Spring 2023: CSCI 4588/5588

Programming Assignment #2

Name: Aviral Kandel ID: 2630609

1. Describe each classifier in 5 to 10 sentences and provide an appropriate reference.
   a. Decision Tree Classifier

      Decision Tree classifier are supervised machine learning technique that use prelabelled data to train an algorithm and make a prediction. It works like a flowchart. Each node comes out of a decision and each of the decisions can turn into decision nodes. It is also called a “white box” algorithm because we can understand the decision-making of the algorithm. Decision trees work by splitting data into a series of binary decisions. These decisions allow you to traverse down the tree based on these decisions. You continue moving through the decisions until you end at a leaf node, which will return the predicted classification. [1]

   b. Support Vector Machine

      SVM algorithm finds a hyperplane in an N-dimensional space that classifies the data points. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. Its memory efficient as it uses a subset of training points in the decision function called support vectors. Different kernel functions can be specified for the decision functions and its possible to specify custom kernels.

   c. Naïve Bayes Classifier

      A Naïve Bayes Classifier is a probabilistic machine learning model that is used for classification task. It is based on Bayes' theorem, which states that the probability of a hypothesis (such as a class label) given some observed evidence (such as a set of features) is proportional to the probability of the evidence given the hypothesis times the prior probability of the hypothesis. In the case of a binary classification problem with classes 0 and 1, the goal is to predict the probability of class 1 given some input features. For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter. A naive Bayes classifier considers each of these features to contribute

independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness, and diameter features. [2]

d. K nearest Neighbors Classifier: K-Nearest Neighbors (KNN) is a non-parametric machine learning algorithm used for classification and regression problems. Given a new input sample, KNN classifies it based on the class labels of the K-nearest training samples in the feature space. The value of K, which represents the number of nearest neighbors to consider, is a hyperparameter that can be chosen by the user. KNN is a simple and effective algorithm that can be used for both binary and multi-class classification problems. One drawback of KNN is that it can be computationally expensive for large datasets, since it requires calculating the distance between the new input sample and all the training samples. However, there are several optimizations and algorithms, such as the k-d tree, that can be used to speed up the KNN algorithm for large datasets.

e. Random Forest Classifier:
Random Forest Classifier is a popular machine learning algorithm that builds a set of decision trees and combines their results to make predictions. The algorithm starts by randomly selecting a subset of features and a subset of observations from the training dataset. It then builds a decision tree using the selected features and observations [3]. This process is repeated multiple times, and a set of decision trees is built. During the prediction phase, each decision tree in the forest makes a prediction, and the final output is the class that receives the most votes from the individual trees.

2. Describe the parameters/hyper-parameters that you have chosen to train the classifiers.
   a. Decision Tree classifier: Default
   b. Support Vector machine
      - Kernel: Linear Kernel.It is less prone to overfitting.
      - Gamma: 1. It denotes how far the influence of a single training example reaches.
   c. Naïve Bayes Classifier: Default
   d. K nearest Neighbors Classifier
      - N_neighbors = 10

   e. Random Forest Classifier
      - N_estimator: 100 (It specifies the number of decision trees that will be included in the random forest ensemble.

3. Define and describe the following terms (provide an appropriate reference(s)) for measuring the performances of a classifier:

a. True Positive rate: TP rate is a metric that measures the percentage of actual positives that are accurately identified. It is the case when the actual class of the data point was 1(True) and the predicted is also 1(True).
$TPR = \frac{TP}{TP+FN}$ where, TP = True Positive, FN = False Negative, TRP = True Positive Rate

b. False Positive rate: The false positive rate is calculated as the ratio between the number of negative events wrongly categorized as positive (false positives) and the total number of actual negative events.
$FPR = \frac{FP}{FP+TN}$ where, FP = False Positive, TN = True Negative, TRP = False Positive Rate

c. Precision: Precision refers to how many classifications we made right based on the attempts we made.
$Precision = \frac{TP}{FP+TP}$

d. Recall: The ability of a model to find all the relevant cases within a data set. It is also called True Positive Rate or Sensitivity.
$Recall = \frac{TP}{TP+FN}$

e. F1-score: Metric to use for classification models as it provides accurate results for both balanced and imbalanced dataset and considers both the precision and recall ability of the model.
$F1\ score = \frac{2*Precision*Recall}{Precision+Recall}$

f. ROC Area: ROC area refers to the area under the ROC curve and is commonly denoted as Area Under the Curve (AUC). The AUC is a metric used to evaluate the performance of a binary classifier in a classification task. The AUC is a useful metric because it is insensitive to class imbalance in the dataset, meaning that it can still accurately evaluate the performance of the classifier even when one class is more prevalent than the other.

g. Confusion Matrix: A confusion matrix is a tabular summary of the number of correct and incorrect predictions made by a classifier. It is used to measure the performance of a classification model. Confusion matrix displays the correctly *and* incorrectly classified instances for all the classes and will, therefore, give a better insight into the performance of your classifier.

