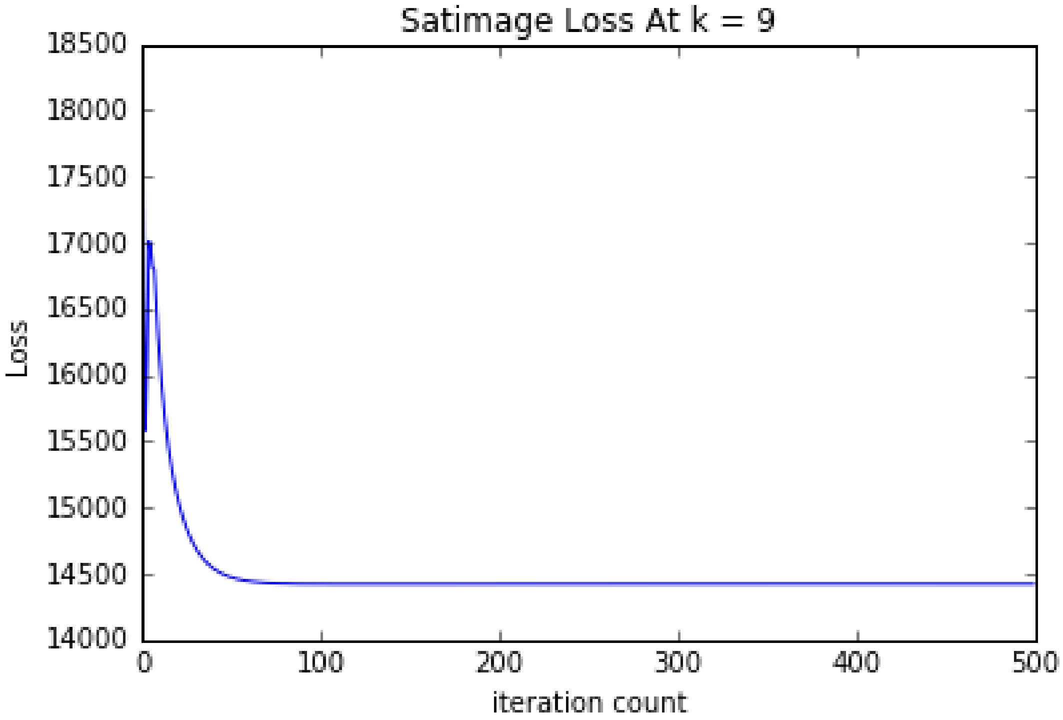
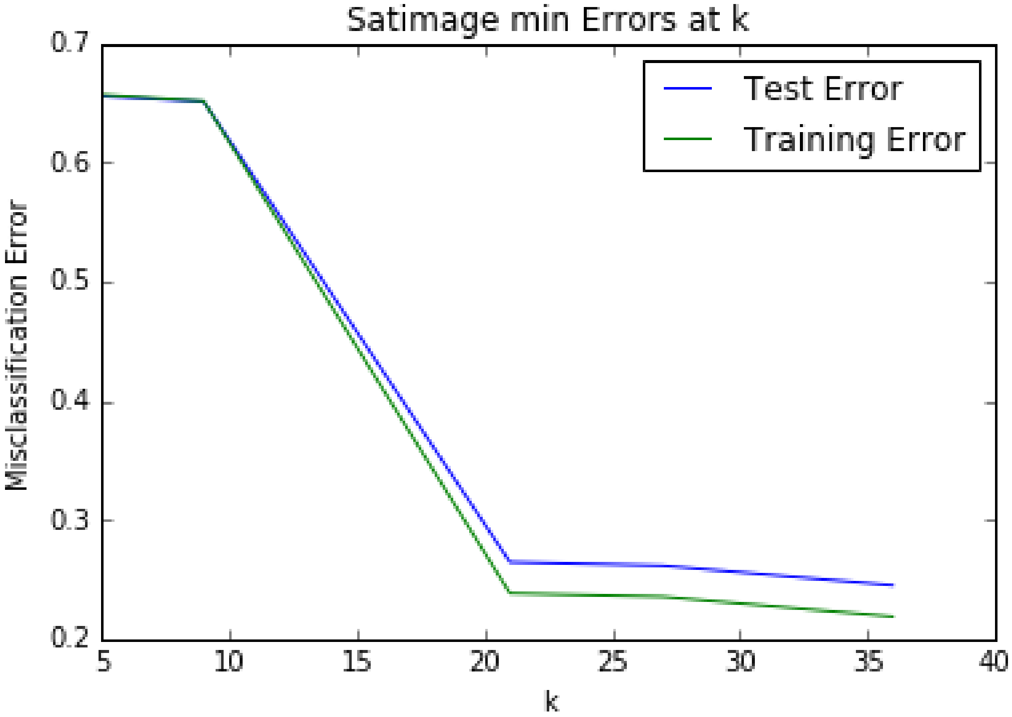
