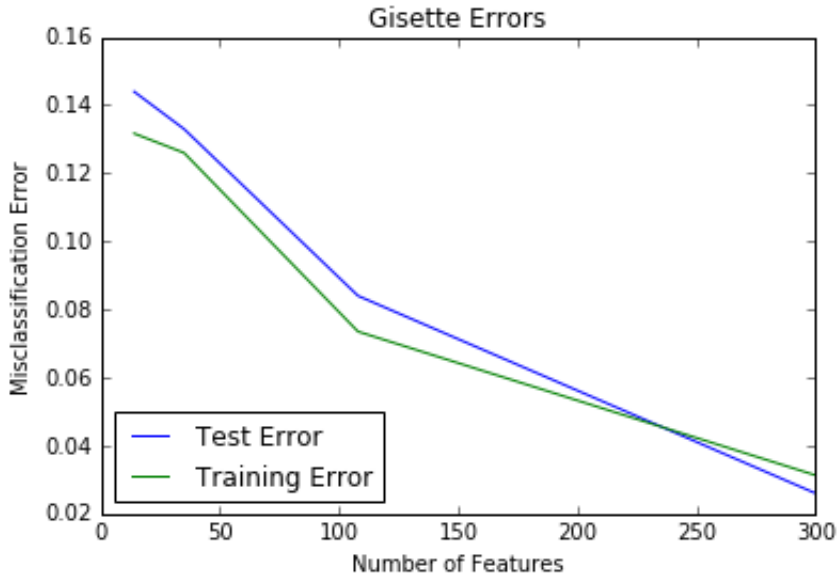
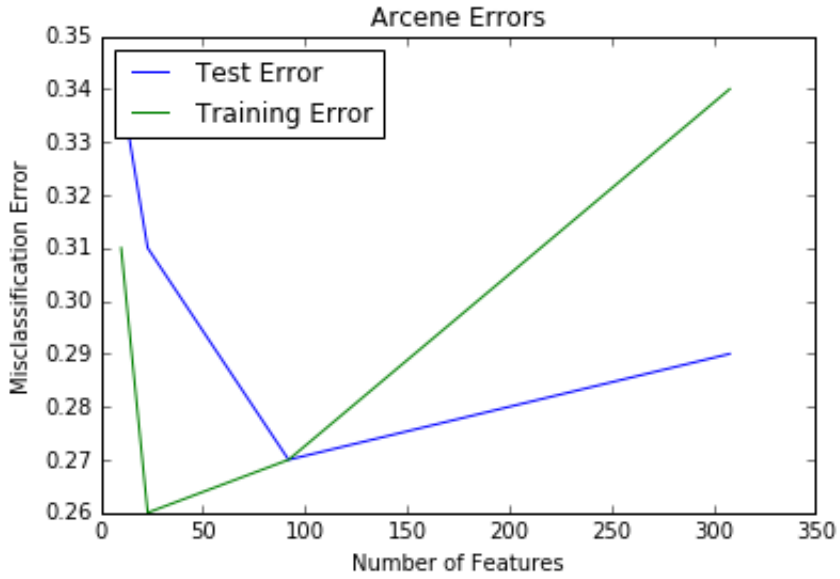
Kyle Shaw, Thomas Johansen
STA 5635

We performed classification on three datasets using logistic regression, employing a Thresholding-based Iterative Selection Procedure (TISP) for feature selection. The three datasets included in the report are Gisette, Arcene, and Madelon. The plots below show the misclassification error vs. the number of features included in the model for the training and testing sets for each dataset. A summary table is also included showing the errors and lambda values for each subset of features.