4. When you run for 5 FCV and 10 FCV, for each case, describe the performance of the classifiers.

(1) Decision Tree Classifier:

(a) 5-Fold Cross Validation

Confusion Matrix:

220	82
82	80

Class 0 metrics:	**Class 1 metrics:**
TP rate: 0.7284768211920529 FP rate: 0.5 Precision: 0.7333333333333333 Recall: 0.7284768211920529 F1 score: 0.7308970099667774 ROC AUC score: 0.61423841059602	TP rate: 0.5 FP rate: 0.271523178807947 Precision: 0.49382716049382713 Recall: 0.5 F1 score: 0.4968944099378882 ROC AUC score: 0.61423841059602

Correct Instances: 300
Incorrect Instances: 162
Accuracy: 69.4%

(b) 10-Fold Cross Validation

Confusion Matrix:

212	90
90	70

Class 0 metrics: TP rate: 0.7019867549668874 FP rate: 0.5625 Precision: 0.7019867549668874 Recall: 0.7019867549668874 F1 score: 0.7019867549668874 ROC AUC score: 0.5697433774834437	**Class 1 metrics:** TP rate: 0.4375 FP rate: 0.2980132450331126 Precision: 0.4375 Recall: 0.4375 F1 score: 0.4375 ROC AUC score: 0.5697433774834437

Correct Instances: 282
Incorrect Instances: 180
Accuracy: 61.04%

(2) Support Vector Machine

(a) 5-Fold Cross Validation

Confusion Matrix:

253	49
80	80

Class 0 metrics: TP rate: 0.8245033112582781 FP rate: 0.5125 Precision: 0.7522658610271903 Recall: 0.8245033112582781 F1 score: 0.7867298578199052 ROC AUC score: 0.6560016556291391	**Class 1 metrics:** TP rate: 0.4875 FP rate: 0.17549668874172186 Precision: 0.5954198473282443 Recall: 0.4875 F1 score: 0.5360824742268041 ROC AUC score: 0.6560016556291391

Correct Instances: 333
Incorrect Instances: 129
Accuracy: 72.07%

(b) 10-Fold Cross Validation

Confusion Matrix:

249	53

82	78

Class 0 metrics: TP rate: 0.8245033112582781 FP rate: 0.5125 Precision: 0.7522658610271903 Recall: 0.8245033112582781 F1 score: 0.7867298578199052 ROC AUC score: 0.6560016556291391	**Class 1 metrics:** TP rate: 0.4875 FP rate: 0.17549668874172186 Precision: 0.5954198473282443 Recall: 0.4875 F1 score: 0.5360824742268041 ROC AUC score: 0.6560016556291391

Correct Instances: 327
Incorrect Instances: 135
Accuracy: 70.78%

(3) Naïve Bayes Theorem

(a) 5-Fold Cross Validation

Confusion Matrix:

230	72
59	101

Class 0 metrics: TP rate: 0.7615894039735099 FP rate: 0.36875 Precision: 0.7958477508650519 Recall: 0.7615894039735099 F1 score: 0.77834179357022 ROC AUC score: 0.696419701986755	**Class 1 metrics:** TP rate: 0.63125 FP rate: 0.23841059602649006 Precision: 0.5838150289017341 Recall: 0.63125 F1 score: 0.6066066066066067 ROC AUC score: 0.696419701986755

Correct Instances: 331
Incorrect Instances: 131
Accuracy: 71.65%

(b) 10-Fold Cross Validation

Confusion Matrix:

228	74
60	100

Class 0 metrics: TP rate: 0.7549668874172185 FP rate: 0.375 Precision: 0.7916666666666666 Recall: 0.7549668874172185 F1 score: 0.7728813559322034 ROC AUC score: 0.6899834437086093	**Class 1 metrics:** TP rate: 0.625 FP rate: 0.24503311258278146 Precision: 0.5747126436781609 Recall: 0.625 F1 score: 0.5988023952095808 ROC AUC score: 0.6899834437086093

Correct Instances: 328
Incorrect Instances: 134
Accuracy: 70.99%

(4) K Nearest Classifier

(a) 5-Fold Cross Validation

Confusion Matrix:

271	31
119	41

Class 0 metrics: TP rate: 0.8973509933774835 FP rate: 0.74375 Precision: 0.6948717948717948 Recall: 0.8973509933774835 F1 score: 0.7832369942196531 ROC AUC score: 0.5768004966887418	**Class 1 metrics:** TP rate: 0.25625 FP rate: 0.10264900662251655 Precision: 0.5694444444444444 Recall: 0.25625 F1 score: 0.3534482758620689 ROC AUC score: 0.5768004966887418

Correct Instances: 312
Incorrect Instances: 150
Accuracy: 67.53%

(b) 10-Fold Cross Validation

Confusion Matrix:

