Introduction to Focus Area Project-1

Gitlab: https://git.imp.fu-berlin.de/kunaphak91/ifabi-2019

Name: 1. Minie Jung

2. Kunaphas Kongkitimanon

3. Chang Wan-Ju

The goal of the project

Based on the available heart disease dataset from UCI machine learning[2], we develop models by training five different classifiers to predict the presence of heart disease in the patient. We also compare and analyze the results of heart disease prediction. Moreover, we conduct further analysis to see the advantages and disadvantages of using various data processing methods and more datasets. In addition, cooperating with preprocessing can aid to improve the procession of classifiers.

Scientific background

Machine learning methods

Multinomial Logistic regression

Multinomial logistic regression is a classification method that generalizes logistic regression for solving multiclass problems (more than two levels)[3]. A linear combination is used along with observed features and model parameters to determine the probability of each distinct value of the dependent variable.

SVM (Support vector machines)

SVM constructs vectors/hyperplanes to separate data. To improve the accuracy of prediction, this method tries to find a vector/plane that has the maximum distance between the data of classes. A new example is predicted based on the space where it is mapped.

KNN(K-Nearest Neighbor)

KNN algorithm assumes that similar things are near to each other. The training set is vectors in multidimensional space. KNN algorithm selects k-closest vectors based on calculating distance between two vectors and predicts label to the most frequency label of k-closest training vectors.

Decision Tree

A decision tree is a tree structure, which branch represents decision rule, an internal node represents features, and a leaf node represents the result. It classifies the data by using the Attribute Selection Measures. After selection measures, each feature gets a score. Each time the data partitioned, the decision tree selects the feature with the best score as a splitting feature.

Random Forest Algorithms

Based on decision tree algorithms, a random forest algorithm builds a multiple depth decision

trees to train data and gets averaging to minimize the loss function. However, it splits features depend on random cut rather than choosing the best cut point.

Description of the Data

This database contains 76 attributes, but all published experiments refer to using a subset of 14 of them.

Attribute Information:

1. age
2. sex
3. chest pain type (4 values)
4. resting blood pressure
5. serum cholesterol in mg/dl
6. fasting blood sugar > 120 mg/dl
7. resting electrocardiographic results (values 0,1,2)
8. maximum heart rate achieved
9. exercise-induced angina
10. oldpeak = ST depression induced by exercise relative to rest
11. the slope of the peak exercise ST segment
12. number of major vessels (0-3) colored by fluoroscopy
13. thal: 3 = normal; 6 = fixed defect; 7 = reversible defect
14. target: integer valued from 0 (no presence) to 4

Methodology

1. We use two datasets to feed and compare our model.
    a. processed.cleveland.data
    b. processed.merged.data; This data merged from 4 different data sources, which are processed.cleveland.data, processed.hungarian.data, processed.switzerland.data, and processed.va.data.
2. Data preprocessing
    a. Taking care of missing data
    b. Encode enum-like features
    c. Detect and remove outliers
    d. Normalize features
    e. Split data into feature and label set
3. Construct five machine learning models
4. Evaluate and Compare the result
    a. List each prediction scores
    b. Compare results between processed.merged.data and processed.cleveland.data.

