UCI Classification Report – Data Science Assignment 4 – Rohan Saini

Cars Dataset

NB

How you partitioned the data into training and test datasets

- Used train_test_split from ScikitLearn.CrossValidation to do a 70 (training) – 30 (testing) split on the data set.

How you tuned the parameters of the training method (if any)

- Default hyperparameters for CategoricalNB() were used from sklearn.

The performance/error on the training dataset

- Accuracy on Training Data (out of 1) :
  - 0.8684863523573201
- Accuracy on Test Data (out of 1) :
  - 0.8805394990366089

The size of the model

- 2955 bytes

The testing time (time it tasks to predict on the test data)

- 0.004388 seconds

The training time.

- 0.014328 seconds

Decision Tree

How you partitioned the data into training and test datasets

- Used train_test_split from ScikitLearn.CrossValidation to do a 70 (training) – 30 (testing) split on the data set.

How you tuned the parameters of the training method (if any)

- Hyperparameter max_depth=5 used to reduce overfitting by pruning branches.

The performance/error on the training dataset

- Accuracy on Training Data (out of 1) :
  - 0.8709677419354839
- Accuracy on Test Data (out of 1) :
  - 0.8766859344894027

UCI Classification Report – Data Science Assignment 4 – Rohan Saini

The size of the model

- 3381 bytes

The testing time (time it tasks to predict on the test data)

- 0.003137 seconds

The training time.

- 0.007572 seconds

SVM

How you partitioned the data into training and test datasets

- Used train_test_split from ScikitLearn.CrossValidation to do a 70 (training) – 30 (testing) split on the data set.

How you tuned the parameters of the training method (if any)

- Hyperparameter gamma was tuned to 0.5, and C was tuned to 0.5.

The performance/error on the training dataset

- Accuracy on Training Data (out of 1) :
  - 0.9602026049204052
- Accuracy on Test Data (out of 1) :
  - 0.8959537572254336

The size of the model

- 86512 bytes

The testing time (time it tasks to predict on the test data)

- 0.045793 seconds

The training time.

- 0.075393 seconds

Neural Net

How you partitioned the data into training and test datasets

- Used train_test_split from ScikitLearn.CrossValidation to do a 70 (training) – 30 (testing) split on the data set.

How you tuned the parameters of the training method (if any)

- Hyperparameter random_state was set to 1 and max_iter was set to 600 (to reduce overfitting).

The performance/error on the training dataset

- Accuracy on Training Data (out of 1) :
  - 0.9511993382961125
- Accuracy on Test Data (out of 1) :
  - 0.9344894026974951

The size of the model

- 48969 bytes

The testing time (time it tasks to predict on the test data)

- 0.007012 seconds

The training time.

- 2.229638 seconds

Abalone Dataset

- Label encoded feature matrix for first column (sex).
- Split rings counts (labels) into 3 age classes: 1 if [1,8], 2 if [9,10], 3 if [11,29]

NB

How you partitioned the data into training and test datasets

- Used train_test_split from ScikitLearn.CrossValidation to do a 70 (training) – 30 (testing) split on the data set.

How you tuned the parameters of the training method (if any)

- Default hyperparameters for GaussianNB() were used from sklearn.

The performance/error on the training dataset

- Accuracy on Training Data (out of 1) :
  - 0.5740677386247006
- Accuracy on Test Data (out of 1) :
  - 0.5972886762360446

The size of the model

- 1101 bytes

UCI Classification Report – Data Science Assignment 4 – Rohan Saini

The testing time (time it tasks to predict on the test data)

- 0.016571 seconds

The training time.

- 0.072666 seconds

Decision Tree

How you partitioned the data into training and test datasets

- Used train_test_split from ScikitLearn.CrossValidation to do a 70 (training) – 30 (testing) split on the data set.

How you tuned the parameters of the training method (if any)

- Default hyperparameters for GaussianNB() were used from sklearn.

The performance/error on the training dataset

- Accuracy on Training Data (out of 1) :
  - 0.667464933287718
- Accuracy on Test Data (out of 1) :
  - 0.6291866028708134

The size of the model

- 6053 bytes

The testing time (time it tasks to predict on the test data)

- 0.004906 seconds

The training time.