Gisette



Arcene



Madelon

Summary of Misclassification Error

Set_NumFeatures	Gisette		Arcene		Madelon	
	error	lambda	error	lambda	error	lambda
Train_10	0.132	0.0260	0.31	0.000210	0.405	0.0000500
Test_10	0.144	0.0260	0.34	0.000210	0.410	0.0000500
Train_30	0.126	0.0230	0.26	0.000200	0.383	0.0000240
Test_30	0.133	0.0230	0.31	0.000200	0.405	0.0000240
Train_100	0.074	0.0130	0.27	0.000180	0.364	0.0000160
Test_100	0.084	0.0130	0.27	0.000180	0.403	0.0000160
Train_300	0.031	0.0075	0.34	0.000163	0.348	0.0000065
Test_300	0.026	0.0075	0.29	0.000163	0.415	0.0000065

Some of the code

```
from import_data import import_data
from sklearn import preprocessing
import matplotlib.pyplot as plt
import numpy as np

def updateWeights(X,X1,Y,w, Ncol,Nrow, learningRate, lam, thresh):
    tmp = w[1:Ncol]
    product = np.dot(X,tmp)
    shiftedValue = w[0] + product
    expValue = np.exp(shiftedValue)
    ratio = expValue / (1 + expValue)
    error = Y - ratio
    dLnew = np.dot(np.transpose(X1),error)
    w = w + learningRate * (-lam*w + dLnew/Nrow)
    w = (w) * (abs(w) >= thresh)
    #print(sum())
    return w

def Test(w, X, Y):
    #nomralize data
    X = preprocessing.scale(X)
    #Add column of 1s
    X = np.array(np.c_[np.ones((len(X), 1)), np.matrix(X)])
    Y = [0 if y <= 0 else 1 for y in np.array(Y)]
    wx = 1/(1+np.exp(-1 * np.dot(X, w)))
    Ypredict = [0 if x < .5 else 1 for x in wx]
    results = np.array(Y) - np.array(Ypredict)
    return sum(abs(results))/len(Y)

def Plot(x,y1,y2,title,legendLoc = 1):
    plt.title(title)
    plt.plot(x,y1,label = 'Test Error')
    plt.plot(x,y2,label = 'Training Error')
    plt.legend(loc = legendLoc)
    plt.xlabel('Number of Features')
```

```
    plt.ylabel('Misclassification Error')
    plt.show()

def TrainWeights(X,Y,Xtest,Ytest,k, learnRate = .01, thresh = .001):
    X = preprocessing.scale(X)
    Y = [0 if y <= 0 else 1 for y in np.array(Y)]
    X1 = np.array(np.c_[np.ones((len(X), 1)), np.matrix(X)])
    #initialize variables
    Nrow = len(X)
    Ncol = len(X1[0])
    learningRate = learnRate
    lam = .001
    w = np.array([0]*(Ncol))
    y1 = []
    y2 = []
    for i in range(k):
        w = updateWeights(X,X1,Y,w, Ncol,Nrow, learningRate, lam, thresh)
        y1.append(Test(w,Xtest, Ytest))
        y2.append(Test(w,X,Y))
        if(i%10 == 0):
            print(sum(abs(w) >= thresh))
    return y1,y2

#Gisette Data
##Read in the Data
X, Y, Xtest, Ytest = import_data('gisette', 'gisette_train.data', 'gisette_valid.data',
'gisette_train.labels', 'gisette_valid.labels', head = None)
X.drop(X.columns[len(X.columns)-1], axis=1, inplace=True)
Xtest.drop(Xtest.columns[len(Xtest.columns)-1], axis=1, inplace=True)

niter = 100
y1, y2 = TrainWeights(X,Y,Xtest,Ytest,niter,.1, .02)
# guess and check to get the numbers for the plot below
Plot([11, 31, 101, 299], [.144, .133, .084, .026], [.1317, .126, .0735, .0313],
'Gisette Errors', 3)
min_table['Gisette'] = [min(y1), min(y2)]
```

Bibliography

1. Stéfan van der Walt, S. Chris Colbert and Gaël Varoquaux. The NumPy Array: A Structure for Efficient Numerical Computation, Computing in Science & Engineering, 13, 22-30 (2011), DOI:10.1109/MCSE.2011.37
2. John D. Hunter. Matplotlib: A 2D Graphics Environment, Computing in Science & Engineering, 9, 90-95 (2007), DOI:10.1109/MCSE.2007.55
3. Wes McKinney. Data Structures for Statistical Computing in Python, Proceedings of the 9th Python in Science Conference, 51-56 (2010)
4. Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Édouard Duchesnay. Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research, 12, 2825-2830 (2011)
5. conner.xyz at https://stackoverflow.com/questions/20517650/how-to-delete-the-last-column-of-data-of-a-pandas-dataframe