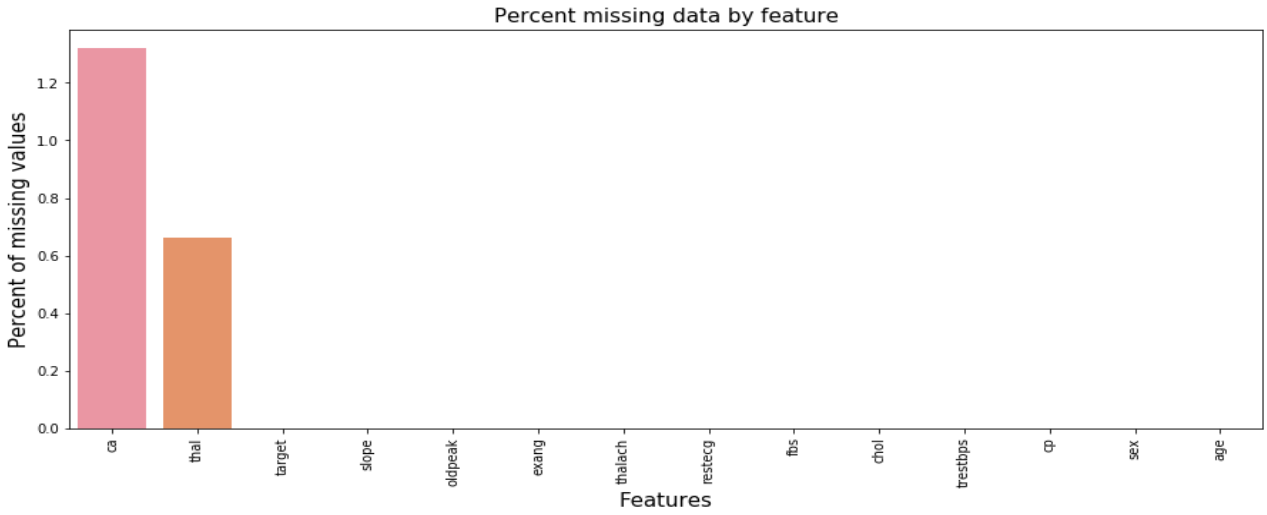
Implementation

We use Scikit learn[1] library to help our implementation.

Data preprocessing

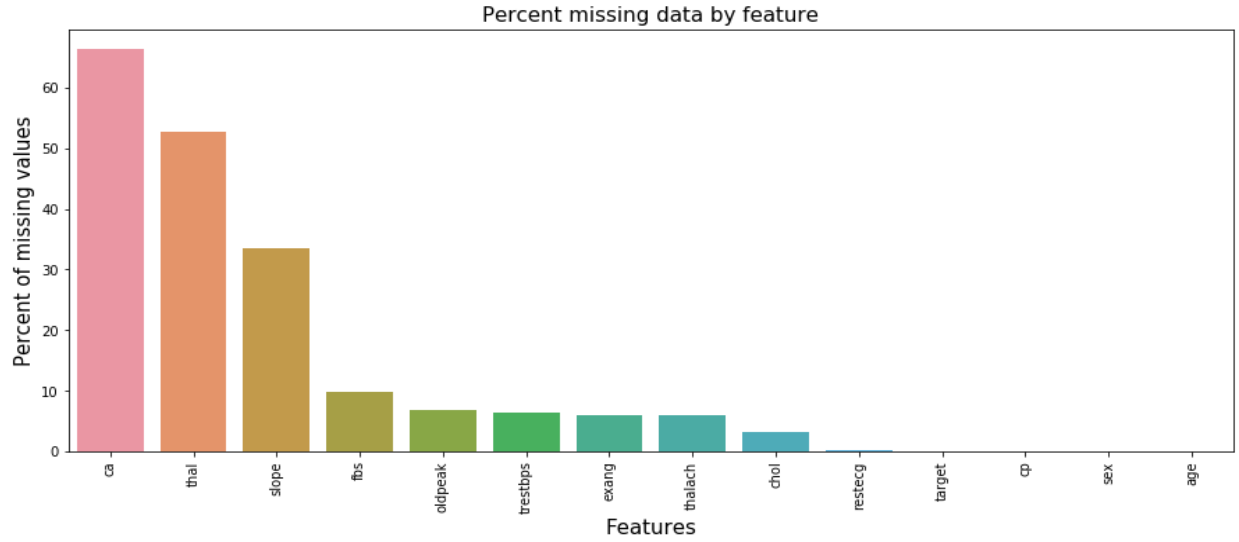
First, we try to identify missing values in a dataset.

For **processed.cleveland.data**



The cleveland dataset contains a few missing value in “ca” and “thal” columns.

For **processed.merged.data**



However, when we did merging data sources, the volume of missing value is significantly raised.

