INFS7203 Data Mining Project Proposal

Chen CHEN 
School of Information Technology and Electrical Engineering
The University of Queensland, Qld., 4072, Australia

1 Data Pre-processing

The analysis of the data shows that this is an imbalanced with some extreme values and missing values (NAN). Each column has 80 to 131 missing values which is less than 10% of the total number of data points. However, in total, there are 1176 rows with missing values. Removal of those values will not only significantly shrink the size of the dataset but also can possibly remove the correct and useful samples. Thus, instead of removing, imputation will be performed. Considering the potential impact of those extreme values in numerical features (the first 103 columns), class-specific median would be a better way to impute the missing values compare to mean in this case. For the nominal features (from Column 104 to Column 106), the missing values will be imputed by the most common class-specific feature value [1].

After imputation, outlier detection and treatment will be conducted. The lecture notes [2] demonstrated that: both density-based and cluster-based outlier detection techniques are and computational intensive and can come with the curse of dimensionality; the model-based methods might be difficult to applied to multiple correlated features; the distance-based method have difficulties in detecting outliers when there is a variation in density. In order to correctly detect outliers when variation in density exists and to reduce computation cost, isolation forest will be selected to detect outliers.

Outliers can be handled in three ways: remove them, replace them with other values and keep them. My lack of domain knowledge about the data makes it difficult to decide which way to choose. Hence, I will try to train each classifier with these three different outlier-handling approaches to figure out which way leads to a better result. Regarding full removal of outliers, the missing value should be re-imputed using the remain data. For replacement, studies conducted by Maniruzzaman *et al.* [3] and Roy *et al.* [4] state that outlier imputation using group/class-specific median outperforms imputation with mean and other KNN and iterative techniques. Thus the numerical outliers will be replaced by class-specific median, and the nominal ones will be imputed by the most common values in corresponding class.

Moreover, data normalization needs to be performed, since the magnitude of some feature values are different. For example, features values in column 17 and 24 have much smaller magnitude compared to other features. Same as handling the outliers, two type of normalization, namely max-min normalization and standardization will be treated as hyperparameters which will be tuned later using cross validation in combination with different classification methods.

2 Classifier Construction

Firstly, the whole data set should be split into training set and test set in 75% and 25% respectively. Because the dataset is imbalanced, I would perform a stratified train-test-split to maintain the imbalance so that the test and train dataset have the same distribution on each class [5]. Then the training set will be split into stratified 10-fold to perform the following model training and cross validation (CV).

Secondly, Naïve Bayes, k-Nearest-Neighbour (KNN), Decision Tree and Random Forest will be used to construct classifiers.

Naïve Bayes. Classifiers will be built using Naïve Bayes combing three different outlier treatment methods and two types of normalization respectively. For instance, the construction and cross-validation process of a Naïve Bayes classifier using max-min normalization with removal of all the outliers will be as follows: The classifier will first be trained using 9 folds (fold-1 to fold-9) and then be validated using 1 fold (fold-10) to obtain the F1 score. Then this process will repeat 10 times using different combination of folds but the same mechanism. Finally, the mean and standard deviation of the F1 scores in these ten trials will be obtained.

K-Nearest-Neighbour. Similar to the previous classifiers, more classifiers will be constructed using KNN algorithm with different combinations of outlier treatment methods and normalization methods, because KNN is distance-based. In addition, hyper-parameters k and the different distance metrics need to be tuned using CV. There are three kinds of distance metrics can be chosen from: *Manhattan distance*, *Euclidean distance* and *Chebyshev distance* [6]. For the value k, in convention, it could be the square root of the data size, which is around 38 in this case [7]. In order to be more accurate and optimise the classifier mostly, [5, 10, 20, 30, 40] will be examined first. Then base on the F1 score and learning curve, the range will be narrowed down step by step with a smaller interval. The computational

cost expense could be reduced using Randomised Search CV [8]. The hyperparameter combination with the highest mean F1 score and lowest standard deviation will be selected for further classification.

Decision Tree. Since decision tree is not distance-based, there is no need to perform normalization while using decision tree. Thus, three outlier treatment methods will be combined with the following hyperparameters and tuned together. Firstly, the split criterion, namely *Information Gain*, *Gain Ratio* and *Gini Index*. Due to the imbalance of the dataset, it is more likely to get overfitted to the class with majority amount of data. To address this issue, pruning of the decision tree should be needed. Pruning could be done by tuning the hyperparameter *max_depth* in sklearn [9]. The starting range of *max_depth* will be [10, 40, 70, 100]. Then the combination of the outlier treatment method, criterion and *max_depth* will be trained using CV.

