CAP4770 Intro to Data Science
Report - Assignment 4

DESCRIPTION

This project aims to build four different machine learning models and testing each of them on four different datasets from the UCI data repositories. The four models used are the Naive Bayesian classifier, Some version of classification trees(Decision Tree), Support Vector Machines, Neural Networks(MLPClassifier). And the four datasets used are Car evaluation, Abalone, KDD Cup 1999. We will analyze the performance of each of the models on each dataset.

DATASETS

- **Car Evaluation**

  Data_set: Car evaluation This Car evaluation dataset is taken from UCI Machine learning repository derived from a simple hierarchical decision model

  Total no. of Observations: 1728

  Input Variable:

  Buying price (vhigh, high, med, low)
  Price of the maintenance (vhigh, high, med, low)
  Number of doors (2, 3, 4, 5more)
  Persons capacity in terms of persons to carry (2, 4, more)
  Size of luggage boot (small, med, big)
  Estimated safety of the car (low, med, high)

  Output Variable:

  Car acceptability (unacc, acc, good, vgood)

- **Abalone**

  Data_set: The Abalone dataset contains the physical measurements of abalones, which are large, edible sea snails.

  Total no. of Observations: 4177

Input Variable:
1 categorical predictor (sex)
7 continuous predictors (Length, Diameter, Height, Whole weight, Shucked weight, Viscera weight, Shell weight)

Output Variable:

number of rings

- **Madelon**

  Data_set: MADELON is an artificial dataset containing data points grouped in 32 clusters placed on the vertices of a five-dimensional hypercube and randomly labeled +1 or -1. The five dimensions constitute 5 informative features. 15 linear combinations of those features were added to form a set of 20 (redundant) informative features. Based on those 20 features one must separate the examples into the 2 classes (corresponding to the +-1 labels). We added several distractor features called 'probes' having no predictive power. The order of the features and patterns was randomized.

  Total no. of Observations: 4400

  Input Variable:
  Real: 20
  Probes: 480
  Total: 500

  Output Variable:

  Target values are provided as +1, -1 labels

- **KDD Cup 1999**

  Data_set: This is the data set used for The Third International Knowledge Discovery and Data Mining Tools Competition, which was held in conjunction with KDD-99 The Fifth International Conference on Knowledge Discovery and Data Mining. The competition task was to build a network intrusion detector, a predictive model capable of distinguishing between ``bad'' connections, called intrusions or attacks, and ``good'' normal connections

  Total no. of Observations: 494,021

  Input Variable:

Provided 41 different attributes to identify the type of connection.

Output Variable:

Type of connections specified as “normal.” and (“smurf”, “ipsweep”...) for attack connections

PARTITIONING THE DATASETS

All the datasets are partitioned with a 70/30 split. That means 70% data for training the model and 30% data for testing the model.

For the Madelon dataset, the UCI repository provides an already partitioned dataset for training in the file “madelon_train.data” and for testing in the file “madelon_valid.data”. We have not used “madelon_test.data” as it is not labeled.

For the KDD Cup dataset, we have used a 10 percent data file provided by the UCI repository as the original data is very long and was taking too long to train the models for it.

The following code was used to split the dataset for all four models:

```
@sk_import model_selection: train_test_split
x_train, x_test, y_train, y_test = train_test_split(data_x,
data_y, test_size=0.3, random_state=1)
```

PARAMETER TUNING FOR MODELS

The Naive Bayes and Decision Tree models require no parameter tuning. Moreover, for the car evaluation and abalone datasets, we have used Categorical Naive Bayes as they contain more categorical features, and for the KDD cup and Madelon datasets, we have used Gaussian Naive Bayes as they contain more continuous features.

For the Support Vector Machine, we have used a *LinearSVC* model that means its kernel parameter is linear. The LinearSVC tends to converge faster for the larger number of samples and helping to train the model faster. And we have limited the max iterations of this model to 10 for Madelon and KDD cup datasets as they are large datasets. Also for the Neural Networks’ MLPClassifier, we have used the limited max iterations. This is done using the “*max_iter = 10”* parameter.