To handle missing value, We apply a Univariate feature imputation[4] to replace missing data with calculated value from the function. In addition, the machine might be considered the enum-like data as continuous data. So we convert the data like [0, 1] → [[1, 0], [0, 1]], also we use the Elliptic Envelope method[5] to detect outliers. This model fits multivariate gaussian distribution to the dataset. It makes ellipsoid shape of data and define outliers which are far from shape. Some features have values on large scale and some are not. After normalization, we can make the feature value to be centered to 0.

Finally we split data into two parts(the data to fit and the target variable to try to predict)to prepare an input for building machine learning model and evaluation.

Classifiers

Multinomial Logistic regression

We set multi_class to “multinomial” for multiclass problems and also set optimization algorithm to “newton-cg” to handle multinomial loss.

SVM (Support vector machines)

There are some different types of SVM-Kernel and we chose to build a support vector machine with RBF(Radial Basis Function) kernel. RBF are especially useful when the data-points are not linearly separable.

K-Nearest Neighbor

After several tests with various k values, we decided k value 3 because we got the best result when k is 3 in this case.

Decision Tree

We train the decision tree through entropy method to measure. Meanwhile, we set the depth limit which prevents decision tree from overfit and get the best result in value 4.

Random Forest Algorithms

Based on it’s feature, we didn’t need to set the limit to prevent it overfitting, so we didn’t adjust the parameters in this algorithm. The function to measure the quality of a split is “gini” impurity.

For testing, the machine learning models along with the features and label dataset were evaluated by K-Folds cross-validation function with 10 splits and shuffling data. The result from cross-validation function returns array of scores of the estimator for each run of the cross validation. Then we use mean of each score to represent a performance for each model.

Some screenshots from final result when using processed cleveland data.

Multinomial Logistic Regression

```
[ ] print_accuracy_report(classifier_logis, X, y, num_validations=kf )

Accuracy: 0.60 (+/- 0.15)
F1: : 0.57 (+/- 0.15)
Precision: : 0.56 (+/- 0.24)
Recall: : 0.58 (+/- 0.22)
```

SVM

```
[ ] print_accuracy_report(classifier_svm, X, y, num_validations=kf )

Accuracy: 0.60 (+/- 0.16)
F1: : 0.53 (+/- 0.14)
Precision: : 0.50 (+/- 0.18)
Recall: : 0.62 (+/- 0.12)
```

Decision Tree

```
[ ] print_accuracy_report(clf_decision_tree, X, y, num_validations=kf )

Accuracy: 0.55 (+/- 0.17)
F1: : 0.51 (+/- 0.26)
Precision: : 0.54 (+/- 0.20)
Recall: : 0.58 (+/- 0.21)
```

KNN

```
[ ] print_accuracy_report(clf_neigh, X, y, num_validations=kf )

Accuracy: 0.59 (+/- 0.15)
F1: : 0.57 (+/- 0.23)
Precision: : 0.56 (+/- 0.18)
Recall: : 0.58 (+/- 0.09)
```

Random Forest

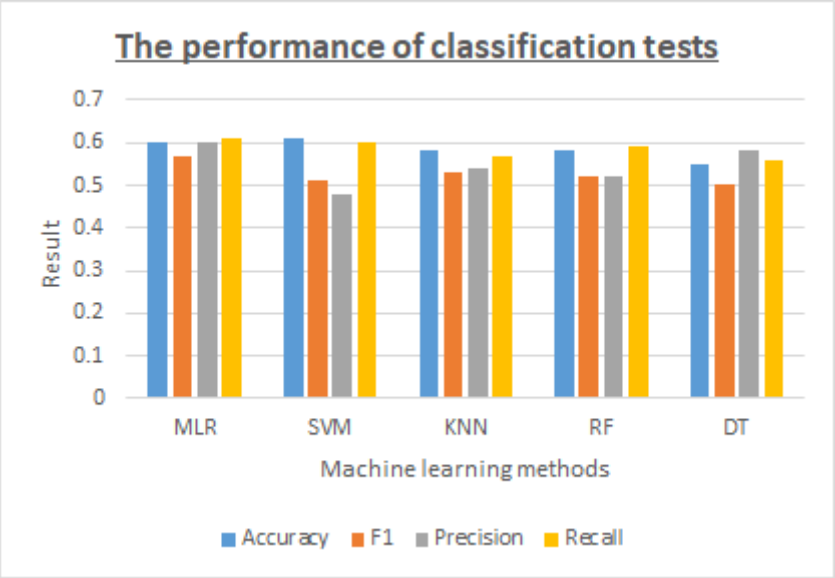
```
[ ] print_accuracy_report(clf_random_forest, X, y, num_validations=kf )

Accuracy: 0.58 (+/- 0.16)
F1: : 0.54 (+/- 0.17)
Precision: : 0.53 (+/- 0.23)
Recall: : 0.57 (+/- 0.19)
```