265	37
124	36

Class 0 metrics: TP rate: 0.8774834437086093 FP rate: 0.775 Precision: 0.6812339331619537 Recall: 0.8774834437086093 F1 score: 0.7670043415340086 ROC AUC score: 0.5512417218543046	**Class 1 metrics:** TP rate: 0.225 FP rate: 0.12251655629139073 Precision: 0.4931506849315068 Recall: 0.225 F1 score: 0.3090128755364807 ROC AUC score: 0.5512417218543046

Correct Instances: 301
Incorrect Instances: 161
Accuracy: 65.15%

(5) Random Forest Classifier
(a) 5-Fold Cross Validation

Confusion Matrix:

253	49
89	71

Class 0 metrics: TP rate: 0.8377483443708609 FP rate: 0.55625 Precision: 0.7397660818713451 Recall: 0.8377483443708609	**lass 1 metrics:** TP rate: 0.44375 FP rate: 0.16225165562913907 Precision: 0.5916666666666667 Recall: 0.44375

F1 score: 0.7857142857142858 ROC AUC score: 0.6407491721854305	F1 score: 0.5071428571428571 ROC AUC score: 0.6407491721854305

Correct Instances: 324
Incorrect Instances: 138
Accuracy: 70.13%

(b) 10-Fold Cross Validation

Confusion Matrix:

250	52
85	75

Class 0 metrics: TP rate: 0.8278145695364238 FP rate: 0.53125 Precision: 0.746268656716418 Recall: 0.8278145695364238 F1 score: 0.7849293563579279 ROC AUC score: 0.6482822847682119	**Class 1 metrics:** TP rate: 0.46875 FP rate: 0.17218543046357615 Precision: 0.5905511811023622 Recall: 0.46875 F1 score: 0.5226480836236934 ROC AUC score: 0.6482822847682119

Correct Instances: 325
Incorrect Instances: 137
Accuracy: 70.35%

References:

[1]https://datagy.io/sklearn-decision-tree classifier/#:~:text=What%20are%20Decision%20Tree%20Classifiers%3F%20Decision%20tree%20classifiers,trees%20can%20also%20be%20used%20for%20regression%20problems.
[2] Naive Bayes classifier - Wikipedia
[3] Random Forest Classification. Background information & sample use… | by Nima Beheshti | Towards Data Science

Code:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import cross_val_score, KFold, cross_val_predict
from sklearn.model_selection import cross_validate
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix,precision_recall_curve, f1_score, roc_curve, roc_auc_score
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn. ensemble import RandomForestClassifier

from sklearn.tree import DecisionTreeClassifier

file_address =
r'C:\Users\akand\Desktop\ML_programming_assignment2\data_csv.csv'
df = pd.read_csv(file_address)

df['famhist'] = df['famhist'].replace({'Absent': 0, 'Present': 1})


x = df.drop('chd', axis = 1)
y = df.chd

x.set_index('row.names', inplace = True)

#model = SVC(kernel = 'linear')
#model = GaussianNB()
#model = KNeighborsClassifier(n_neighbors = 10, weights = 'uniform')
model = RandomForestClassifier(n_estimators = 100)
x_array = np.array(x)
y_array = np.array(y)

from sklearn.model_selection import cross_val_predict
y_pred = cross_val_predict(model, x_array, y_array, cv = 10)

conf_mat = confusion_matrix(y_array, y_pred)
print(f'Confusion Matrix: {conf_mat}')

num_classes = len(np.unique(y_array))
```

```
for i in range(num_classes):
    tp = conf_mat[i, i]
    fp = np.sum(conf_mat[:, i]) - tp
    fn = np.sum(conf_mat[i, :]) - tp
    tn = np.sum(conf_mat) - tp - fp - fn


    tp_rate = tp / (tp + fn)
    fp_rate = fp / (fp + tn)
    precision = tp / (tp + fp)
    recall = tp / (tp + fn)
    f1 = f1_score(y_array, y_pred, pos_label=i)
    fpr, tpr, thresholds = roc_curve(y_array, y_pred, pos_label=i)
    roc_auc = roc_auc_score(y_array, y_pred, multi_class='ovr')

    print(f'Class {i} metrics:')
    print(f'TP rate: {tp_rate}')
    print(f'FP rate: {fp_rate}')
    print(f'Precision: {precision}')
    print(f'Recall: {recall}')
    print(f'F1 score: {f1}')
    print(f'ROC AUC score: {roc_auc}')


correct_indices = np.where(y == y_pred)[0]
currect_len = len(correct_indices)
print(f'Correct Instances: {currect_len}')

incorrect_len = len(y_array) - len(correct_indices)
print(f'Incorrect Instances : {incorrect_len}')

accuracy = (currect_len / (currect_len + incorrect_len)) * 100
print(f'Accuracy: {accuracy} %')
```