Random Forest. Similarly, normalization is not necessary for random forest. The hyperparameter needed to tune here are the number of decision tree constructed (*n_estimator*), the number of feature sampled (*max_feature*), split criterion and the depth of the tree (*max_depth*) [10]. With regard to the hyperparameter tuning on decision and random forest, the physiology behind is the same as KNN. Aurélien Géron [11] stated: "When you have no idea what value a hyperparameter should have, a simple approach is to try out consecutive powers of 10 (or a smaller number if you want a more fine-grained search. The starting range for *n_estimator* will be [10, 40, 70, 100] and for *max_feature* will be [3, 10, 40, 80, 102]. Then the hyperparameters will be fine-grained by observing and comparing the learning outcome of each step.

3 Model Selection and Evaluation

Since the data is imbalanced, F1 score will be used to evaluate the performance of the classifiers instead of accuracy [12]. When the construction and hyperparameter-tuning are done, each classifier will have the mean and the standard deviation of the F1 score. Generally, a classifier with highest mean and lowest standard deviation will be selected. Thus, four classifiers using different algorithms will stand out, and the fusion mode and all hyperparameters are finalized for each of them.

Before determined the ultimate classifier, ensemble of these classifiers should be examined. To start with, concatenate all classifier results by combining them with majority vote. Next, compare the concatenated result to find out the ensemble combination leads to the best mean and standard deviation of F1 score. Lastly, compare the training result of the ensemble with all the classifier using only one algorithm and select the one with better performance.

Before applying this selected classifier to the test set, two important steps need to be done.

1. Retrain the classifier on the whole training set with all labelled data [6].

2. Pre-process the test data using the same outlier handling and normalization method that revealed in model selection.

Then the classifier is ready to be examined using the test data.

4 Timeline for Phase 2 Implementation

The implementation plan will be as follows:

1. By the end of Mid-break: Explore and read more about the relevant packages that I will need in coding phase (numpy, pandas and ski-learn etc.) and get familiar with them

2. In Week 10: Finish data pre-processing

3. In Week 11: Finish classifier construction and hyperparameter tuning.

4. In Week 12: Finish model selection, evaluation and the project report.

References

[1] M. Xu, "Lecture 2: Introduction to Classification." University of Queensland, Data Mining (INFS4203/7203), 2022.

[2] M. Xu, "Lecture 7: Anomaly Detection." University of Queensland, Data Mining (INFS4203/7203), 2022.

[3] Md. Maniruzzaman *et al.*, "Accurate Diabetes Risk Stratification Using Machine Learning: Role of Missing Value and Outliers," *J. Med. Syst.*, vol. 42, no. 5, p. 92, Apr. 2018, doi: 10.1007/s10916-018-0940-7.

[4] K. Roy *et al.*, "An Enhanced Machine Learning Framework for Type 2 Diabetes Classification Using Imbalanced Data with Missing Values," *Complexity*, vol. 2021, p. 9953314, Jul. 2021, doi: 10.1155/2021/9953314.

[5] scikit-learn, "sklearn.model_selection.train_test_split," scikit-learn.org. https://scikit-learn/stable/modules/generated/sklearn.model_selection.train_test_split.html (accessed Sep. 16, 2022).

[6] M. Xu, "Lecture 4: k-Nearest Neighbors and Naïve Bayes." University of Queensland, Data Mining (INFS4203/7203), 2022.

[7] P. Nadkarni, "Chapter 10 - Core Technologies: Data Mining and 'Big Data,'" in *Clinical Research Computing*, P. Nadkarni, Ed. Academic Press, 2016, pp. 187–204. doi: 10.1016/B978-0-12-803130-8.00010-5.

[8] scikit-learn, "sklearn.model_selection.RandomizedSearchCV," scikit-learn.org.

https://scikit-learn/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html (accessed Sep. 16, 2022).
[9] scikit-learn, "sklearn.tree.DecisionTree Classifier," scikit-learn.org. https://scikit-learn/stable/modules/generated/sklearn.tree. DecisionTreeClassifier.html (accessed Sep. 16, 2022).
[10] scikit-learn, "sklearn.ensemble.RandomForest Classifier," scikit-learn.org. https://scikit-learn/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (accessed Sep. 16, 2022).
[11] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, Inc., 2019.
[12] P. Nair and I. Kashyap, "Hybrid Pre-processing Technique for Handling Imbalanced Data and Detecting Outliers for KNN Classifier," in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, Feb. 2019, pp. 460–464. doi: 10.1109/COMITCon.2019.8862250.