Comparison of the used methods

	Strengths	*Weakness*
Logistic Regression	•Have nice probabilistic interpretation for outputs •Can be regularized to avoid overfitting	•When having multiple/non-linear decision boundaries, logistic regression tends to underperform •Not flexible enough
Support Vector Machines	• Can model non-linear decision boundaries •Fairly robust against overfitting(especially in high-dimensional space) • Work well in continuous value inputs	• SVM's are memory intensive, trickier to tune due to the importance of picking the right kernel •Don't scale well in larger datasets
K-Nearest Neighbor	• Work well in continuous value inputs • No Training Period • New data can be added seamlessly • Few parameters need to adjust	•Not work well with large dataset and high dimensions •If we don't do features scaling, the prediction will be wrong.
Decision Trees	•Interpretability (easy to explain) •Requires less effort for data pre-processing compared with other algorithms •Missing values don't affect the tree building	•Little change will affect tree structure easily (unstable) •Calculate is more complexity than other algorithms •Training the model need higher time
Random Forest	•Be able to handle thousands of input variables without variable deletion. •Provide a reliable feature importance estimate	•Difficult for humans to interpret. •If the data contain groups of correlated features of similar relevance for the output, smaller groups are favored over larger groups.

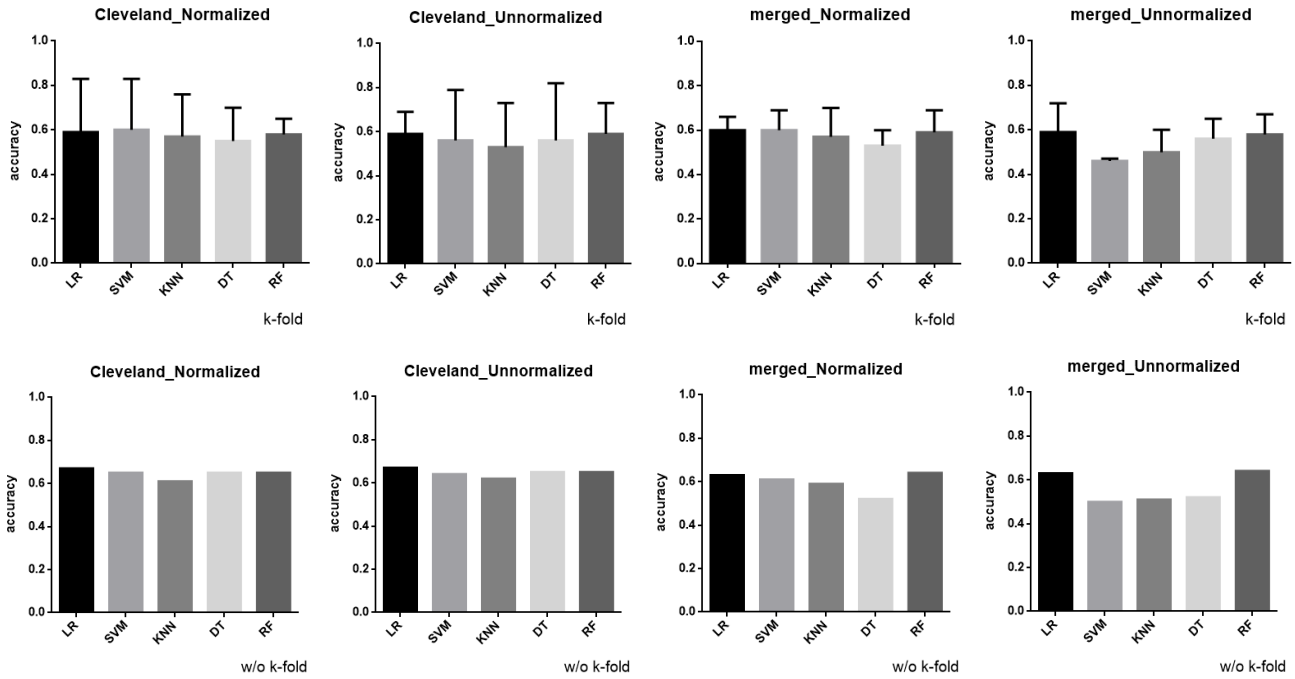
Evaluation Results



It is better to look at both the precision and recall scores, so we consider the F1 score as a weighted average of precision and recall. Overall, The best performance among tested algorithms is in MLR (Multiple Linear Regression).

Through code testing, We found several parameters that might affect accuracy. Therefore, we want to observe the difference by testing those parameters separately (normalized data, unnormalized data, with k-fold, without k-fold).

		with k-fold		w/o k-fold	
		with normalization	without normalization	with normalization	without normalization
cleveland data	Logistic Regression	0.59 (+/- 0.24)	0.59 (+/- 0.10)	0.67	0.67