- 0.016741 seconds

SVM

How you partitioned the data into training and test datasets

- Used train_test_split from ScikitLearn.CrossValidation to do a 70 (training) – 30 (testing) split on the data set.

How you tuned the parameters of the training method (if any)

- Hyperparameter gamma was tuned to 0.5, and C was tuned to 0.5.

The performance/error on the training dataset

- Accuracy on Training Data (out of 1) :
  - 0.630516592541909
- Accuracy on Test Data (out of 1) :
  - 0.6275917065390749

The size of the model

- 222837 bytes

The testing time (time it tasks to predict on the test data)

- 0.263678 seconds

The training time.

- 0.334587 seconds

Neural Net

How you partitioned the data into training and test datasets

- Used train_test_split from ScikitLearn.CrossValidation to do a 70 (training) – 30 (testing) split on the data set.

How you tuned the parameters of the training method (if any)

- Hyperparameters random_state = 1 (for reproducibility) and max_iter = 700 used (to limit overfitting).

The performance/error on the training dataset

- Accuracy on Training Data (out of 1) :
  - 0.6616489907629148
- Accuracy on Test Data (out of 1) :
  - 0.6722488038277512

The size of the model

- 48074 bytes

The testing time (time it tasks to predict on the test data)

- 0.007193 seconds

The training time.

- 2.639389 seconds

UCI Classification Report – Data Science Assignment 4 – Rohan Saini

Madelon Dataset

NB

How you partitioned the data into training and test datasets

- Used train data file from dataset for training, validation data for testing.
- Labels were loaded from labels file for both train and validation data.

How you tuned the parameters of the training method (if any)

- Default hyperparameters for GaussianNB() were used from sklearn.

The performance/error on the training dataset

- Accuracy on Training Data (out of 1) :
  - 0.714
- Accuracy on Test Data (out of 1) :
  - 0.5916666666666667

The size of the model

- 16696 bytes

The testing time (time it tasks to predict on the test data)

- 0.014023 seconds

The training time

- 0.085345 seconds

Decision Tree

How you partitioned the data into training and test datasets

- Used train data file from dataset for training, validation data for testing.
- Labels were loaded from labels file for both train and validation data.

How you tuned the parameters of the training method (if any)

- Hyperparameter max_depth=2 used to prune branches and limit overfitting.

The performance/error on the training dataset

- Accuracy on Training Data (out of 1) :
  - 0.651
- Accuracy on Test Data (out of 1) :
  - 0.665

The size of the model

- 1673 bytes

The testing time (time it tasks to predict on the test data)

- 0.001004 seconds

The training time.

- 0.080953 seconds

SVM

How you partitioned the data into training and test datasets

- Used train data file from dataset for training, validation data for testing.
- Labels were loaded from labels file for both train and validation data.

How you tuned the parameters of the training method (if any)

- LinearSVC() with hyperparameter max_iter set to 3000 to decrease overfitting and increase testing accuracy.

The performance/error on the training dataset

- Accuracy on Training Data (out of 1) :
  - 0.524
- Accuracy on Test Data (out of 1) :
  - 0.5166666666666667

The size of the model

- 4745 bytes

The testing time (time it tasks to predict on the test data)

- 0.000987 seconds

The training time.

- 9.956938 seconds

Neural Net

How you partitioned the data into training and test datasets

- Used train data file from dataset for training, validation data for testing.
- Labels were loaded from labels file for both train and validation data.

How you tuned the parameters of the training method (if any)

- MLPClassifier() used with hyperparameters random_state=1 and max_iter=500 to reduce overfitting and increase test accuracy.

The performance/error on the training dataset

- Accuracy on Training Data (out of 1) :
  - 0.547
- Accuracy on Test Data (out of 1) :
  - 0.565

The size of the model

- 1612019 bytes

The testing time (time it tasks to predict on the test data)

- 0.002618 seconds

The training time.

- 1.115958 seconds

KDD Dataset

- Label encoded categorical features 2,3,4 (strings).
- Did binary classification: attack vs no attack (normal).
  - Hence the label vector is 0s and 1s based on attack type.
- 10% dataset was used in this problem.

