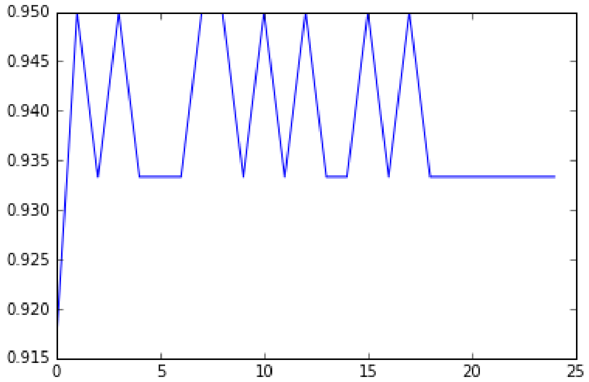
Darren's Data Analytics Blog

A journey through data analytics for a programming geek.

# Wesleyan’s Machine Learning for Data Analysis Week 2



Week 2’s assignment for this machine learning for data analytics course delivered by Wesleyan University, Hartford Connecticut Area in conjunction with Coursera was to build a random forest to test nonlinear relationships among a series of explanatory variables and a categorical response variable. I continued using Fisher’s Iris data set comprising of 3 different types of irises’ (Setosa, Versicolour, and Virginica) with 4 explanatory variables representing sepal length, sepal width, petal length, and petal width.

Using Spyder IDE via Anaconda Navigator and then began to import the necessary python libraries:

```
from pandas import Series, DataFrame
import pandas as pd
import numpy as np
import os
import matplotlib.pylab as plt
from sklearn.cross_validation import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report
import sklearn.metrics
 # Feature Importance
from sklearn import datasets
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.ensemble import RandomForestClassifier
```

Now load our Iris dataset of 150 rows of 5 variables:

```
#Load the iris dataset
iris = pd.read_csv("iris.csv")

# or if not on file could call this.
#iris = datasets.load_iris()
```

Now we begin our modelling and prediction. We define our predictors and target as follows:

```
predictors = iris[['SepalLength','SepalWidth','PetalLength','PetalWidth']]

targets = iris.Name
```

Next we split our data into our training and test datasets with a 60%, 40% split respectively:

```
pred_train, pred_test, tar_train, tar_test = train_test_split(predictors, targets, tes

pred_train.shape
pred_test.shape
tar_train.shape
tar_test.shape
```

Training data set of length 90, and test data set of length 60.

Now it is time to build our classification model and we use the random forest classifier class to do this.

```
classifier = RandomForestClassifier(n_estimators=25)
classifier = classifier.fit(pred_train,tar_train)
```

Finally we make our predictions on our test data set and verify the accuracy.

```
predictions = classifier.predict(pred_test)

sklearn.metrics.confusion_matrix(tar_test,predictions)
sklearn.metrics.accuracy_score(tar_test, predictions)
```

```
Out[1]: 0.94999999999999996
```

Next we figure out the relative importance of each of the attributes:

# fit an Extra Trees model to the data

```
model = ExtraTreesClassifier()
model.fit(pred_train,tar_train)
print(model.feature_importances_)
```

```
[ 0.09603246  0.06664688  0.40937484  0.42794582]
```

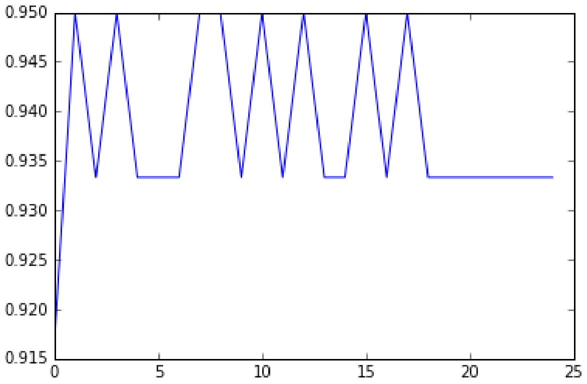
Finally displaying the performance of the random forest was achieved with the following:

```
trees=range(25)
accuracy=np.zeros(25)

for idx in range(len(trees)):
   classifier=RandomForestClassifier(n_estimators=idx + 1)
   classifier=classifier.fit(pred_train,tar_train)
   predictions=classifier.predict(pred_test)
   accuracy[idx]=sklearn.metrics.accuracy_score(tar_test, predictions)

plt.cla()
plt.plot(trees, accuracy)
```

And the plot success was output:



Random forest analysis was performed to evaluate the importance of a series of explanatory variables in predicting a binary or categorical response variable. The following explanatory variables were included as possible contributors to a random forest evaluating the type of Iris based on petal width, petal length, sepal width, sepal length.

The explanatory variables with the highest relative importance scores were petal width (42.8%), petal length (40.9%), sepal length (9.6%), and finally sepal width (6.7%). The accuracy of the random forest was 95%, with the subsequent growing of multiple trees rather than a single tree, adding little to the overall accuracy of the model, and suggesting that interpretation of a single decision tree may be appropriate.

So our model seems to be behaving very well at categorising the iris flowers based on the variables we have available to us.

darren / October 2, 2016 / Coursera, Machine Learning for Data Analysis, Wesleyan Univers