	SVM	0.6 (+/- 0.23)	0.56 (+/- 0.23)	0.65	0.64
	K-Nearest Neighbor	0.57 (+/- 0.19)	0.53 (+/- 0.20)	0.61	0.62
	Decision Tree	0.55 (+/- 0.15)	0.56 (+/- 0.26)	0.65	0.65
	Random Forest	0.58 (+/- 0.07)	0.59 (+/- 0.14)	0.65	0.65
merged data	Logistic Regression	0.6 (+/- 0.06)	0.59 (+/- 0.13)	0.63	0.63
	SVM	0.6 (+/- 0.09)	0.46 (+/- 0.011)	0.61	0.5
	K-Nearest Neighbor	0.57 (+/- 0.13)	0.5 (+/- 0.10)	0.59	0.51
	Decision Tree	0.53 (+/- 0.07)	0.56 (+/- 0.09)	0.52	0.52
	Random Forest	0.59 (+/- 0.10)	0.58 (+/- 0.09)	0.64	0.64



In most classifiers, it does not have significant difference except K-NN. We observed that KNN's performance is improved after normalized with k-fold. However, the group of cleveland without k-fold seems not different after normalized.

Discussion

Why is this a typical project for a data-scientist?

In our opinion, data scientist is a group of people who can develop available tools by different methods(mathematics, computers, statistics) to obtain meaningful information from various data.

Through this project, we learn how to utilize machine learning to develop a tool and apply it in the medical field. Besides, learning how algorithms operate in the progress of disease prediction. We also observe which parameters may cause influences in the models. Moreover, we can enhance our models by using other techniques to assist us in achieving better model performance, such as hyper-parameter tuning, which is attempting to adjust kernel or parameters automatically to reach a better score or avoiding dummy variable trap. We may also select other machine learning methods, for example, ordinal logistic regression, to find an algorithm that matches our datasets.

Since it corresponds with the goal in data science, we consider that it belonged to one of the typical projects for data scientists.

References

1. [scikitlearn] https://scikit-learn.org/stable/
2. [UCI dataset]https://archive.ics.uci.edu/ml/datasets/Heart+Disease
3. [Multinomial]https://en.wikipedia.org/wiki/Multinomial_logistic_regression
4. [Impute]https://scikit-learn.org/stable/modules/impute.html
5. [Elliptic Envelope]https://scikit-learn.org/stable/modules/outlier_detection.html