CS 519: Applied Machine Learning I

HW3: Classification

Submitted By: Md Ishtiaq Ahmed

Aggie ID: 800606216

Task 4: Report

Dataset Description:

Two datasets is used in the assignment:

1. Digits dataset from sklearn: Digits Dataset is a part of sklearn library. This dataset is made up of 1797 8x8 images. It has 64 numerical features(8×8 pixels) and a 10 class target variable(0-9). This dataset is used to show shows how scikit-learn can be used to recognize images of hand-written digits, from 0-9

2. Wifi_localization dataset: The data set contains wifi signal strength observed from 7 wifi devices on a smartphone collected in indoor space. The data could be used to estimate the location in one of the four rooms. It has total 7 features, 2000 instances and 4 class labels. . The attributes are different parameters of wifi device and based on that it is predicted that the router is situated in what room. There are 4 class label 1,2,3,4 which represents the different room.

Dataset link: https://archive.ics.uci.edu/ml/machine-learning-databases/00422/wifi_localization.txt

Standardization: Both datasets training data is standardized before fitting it to model. However, for decision tree classifier, original data is used since there is no impact of standardized data in decision tree.

Running Time: Running time is measured for both training and testing. Training time is the time required to fit the data in the model. While test time is the time required to predict and measuring accuracy. Both time may vary from the data provided here depending on where and in which condition the program is running.

Part a:

Below is the summary of accuracy and running time for different classifiers tested in Digits dataset:

		Digits Dataset			
		Accuracy		Running Time (ms)	
Classifiers	Parameters	Training Data	Test Data	Training	Testing
perceptron	max_iter=40, eta =0.001, random_state=1	0.956	0.914	29.9847	0.9977
logistic regression	C=100.0, solver='lbfgs', multi_class='multinomial', max_iter=200	1	0.958	263.0462	0.9989

linear support vector machine (SVM)	kernel='linear', C=0.1, random_state=1	0.998	0.983	48.7289	10.992
non-linear SVM (RBF)	kernel='rbf', random_state=1, gamma=0.2, C=1.0	1	0.775	348.9258	78.1178
decision tree	criterion='gini', max_depth=4, random_state=1	0.596	0.592	12.994	1.0004
KNN	n_neighbors=5, weights = 'uniform', p=2	0.984	0.964	1.9969	41.5666

Perceptron, logistic regression, SVM and KNN performed better on the digits dataset and all the classifiers achieved more than 90% accuracy while the decision tree classifier only provided around 60% accuracy. SVM is the best classifier among all considering accuracy and training time.

Below is the summary of accuracy and running time for three different classifiers tested in wifi-localization dataset. This dataset only tested for three classifiers as per the assignment's requirement.

		WiFi Localization Dataset			
		Accuracy		**Running Time**	
Classifiers	**Parameters**	**Training Data**	**Test Data**	**Training**	**Testing**
logistic regression	C=100.0, solver='lbfgs', multi_class='multinomial', max_iter=200	0.986	0.983	97.6159	0.9996
non-linear SVM (RBF)	kernel='rbf', random_state=1, gamma=0.2, C=1.0	0.987	0.978	16.6826	15.6245
decision tree	criterion='gini', max_depth=4, random_state=1	0.983	0.975	13.1509	1.0013

Unlike the digits dataset, the decision tree classifier gives better results on the wifi dataset using the same parameters. Training time is also less than logistic regression and non-linear SVM.

Part b:

For each classifier, performance was checked by tuning one hyper-parameter. Three different value of the parameter is used to analyze the performance.

Some of the common observations are :

- Training time is always higher than the testing time except for KNN classifier. Because KNN classifier do not train the data like other classifiers, it did all the calculation during testing. So, training time is almost zero for KNN while testing time is higher.
- For other classifiers except for KNN, testing time is less and in many cases , it is almost zero
- Database size (no of feature, instances) are also a factor when checking different values of hyper parameter. If the number of instances is less, then some parameter values will not have any impact.

Below is the performance analysis of the classifiers using the digits dataset and wifi localization dataset. For digits dataset all the classifiers were tested but for the wifi localization dataset only logistic regression, non-linear SVM(rbf) and decision tree classifiers were tested as per the assignment requirement.

Perceptron:

			Digits Dataset			
			Accuracy		Running Time (ms)	
Classifiers	**tuned parameter**	**Other Parameters**	**Training Data**	**Test Data**	**Training**	**Testing**
perceptron	eta = 0.001	max_iter=40, random_state=1	0.956	0.914	29.9847	0.9977
perceptron	eta = 0.01		0.969	0.947	32.1354	0.9982
perceptron	eta = 0.1		0.976	0.95	26.9837	1.0013

Above table shows that higher learning rate provides better accuracy for digits dataset. However, there is a jump on accuracy when learning rate is changed from 0.001 to 0.01 but the change is minimal when learning rate changes from 0.01 to 0.1 . This indicates that at this point the classifier is almost converged.

Logistic Regression:

In logistic regression, C is a parameter related to control overfitting which is called inverse-regularization parameter. By decreasing value of C, we increase the regularization strength.

