4.6.3 - Linear Discriminant Analysis

May 18, 2018

1 Lab 4.6.3

Import the stock market data.

```
In [11]: # conventional way to import pandas
         import pandas as pd
         # conventional way to import seaborn
         import seaborn as sns
         # conventional way to import numpy
         import numpy as np

         from sklearn import metrics
         import matplotlib.pyplot as plt

         data = pd.read_csv("https://raw.github.com/vincentarelbundock/Rdatasets/master/csv/ISL

         data.head()
```

Out[11]:

	Year	Lag1	Lag2	Lag3	Lag4	Lag5	Volume	Today	Direction
1	2001	0.381	-0.192	-2.624	-1.055	5.010	1.1913	0.959	Up
2	2001	0.959	0.381	-0.192	-2.624	-1.055	1.2965	1.032	Up
3	2001	1.032	0.959	0.381	-0.192	-2.624	1.4112	-0.623	Down
4	2001	-0.623	1.032	0.959	0.381	-0.192	1.2760	0.614	Up
5	2001	0.614	-0.623	1.032	0.959	0.381	1.2057	0.213	Up

We will split the data into data before 2005 and after. Next we will make our tranining data.

```
In [12]: import statsmodels.api as sm
         from scipy import stats
         from patsy import dmatrices

         MarketAfter_2005 = data.query('Year >= 2005')
         MarketBefore_2005 = data.query('Year < 2005')

         y_train, X_train = dmatrices('Direction~Lag1+Lag2', MarketBefore_2005, return_type =

         y_test, X_test = dmatrices('Direction~Lag1+Lag2', MarketAfter_2005, return_type = 'da
```

Now we will use the sklearn lib to do our Linear Discriminant Analysis.

```
In [13]: from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
```

The training process. Please note we will only use Lag1 and Lag2. Hint iloc[:,1:3] and that means. Get first and the secound column of data frame.

```
In [14]: sklearn_lda = LDA(n_components=2) #creating a LDA object

         lda = sklearn_lda.fit(X_train.iloc[:,1:3], y_train.iloc[:,1]) #learning the projectio

         X_lda = lda.transform(X_train.iloc[:,1:3]) #using the model to project X

         X_labels = lda.predict(X_train.iloc[:,1:3]) #gives you the predicted label for each s

         X_prob = lda.predict_proba(X_train.iloc[:,1:3]) #the probability of each sample to be
```

Next we will look at the coefficients of the model for Lag1 and Lag2

```
In [15]: lda.coef_

Out[15]: array([[-0.05544078, -0.0443452 ]])
```

Now we will look at the priors. Therefor we can see that.

$$\hat{\pi}_1 = -0.05544078 \hat{\pi}_2 = -0.0443452$$

```
In [16]: lda.priors_

Out[16]: array([ 0.49198397,  0.50801603])
```

Testing step. Now we will test out model using the data.

```
In [17]: X_test_labels=lda.predict(X_test.iloc[:,1:3])
         X_test_prob = lda.predict_proba(X_test.iloc[:,1:3])
```

To Get the accuracy of the test set. We use the following command.

```
In [18]: np.mean(y_test.iloc[:,1]==X_test_labels)

Out[18]: 0.55952380952380953
```

Let's change the threshod a bit to see whether we can improve the accuracy. The 2nd column of X_test_prob is the probability belongs to UP group. The default value is 0.5, let us first check that.

```
In [19]: threshold = 0.5
         np.mean(y_test.iloc[:,1]==(X_test_prob[:,1]>=threshold))

Out[19]: 0.55952380952380953

In [21]: threshold = 0.48
         np.mean(y_test.iloc[:,1]==(X_test_prob[:,1]>=threshold))

Out[21]: 0.56349206349206349
```