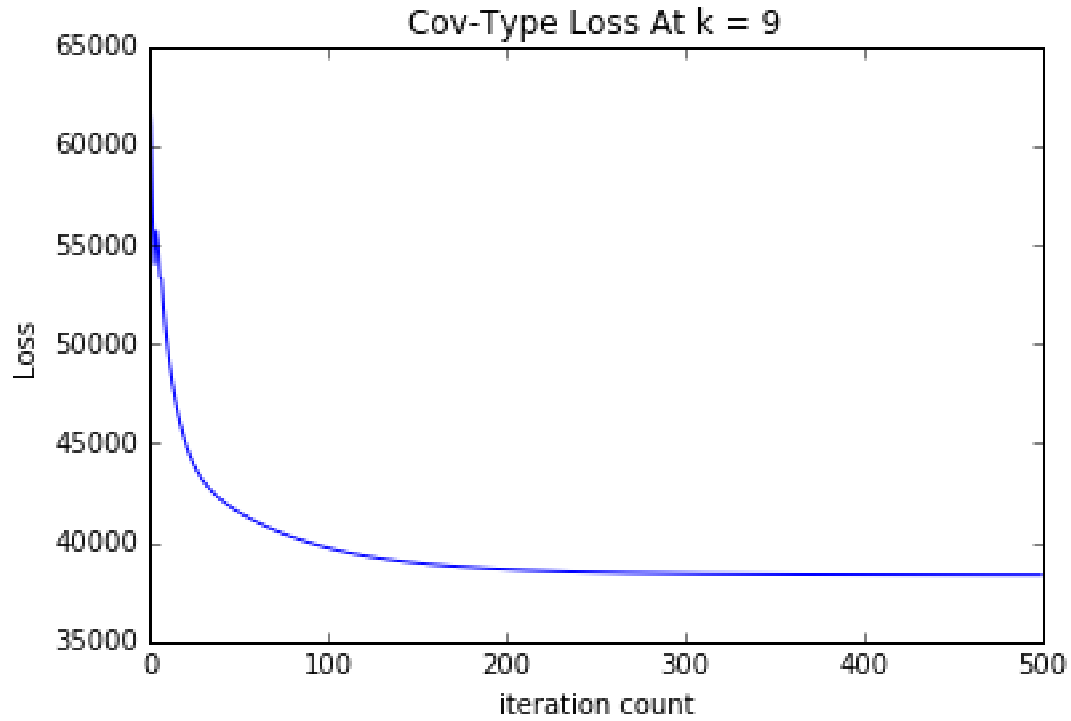
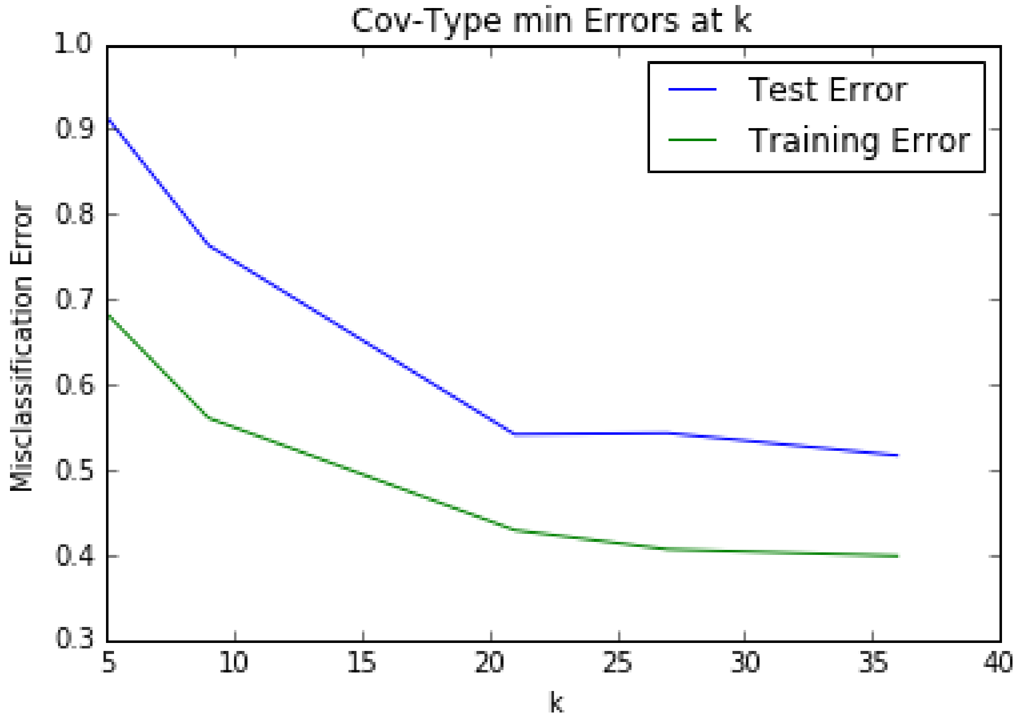
Satimage


