APPLIED AI MODULE COMP534

ASSIGNMENT 1

Student's name	Huy Pham	Saeth Wannasuphoprasit
ID	201534475	201585689

Date: March 15th 2020

I. Introduction

1. The Used Libraries

Table 1. The used libraries and their purposes

Library’s name	Used Purpose
pandas, Series, DataFrame	load and read data
matplotlib.pyplot, seaborn	visualize data
numpy	calculate with array
warnings	ignore warnings
from mlxtend.preprocessing import minmax_scaling	data scaling into [0, 1]
from sklearn.model_selection import train_test_split	split data
from sklearn.model_selection import cross_val_score	cross validation
from sklearn.model_selection import ParameterGrid	list all hyperparameters
from sklearn.tree import DecisionTreeClassifier	import classifier
from sklearn.naive_bayes import GaussianNB	import classifier
from sklearn.ensemble import RandomForestClassifier	import classifier
from sklearn.metrics import confusion_matrix	calculate confusion matrix
from sklearn import metrics	calculate accuracy
import timeit	find running time
from sklearn.feature_selection import mutual_info_classif	calculate Mutual Information score

2. Classification method and parameters

Table 2. Name of the used classifier and their list of hyperparameter

Hyperparameter	criterion	["gini","entropy"]	**Decision Tree**
	max_depth	[None, 2,3,4,5,6,7,8,9,10,11,12,13,14,15]	
	splitter	['best', 'random']	
	max_features	[None, 'auto', 'sqrt', 'log2']	
	priors	[None]	**Naïve Bayes**
	var_smoothing	list(np.logspace(0, -9, num= 100))	
	max_features	['auto', 'sqrt', 'log2']	**Random Forest**
	n_estimators	[2,5,10,25,50,75,100,150,200,250,300,400,500]	
	criterion	['gini', 'entropy']	

max_depth	[5,10,15,20,25,30]
n_jobs	[-1]

3. The training and testing process

3.1. Data cleaning, scaling, feature selection, and train-test splitting.

The dataset contains zeros, which are logically impossible for glucose level, blood pressure and body mass index (BMI). For the SkinThickness and the DiabetesPedigreeFunction feature, the valid values must be higher or equal to 10 mm and less or equal to 1, respectively. All invalid values were dropped while cleaning data. After cleaning, the total length of the dataset is 487. Mutual Information score (MIs) was applied to select features. The top 4 features with the highest MIs (DiabetesPedigreeFunction, BMI, Glucose, and Insulin) were chosen to train and test the model with the 0 to 1 scaling. Then, the processed data was spitted into 80:20 (train:test) ratio.

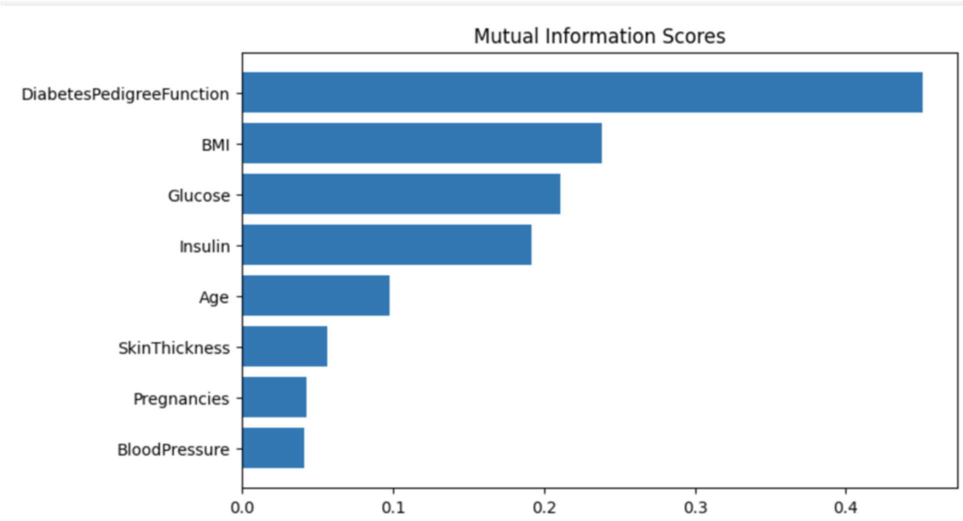

Figure 1. Mutual Information scores of each feature.

3.2. Training

All possible hyperparameters of each algorithm were trained from the processed training data mentioned above using five folds cross-validation.

3.3. Testing

For each algorithm, all trained models were tested on the processed test data and the top ten models with the highest test and training accuracy were shown with the confusion matrix of the best model.

