as Manhattan, Euclidean and Chebyshev distance which are treated as hyperparameters here. "K" will be varied from 3 to 15 (odd numbers only) and cross-validation will be performed to find the model with the highest evaluation result averaged across cross-validation trials.

The fourth classifier is the naïve Bayes classifier which estimates the posterior distribution, for instance x and label y, as p(y|x) = [p(x|y)*p(y)] / p(x) that is also called as Bayes' theorem. If $p(y_1|x) > p(y_2|x)$, then y_1 is more likely to be the true label for x than y_2. Here, Laplacian correction of 1 will be used. This model assumes that all features are conditional independent from each other and the continuous values conditioned on y follow a univariate Gaussian distribution.

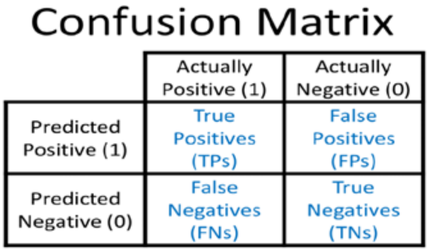
Finally, an ensemble classification model is constructed having the above four classifiers trained on the same dataset. The results of these classifiers are combined using statistical methods to give a prediction, for instance, based on majority voting. There are also advanced ensemble techniques like Gradient Boost, AdaBoost and XGBoost which will be explored here.

Model Evaluation

To create a good machine learning model, there should be a reasonable bias-variance trade-off so that the model is not too simple that it underfits the data resulting in poor classification or the model is not too complex that it overfits the data thereby losing the ability to generalize the results. To overcome this problem k-fold cross-validation is performed. The value of k is selected such that each train/test batch of data samples is statistically representative of the whole dataset. The performance of the model can be evaluated using the averaged evaluation metric across all the folds. Hence, cross-validation will be used for model selection as well as hyperparameter tuning.



By default, cross-validation uses accuracy as the evaluation metric. As seen from the above bar chart, the count of the negative class is significantly more than the count of the positive class. This is known as a class imbalance. If the dataset is imbalanced, accuracy may not be the most appropriate metric, instead, the F1 score will be used here.



Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

The above confusion matrix is created to compare the performance across different classification models in supervised learning. This test is performed on unseen test data.

Metrics that will be used include:

Recall (or sensitivity or true positive rate): In all positive samples, the rate of those predicted as “positive”.

$$\text{Recall} = TP / (TP + FN)$$

Precision: In all predicted as positive samples, the rate of those that are actually “positive”.

$$\text{Precision} = TP / (TP + FP)$$

F1 score: Harmonic mean of precision and recall.

$$\text{F1 Score} = \frac{1}{\frac{1}{2}\left(\frac{1}{Recall} + \frac{1}{Precision}\right)} = \frac{\text{TP}}{\text{TP} + \frac{1}{2}(\text{FP} + \text{FN})}$$

Timeline

- The first milestone will be completing data pre-processing by the 30th of September, 2021.
- The second milestone will be implementing all the classification and ensemble models by the 4th of October, 2021.
- The third milestone will be model selection, evaluation and completion of the report by the 25th of October, 2021.

References

[1] K. B. Xavier, and, B. L. Bassler, “Interference with AI-2-mediated bacterial cell-cell communication,” *Nature*, vol. 437, pp. 750–753, Sept. 2005

Bibliography

This project proposal gave me a formal and structured understanding of the life cycle of a data science project. I will use this opportunity to gain practical exposure and improve my analytical capabilities. This proposal enabled me to do a comprehensive literature review of data mining concepts from various journals, articles and technical blogs in addition to the course provided lecture materials. I gained an appreciation for the diverse applications of data mining techniques in various domains and the impact they could potentially create.