Satimage Loss At k = 9
Loss
18500
18000
17500
17000
16500
16000
15500
15000
14500
14000
0
100
200
300
400
500
iteration count

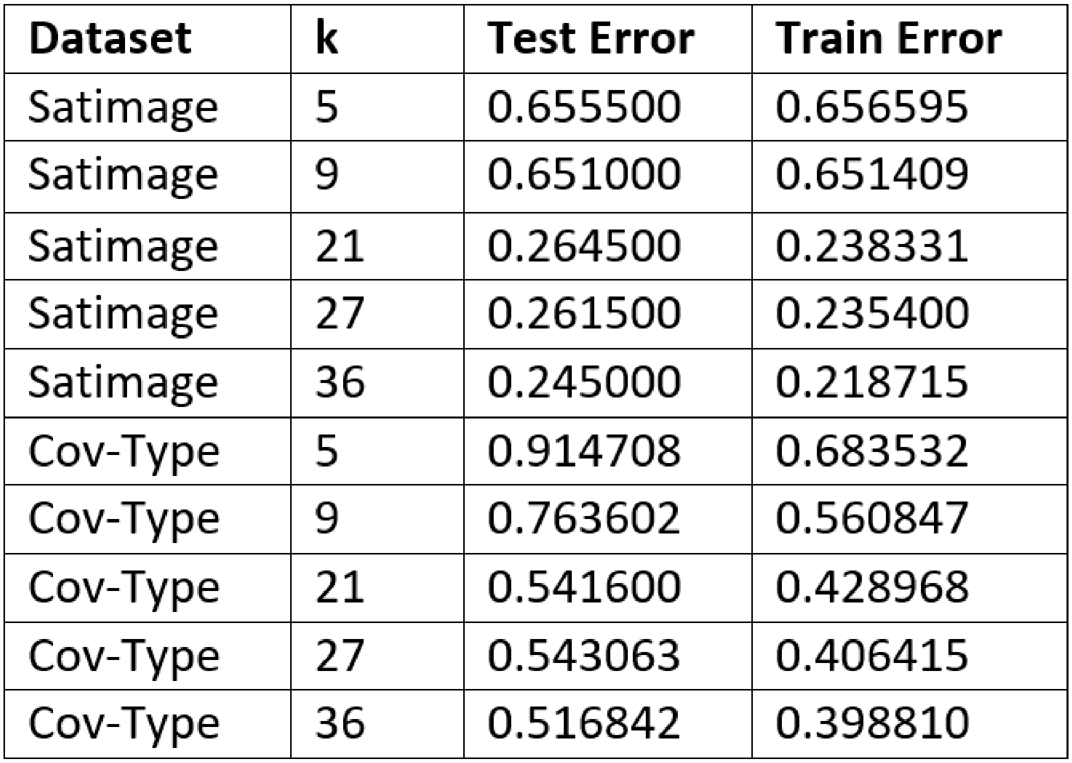

Satimage min Errors at k
Test Error
Training Error
Misclassification Error
0.7
0.6
0.5
0.4
0.3
0.2
5
10
15
20
25
30
35
40
k