II. Results

1. Best hyperparameters

The combination of each hyperparameter produced different train and test accuracy. The purpose of this project was to find the combination that resulted in the highest test accuracy. The table below shows the train and test accuracy of a particular classifier using different combinations of hyperparameters. For the Decision Tree classifier, the compound of hyperparameters of criterion: “gini”, max_depth: 5.0, max_feauters: “sqrt” and splitter: “random” produced the highest test accuracy 0.765306. The three combinations in the Naïve Bayes model shared the same train and test accuracy, 0.665834 and 0.75512, respectively. In the Random Forest model, the combination of criterion: “gini”, max_depth: 5.0, max_feauters: “log2” and n_estimators: 25 achieved the highest test accuracy (0.775510).

Table 3. Top three hyperparameters with highest train and test accuracy of each classifier

Hyperparameter						Classifier
criterion	max_depth	max_features	splitter	train_acc	test_acc	Decision Tree
gini	5.0	sqrt	random	0.668365	0.765306	
entropy	5.0	sqrt	random	0.663203	0.765306	
entropy	5.0	log2	random	0.668398	0.755102	
priors	var_smoothing	train_acc	test_acc			Naïve Bayes
None	0.000187	0.665834	0.755102			
None	0.000152	0.665834	0.755102			
None	0.000123	0.665834	0.755102			
criterion	max_depth	max_features	n_estimators	train_acc	test_acc	Random Forest
gini	5.0	log2	25	0.652947	0.775510	
gini	3.0	sqrt	5	0.678655	0.765306	
entropy	5.0	auto	50	0.676057	0.755102	

2. Confusion matrix

Table 4. Confusion matrix of the best version of each classifier

		Decision Tree		Naïve Bayes		Random Forest	
		Predicted Values		Predicted Values		Predicted Values	
		False	True	False	True	False	True
Actual Values	False	71	1	70	2	70	2
	True	22	4	22	4	20	6

Table 5. Results of each classifier

	Decision Tree	Naïve Bayes	Random Forest
Precision	0.8	0.67	0.75
Recall	0.18	0.18	0.23
Specificity	0.99	0.97	0.97
F1-score	0.29	0.28	0.35
Train accuracy	0.66836	0.66583	0.65295
Test accuracy	0.76530	0.75510	0.77551
Average run time (sec)	0.015	0.014	0.883

3. Conclusion

Overall, the three classifiers had high Precision and Specificity but low Recall. The Decision Tree produced the highest value in Precision and Specificity with 0.8 and 0.99, respectively. Both Random Forest and Naïve Bayes classifiers had the same specificity value (0.97). However, the Precision score of the Random Forest (0.75) model was higher than Naïve Bayes

(0.67). The three models are useful in predicting positive cases because of high Precision. In other words, when the models predict the patients having Diabetes, the chance of them truly having the disease is extremely high. However, due to low Recall, the number of false negatives is much higher than true positives. In other words, the three models are not trustful in predicting negative cases. If the patients are predicted as not having Diabetes, the chance of them truly negative is very low. By definition, F1-score is the harmonic mean of precision and recall. F1-score has a range from 0 to 1. The closer it comes to 1, the better the model is. Unfortunately, in this scenario, the three F1-score were lower than 0.5, in other words, they were not good models. In terms of test accuracy, random forest had slightly higher accuracy than decision tree because it consisted of many trees (ensemble models). The accuracy of naive bayes was the lowest. This maybe because of the fact that the model was based on statistical model which had random variation of the outcomes. In general, there were no significant differences in the train and test accuracy among the three classifiers. The Naïve Bayes model finished the training process with an average time per iteration was 0.014 sec, slightly lower than Decision Tree (0.015 sec) and significantly lower than Random Forest (0.883 sec). In conclusion, among the three classifiers, the Decision Tree was considered as the best model. Because it had the highest results in Precision and Specificity and trained the data fast.

III. Final Conclusion

1. Challenges of the project

There were two biggest challenges of the project. First, the dataset contained many invalid values. For instance, the SkinThickness feature had more than 200 impossibly logical values, which are less than 10 mm. Since the dataset consists of a small number of observations (768) but a high number of invalid values of 281 cases, it was difficult to decide whether to delete or replace them with the mean. Moreover, picking the right features for an Machine Learning (ML) is one of the most important steps to take, even important than the ML model itself. Thus, we used the Mutual Information (MI) score to select features. The MI score is in between 0 and 1. The higher the value, the more closely this feature and the target are linked. Second, each classifier has many hyperparameters to tune. For example, the Decision Tree had 4 hyperparameters and each hyperparameter had at least 2 sub-hyperparameter to tune. Therefore, it takes a lot of time to try and find the best combination of hyperparameters. However, this problem can be solved by using the GridSearchCV of the Sklearn library. The tool allows us to try as many combinations as we want in a single run. Moreover, it provides the train and test accuracy of all events, thus, we can compare easily compare the results and find the optimal compound.

2. Task allocation

Task	Person in charge
Data Cleaning	Huy and Saeth
Feature Selecting	Saeth
Data training and testing	Huy and Saeth
Hyperparameter tuning	Saeth
Confusion matrix finding	Saeth
F1, Precision, Recall and Specificity finding	Huy
Report writing	Huy