DataScience Hw2 report

R05943011 沈恩禾

Regression

This is a binary classification problem, therefore I use logistic regression with following setting:

```
def Regression(self):
        regressor = linear_model.LogisticRegression(
                                    solver = 'liblinear',
                                    multi_class='ovr',
                                    class_weight='balanced')
```

and simply sort the result with a threshold 0.5

```
        regressor.fit(self.trn_x , self.trn_y)
        self.threshold = 0.5
        return [ 1 if x>self.threshold else 0 for x  in
regressor.predict(self.test) ]
```

Decision Tree

I set up decision tree as follow, both entropy and gini criterion are tested and no significant difference presented.

```
    def DecisionTree(self):
        dectree = tree.DecisionTreeClassifier(criterion='entropy')
        dectree.fit(self.trn_x,self.trn_y)
```

SVM

SVM with linear kernel is used.

```
    def Svm(self):
        svm = SVC(kernel='linear')
        svm.fit(self.trn_x,self.trn_y)
```

NN:MLP

A one layer 128 hidden size MLP is used with relu activation and Adam optimizer. This achieves the highest accuracy from my cross validation results using the original dataset from spambase.csv.

```
    def NN(self):
        scalar = StandardScaler()
        scalar.fit(self.trn_x)
        norm_trn_x = scalar.transform(self.trn_x)
        norm_test_x  = scalar.transform(self.test)

        mlp = neural_network.MLPClassifier(
                        hidden_layer_sizes=(128,),
                        activation='relu',
                        learning_rate='adaptive',
                        solver = 'adam'
                        )
        mlp.fit( norm_trn_x , self.trn_y)
```

Results

	LogRegressor	Decision Tree	SVM	MLP
Acc(%)	91.9	92.8	93.5	94.5

- Execution and training time: SVM>>MLP>R>DT
- MLP achieves the best result with the power of neural network, exploring both the linear and non-linear aspects of the problem
- SVM result tells us that there are some outliers that are too close or even exceed the decision boundaries Decision tree method finds out, with SVM we have better chance to sort these instances back to where they belong better: such as something like a spam word frequency exceeds a threshold doesn't guarantee the spam letter conclusion since this attribute needs to coincide with another spam word frequency exceeding the threshold.
- Decision tree and logistic regression tell us that there are multiple decision boundaries which lie in hyper-rectangular regions: like some spam word frequency exceeds the threshold, we categorized by this single result, rather than finding out a trend lies within multiple words combination.