			Digits Dataset			
			Accuracy		Running Time (ms)	
Classifiers	tuned parameter	Other Parameters	Training Data	Test Data	Training	Testing
logistic regression	C=100.0	solver='lbfgs', multi_class='multinomial', max_iter=200	1	0.958	263.046	0.9989
logistic regression	C=10.0		1	0.964	218.453	0.997
logistic regression	C=0.1		0.989	0.967	64.928	0. 810

			Wifi Localization Dataset			
			Accuracy		Running Time (ms)	
Classifiers	tuned parameter	Other Parameters	Training Data	Test Data	Training	Testing
logistic regression	C=100.0	solver='lbfgs', multi_class='multinomial', max_iter=200	0.986	0.983	97.6159	0.7643
logistic regression	C=10.0		0.984	0.983	50.0686	0.7452
logistic regression	C=0.1		0.976	0.98	15.629	0.7448

Higher value of C provides lower accuracy for digits dataset but the behavior is completely opposite for WiFi dataset. It indicates that to get best output, the value of C needs to be optimized based on dataset. Normally a higher C value indicates overfitting. Here, c=100 is fitting the digits training data more closely but changing in C value has no visible impact on WiFi dataset.

Linear support vector machine (SVM):

The C value controls the misclassification penalty in the SVM algorithm.

			Digits Dataset			
			Accuracy		Running Time (ms)	
Classifiers	tuned parameter	Other Parameters	Training Data	Test Data	Training	Testing
linear support vector machine (SVM)	C= 0.1	kernel='linear', random_state=1	0.998	0.983	48.7289	10.992

linear support vector machine (SVM)	C= 1.0		1	0.975	88.0947	27.8429
linear support vector machine (SVM)	C= 10.0		1	0.975	67.7428	15.6288

C = 0.1 provides better accuracy in digits dataset. With increasing, c value accuracy reduced and remain same. So, it will not converge with higher c value. It might converge if c value is lower than 0.1

Non-linear SVM (RBF)

Higher gamma value leads to a tighter decision boundary while lower value leads to a more softer boundary.

			Digits Dataset			
			Accuracy		Running Time (ms)	
Classifiers	**tuned parameter**	**Other Parameters**	**Training Data**	**Test Data**	**Training**	**Testing**
non-linear SVM (RBF)	gamma=0.2	kernel='rbf', random_state=1, C=1.0	1	0.775	348.9258	78.1178
non-linear SVM (RBF)	gamma= 20		1	0.103	493.6149	119.9891
non-linear SVM (RBF)	gamma= 200		1	0.119	475.1045	115.905

			Wifi Localization Dataset			
			Accuracy		Running Time (ms)	
Classifiers	**tuned parameter**	**Other Parameters**	**Training Data**	**Test Data**	**Training**	**Testing**
non-linear SVM (RBF)	gamma=0.2	kernel='rbf', random_state=1, C=1.0	0.987	0.978	39.9961	26.5655
non-linear SVM (RBF)	gamma= 20		1	0.455	500.8051	216.1931
non-linear SVM (RBF)	gamma= 200		1	0.26	809.5426	269.5817

For both digits and wifi dataset, higher gamma value is reducing the accuracy significantly and also increasing the running time. In both cases, the classifier might be overfitting the training data which leads to poor generalization performance.

Decision tree

The maximum depth parameter is a hyperparameter that controls the maximum depth of the tree. The depth of a tree is defined as the number of levels of nodes from the root to the deepest leaf node. A decision tree that is too deep may increase the risk of overfitting but it is also depends on the specific problem and dataset.

			Digits Dataset			
			Accuracy		Running Time (ms)	
Classifiers	**tuned parameter**	**Other Parameters**	**Training Data**	**Test Data**	**Training**	**Testing**
decision tree	max_depth=4	criterion='gini', random_state=1	0.596	0.592	12.994	1.0004
decision tree	max_depth=40		1	0.828	47.1839	1.8692
decision tree	max_depth=400		1	0.828	57.6543	1.3997

Better accuracy for digits dataset is achieved by changing max_depth from 4 to 40. After 40 , accuracy remains same if max_depth is increased which means for this dataset , it started to overfitting. Training time also increased for higher value as the decision tree goes deeper.

			Wifi Localization Dataset			
			Accuracy		Running Time (ms)	
Classifiers	**tuned parameter**	**Other Parameters**	**Training Data**	**Test Data**	**Training**	**Testing**
decision tree	max_depth=4	criterion='gini', random_state=1	0.983	0.975	7.9939	1.3280
decision tree	max_depth=40		1	0.973	8.0008	0.8926
decision tree	max_depth=400		1	0.973	7.997	0.6260

For wifi dataset, the decision tree classifier achieves high accuracy, with all three instances with different max_depth value having an accuracy score of at least 0.973 on the test data. The highest accuracy score of 0.983 is achieved when the maximum depth of the decision tree is set to 4.

KNN

The "p" parameter in KNN refers to the distance metric used to calculate the distance between the data points. It is the power parameter used in the Minkowski distance metric.

			Digits Dataset			
			Accuracy		Running Time (ms)	
Classifiers	**tuned parameter**	**Other Parameters**	**Training Data**	**Test Data**	**Training**	**Testing**
KNN	p=2	n_neighbors=5, weights = 'uniform'	0.984	0.964	0	31.9914
KNN	p=1		0.986	0.989	0	40.5821
KNN	p=3		0.981	0.969	0	1023.053

The highest accuracy is achieved with p=1 with a testing time of 40.5821ms. On the other hand, with p=2 testing time of 31.9914ms is lower, but its accuracy is also slightly lower than the model with p=1. Testing time is much longer with p=3 which is 1023.053ms. If accuracy is concerned, then p=1 is a better choice. If time is concerned, then p=2 also can be used. It is basically a design choice based on the problem statement and the requirement.

Part c: Both training time and testing time is reported in above analysis of part a and part b