ReviewExercise

April 28, 2015

0.1 Machine Learning Review Exercise

0.1.1 About the exercise

For this exercise, you'll be working in groups to build a model to classify emails as spam based on a variety of feaures which have been extracted from the raw text of those emails.

The methods and techniques for solving the questions below can be found throughout the lecture notes, however, you should also apply your own knowledge, intuition, and creativity to build the best possible model. The performance of your model will measured by the classification accuracy on the test data.

0.1.2 About the data

The dataset was built by researchers at the UCI Machine Learning Institute: https://archive.ics.uci.edu/ml/datasets/Spambase

From the UCI dataset documentation:

Data Set Information:

The "spam" concept is diverse: advertisements for products/web sites, make money fast schemes, chain letters, pornography...

Attribute Information:

The last column of 'spambase.data' denotes whether the e-mail was considered spam (1) or not (0), i.e. unsolicited commercial e-mail. Most of the attributes indicate whether a particular word or character was frequently occuring in the e-mail. The run-length attributes (55-57) measure the length of sequences of consecutive capital letters.

0.1.3 Attributes:

---

48 continuous real [0,100] attributes of type word_freq_WORD = percentage of words in the e-mail that match WORD, i.e. 100 * (number of times the WORD appears in the e-mail) / total number of words in e-mail. A "word" in this case is any string of alphanumeric characters bounded by non-alphanumeric characters or end-of-string.

6 continuous real [0,100] attributes of type char_freq_CHAR] = percentage of characters in the e-mail that match CHAR, i.e. 100 * (number of CHAR occurences) / total characters in e-mail

1 continuous real [1,...] attribute of type capital_run_length_average = average length of uninterrupted sequences of capital letters

1 continuous integer [1,...] attribute of type capital_run_length_longest = length of longest uninterrupted sequence of capital letters

1 continuous integer [1,...] attribute of type capital_run_length_total = sum of length of uninterrupted sequences of capital letters = total number of capital letters in the e-mail

1 nominal {0,1} class attribute of type spam = denotes whether the e-mail was considered spam (1) or not (0), i.e. unsolicited commercial e-mail.

0.2 Relevant Documentation

Relevant Classifiers

- **Logistic Regression** http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
- **K Neighbors Classifier** http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.htr

Preprocessing and Feature Selection

- **Select K Best** http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html#sklearn.f
- **f_classif** http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.f_classif.html#sklearn.feature_selec

Model Evaluation

- **ROC Curve** http://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_curve.html
- **cross validation** http://scikit-learn.org/stable/modules/cross_validation.html

In [53]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

pd.set_option('display.max_rows',100)
pd.set_option('display.max_columns',60)

%matplotlib inline
```

In [54]:
```
# Load the spambase.csv as a pandas DataFrame (last column of data contains Target Data - is_s
spam_data = pd.read_csv("../../data/spambase.csv")

## For quick exploration:
# spam_data.head()
# spam_data.describe()
# spam_data.info()
```

0.2.1 Exercise 1: Explore the dataset and display some visualizations showing how the variables relate to each other

In [55]:
```
# your code here
```

Note: There are 58 features here so you might want to reduce the dimensionality in subsequent steps

0.2.2 Exercise 2: Build a simple logistic regression and visualize it

use the variable "capital_run_length_longest" to predict "is_spam" How accurate is this single feature?

In [56]:
```
# your code here
```

0.2.3 Exercise 3: Use train-test spit to split your data at a 30% mark and run another logistic regression using all variables

Be sure to use random state = 12 so that we can compare results

In [57]:
```
# your code here
```

0.2.4 Exercise 4: Apply cross-validation to see how the model fares across different splits of your date

Use crossvalidation to score model

```
In [58]: # your code here
```

0.2.5 Exercise 5: Compare Performance of Logistic Regression to KNN with 3 neighbors

-Which model is more accurate?

```
In [59]: # your code here
```

0.2.6 Exercise 6: Evaluate Feature Importance

Which features are the most influential in this model?

```
In [60]: # your code here
```

Produce a plot which shows classification accuracy as a function of number of features (k)

```
In [61]: # your code here
```

0.2.7 Exercise 7: Plot the ROC Curve for the logistic regression you chose

```
In [62]: # your code here
```

0.2.8 Exercise 8: Demonstrate how the accuracy of your predictions changes when you set your CV threshold to 50%

```
In [63]: # your code here
```

0.2.9 Exercise 9: Discuss the pro's/con's of moving the threshold away from 50%, why is/isn't this a good idea?

```
In [64]: # your discussion here
```

Explore and use your own reasoning to improve upon your results

What is the highest score you achieve on the training data, as measured by the area under your AUC curve?

0.2.10 Bonus:

Work through the example in the `SupportVectorMachines` folder and repeat the above steps using support vector machines. You'll also find the documentation on the SVC classifier in SkLearn: http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html to be useful.

For computational efficiency, keep the threshold at 50% and use the top K features calculated in Exercise 6.

```
In [65]: # your code here
```

Plot the decision boundary with the two features that best predict whether an email is spam or not spam

```
In [66]: # your visualization here
```

0.3 On your own:

Answer the following questions:

1. Explain which techniques did well and which techniques did not and your reasoning behind why they did/did not perform well.
2. Between Logistic Regression and KNN, which classifier performed better? Why do you think this is?
3. How might you improve this model?
4. How does the number of features you used influence the performance and overfitting?

Submit your answers the above and to the exercises to the class repository, individually