NB

How you partitioned the data into training and test datasets

- Used train_test_split from ScikitLearn.CrossValidation to do a 70 (training) – 30 (testing) split on the data set.

How you tuned the parameters of the training method (if any)

- Default hyperparameters for GaussianNB() were used from sklearn.

The performance/error on the training dataset

- Accuracy on Training Data (out of 1) :
  - 0.9798099556408937
- Accuracy on Test Data (out of 1) :
  - 0.9798659982322022

The size of the model

- 2005 bytes

The testing time (time it tasks to predict on the test data)

- 0.098137 seconds

The training time.

- 0.266101 seconds

Decision Tree

How you partitioned the data into training and test datasets

- Used train_test_split from ScikitLearn.CrossValidation to do a 70 (training) – 30 (testing) split on the data set.

How you tuned the parameters of the training method (if any)

- Hyperparameter max_depth = 2 used to reduce overfitting (pruning branches).

The performance/error on the training dataset

- Accuracy on Training Data (out of 1) :
  - 0.9868917973245733
- Accuracy on Test Data (out of 1) :
  - 0.9865998232202258

The size of the model

- 1669 bytes

The testing time (time it tasks to predict on the test data)

- 0.015627 seconds

The training time.

- 0.324659 seconds

SVM

How you partitioned the data into training and test datasets

- Used train_test_split from ScikitLearn.CrossValidation to do a 70 (training) – 30 (testing) split on the data set.

How you tuned the parameters of the training method (if any)

- LinearSVC() used with hyperparameter max_iter=250 to reduce computation time.

The performance/error on the training dataset

- Accuracy on Training Data (out of 1) :
  - 0.9904081384790668
- Accuracy on Test Data (out of 1) :
  - 0.9901759026226832

The size of the model

- 1070 bytes

The testing time (time it tasks to predict on the test data)

- 0.010161 seconds

The training time.

- 14.306521 seconds

Neural Net

How you partitioned the data into training and test datasets

- Used train_test_split from ScikitLearn.CrossValidation to do a 70 (training) – 30 (testing) split on the data set.

How you tuned the parameters of the training method (if any)

- MLPClassifier used with hyperparameters random_state = 1 and max_iter=200 (default).

The performance/error on the training dataset

- Accuracy on Training Data (out of 1) :
  - 0.9973020178477447
- Accuracy on Test Data (out of 1) :
  - 0.9971998623546796

The size of the model

- 143117 bytes

The testing time (time it tasks to predict on the test data)

- 0.154773 seconds

The training time.

- 51.398093 seconds

UCI Classification Report – Data Science Assignment 4 – Rohan Saini

Table of Data

Set - Mod	Testing Time (Seconds)	Training Time (Seconds)	Model Size (Bytes)	Training Accuracy	Testing Accuracy
Cars - NB	0.004388	0.014328	2955	86.85%	88.05%
Cars-DT	0.003137	0.007572	3381	87.10%	87.67%
Cars-SVM	0.045793	0.075393	86512	96.02%	89.60%
Cars-NN	0.007012	2.229638	48969	93.45%	95.12%
Ab- NB	0.016571	0.072666	1101	57.41%	59.73%
Ab- DT	0.004906	0.016741	6053	66.75%	62.92%
Ab- SVM	0.263678	0.334587	222837	63.05%	62.76%
Ab- NN	0.007193	2.639389	48074	66.16%	67.22%
Mad - NB	0.014023	0.085345	16696	71.40%	59.17%
Mad - DT	0.001004	0.080953	1673	65.10%	66.50%
Mad - SVM	0.000987	9.956938	4745	52.40%	51.67%
Ma- NN	0.002618	1.115958	1612019	54.70%	56.50%
KDD-NB	0.098137	0.266101	2005	97.98%	97.99%
KDD-DT	0.015627	0.324659	1669	98.69%	98.66%
KDD-SVM	0.010161	14.306521	1070	99.04%	99.02%
KDD-NN	0.154773	51.398093	143117	99.73%	99.72%