Cov-Type


Cov-Type Loss At k = 9
Loss
65000
60000
55000
50000
45000
40000
35000
0
100
200
300
400
500
iteration count


Cov-Type min Errors at k
Misclassification Error
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
5
10
15
20
25
30
35
40
k
Test Error
Training Error

Misclassification Error



Dataset	k	Test Error	Train Error
Satimage	5	0.655500	0.656595
Satimage	9	0.651000	0.651409
Satimage	21	0.264500	0.238331
Satimage	27	0.261500	0.235400
Satimage	36	0.245000	0.218715
Cov-Type	5	0.914708	0.683532
Cov-Type	9	0.763602	0.560847
Cov-Type	21	0.541600	0.428968
Cov-Type	27	0.543063	0.406415
Cov-Type	36	0.516842	0.398810

Code

Included are functions for training and updating the weights, and setting the schedule for eliminating variables. For full code see `https://github.com/johansent/ML_Fall2017/tree/master/hw7`.

```
def updateWeights1(X,Y,w, Ncol,Nrow, learningRate, s):
    L = len(w)
    derivative = np.array([[0.0]*(Ncol)]*L)
    lorenz = 0.0
    X = np.array(X)
    U = np.dot(X, np.transpose(w))
    Uy = [U[i][Y[i]] for i in range(Nrow)]
    for l in range(L):
        diff = (Uy - U[:,l]) - 1
        diff = np.array([-((2 * d) / (1 + d*d)) if d <= 0 else 0 for d in diff])
        logical = [l != Y[i] for i in range(Nrow)]
        diff = diff * logical
        lorenz += sum(np.log(1 + diff**2))
        dmat = diff * X.transpose()
        derivative[l] = np.sum(dmat, axis = 1)
    total = [X[i][0] if logical[i] else 0 for i in range(Nrow)]
    w = w - (learningRate * (derivative + (s * w)))
    loss = (lorenz + s * np.linalg.norm(w, ’fro’))
    return w, loss

def TrainWeights(X,Y,Xtest,Ytest,niter,k, learnRate = .01):
    Nrow = len(X)
    Ncol = len(X.columns)
    L = 7
    X1 = X.copy()
    Xtest1 = Xtest.copy()
    s = .001
    w = np.array([[0]*(Ncol)]*L)
    loss = []
    testErrors = []
    trainingErrors = []
    for i in range(niter):
        w, newloss = updateWeights1(X1,Y,w, Ncol,Nrow, learnRate, s)
        loss.append(newloss)
        if Ncol > k:
            Mi = getMi(Ncol, k, niter, i, u = 100)
            w,X,Xtest,Ncol = getMBest(w, X1, Xtest1, Mi, Ncol)
        testErrors.append(Test(w,Xtest1, Ytest))
        trainingErrors.append(Test(w,X1,Y))
        if(i % 50 == 0):
            print(’i’, i)
    return testErrors,trainingErrors, loss

def getMi(M, k, N, i, u = 100):
    return round(k + (M - k) * max([0,(N - 2 * i)/(2 * i * u + N)]))
```

```
def getMBest(w, X, Xtest, M, Ncol):
    summation = sum(w)
    best = sorted(range(len(summation)), key=lambda i: summation[i])[-M:]
    worst = sorted(range(len(summation)), key = lambda i: summation[i])[0:(Ncol - M)]
    w = np.array([[x[i] for i in sorted(best)] for x in w])
    X.drop(X.columns[worst], axis=1, inplace=True)
    Xtest.drop(Xtest.columns[worst], axis=1, inplace=True)
    return w, X, Xtest, len(X.columns)
```

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