PERFORMANCE

- **Car Evaluation Dataset**

Model	**Accuracy on Training Dataset**	**Accuracy on Testing Dataset**
Naive Bayes	0.8817204301075269	0.8689788053949904
Decision Tree	1.0	0.9691714836223507
Support Vector Machine	0.6989247311827957	0.6955684007707129
Neural Network	0.9007444168734491	0.8689788053949904

The decision tree gives the best performance with around 97% accuracy and the support vector machine gives the worst performance with around 70% accuracy on testing data.

- **Abalone Dataset**

Model	**Accuracy on Training Dataset**	**Accuracy on Testing Dataset**
Naive Bayes	0.9367088607594937	0.9401913875598086
Decision Tree	1.0	0.9027113237639554
Support Vector Machine	0.9401300034211426	0.9433811802232854
Neural Network	0.9397878891549778	0.9449760765550239

All model except the decision tree model gives around 94% accuracy on the testing dataset.

- **Madelon Dataset**

Model	**Accuracy on Training Dataset**	**Accuracy on Testing Dataset**
Naive Bayes	0.714	0.5916666666666667
Decision Tree	1.0	0.7566666666666667

Support Vector Machine	0.6345	0.5483333333333333
Neural Network	0.4985	0.49666666666666665

The Decision tree gives the best performance with around 75% accuracy and the neural network gives the worst performance with around 50% accuracy on the testing dataset.

- **KDD Cup 1999 Dataset**

Model	**Accuracy on Training Dataset**	**Accuracy on Testing Dataset**
Naive Bayes	0.9798301977363554	0.9792992233835109
Decision Tree	1.0	0.9996626340186361
Support Vector Machine	0.9863134517399527	0.9860532903304162
Neural Network	0.9964778753896604	0.9965183830723245

Almost every model gives a good performance but the Decision tree gives the best performance with around 99% accuracy on the testing dataset.

SIZE OF MODELS (in bytes)

Dataset	**Size of Naive Bayes**	**Size of Decision Tree**	**Size of SVM**	**Size of Neural Network**
Car Evaluation	2424	14757	884	34777
Abalone	1690	28248	716	32286
Madelon	16589	22348	4657	1209764
KDD Cup 1999	1898	275	982	107879

TESTING TIME (in secs)

Time the model takes for predicting on testing data.

Dataset	Size of Naive Bayes	Size of Decision Tree	Size of SVM	Size of Neural Network
Car Evaluation	0.028887	0.000549	0.002837	0.003514
Abalone	0.032503	0.000651	0.003094	0.004753
Madelon	0.453962	0.002261	0.003326	0.014286
KDD Cup 1999	0.398642	0.133567	0.029900	0.417220

TRAINING TIME (in secs)

Time the model takes for predicting on training data.

Dataset	Size of Naive Bayes	Size of Decision Tree	Size of SVM	Size of Neural Network
Car Evaluation	0.000969	0.000589	0.000658	0.003493
Abalone	0.001367	0.001651	0.000592	0.005733
Madelon	0.058329	0.007318	0.008728	0.023551
KDD Cup 1999	0.643015	0.426378	0.031308	1.023979

Running Jupyter File with Julia

Programs and libraries required

Jupyter notebook with Julia installed
ScikitLearn
DataFrames
CSV
PyCall

Unzip the file “Assignment4.zip” and run the following command in the terminal.

```
cd Assignment4
jupyter notebook
```

This will open the jupyter notebook in the default browser. Open the folder “*MLmodels”* in jupyter. You will see four folders:

1. Abalone
2. car evaluation
3. KDD
4. Madelon

Each folder contains the corresponding “.data” file containing the dataset and a “.ipynb” file containing all the models for that dataset. Clicking on the “.ipynb” file will open a new tab in the browser. This file contains the code with some markdown as comments. Run the each cell that has code in the given order(from top to bottom) or else it will give errors.