Matthew Virgin

Dr. Chaofan Chen

COS 482

5 May 2023

Homework 4

Task 1: I read in the data by importing numpy and using “loadtxt”:

```
## Import data
data = np.loadtxt("spambase.data", delimiter = ',')
```

I then split the data into inputs and outputs and split that into training inputs, testing inputs, training outputs, and testing outputs using the sklearn package’s train_test_split:

```
## split into train and test sets
X = data[:, :-1]
y = data[:, -1]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
stratify=y)
```

Next, I apply min-max scaling according to the description provided in lecture 23 + 24:

```
## use min-max scaling to scale features in training set between 0 and 1
min_vals = np.amin(X_train, axis = 0)
max_vals = np.amax(X_train, axis = 0)

X_train = (X_train - min_vals) / (max_vals - min_vals)

## use mins and max's of columns from training set to scale test set
X_test = (X_test - min_vals) / (max_vals - min_vals)
```

Task 2: Running the svm gave an accuracy of 93%:

```
Accuracy on test set w/ SVM: 0.9305103148751357
coeffs:  [-1.38118405e+00  9.73120659e-03 -6.38223759e-02  4.49062520e+01
  6.74342765e+00  1.33808745e+00  1.22268101e+01  4.99200526e+00
  3.86463332e+00 -6.81527268e-02 -2.37110123e-01 -1.34310898e+00
  1.57696841e-01 -4.04612322e-02 -4.03060147e-01  1.42493291e+01
  3.82169848e+00  1.23463728e+00  2.89695654e-01  1.27924268e+01
  1.22765082e+00  1.16946695e+00  7.02383526e+00  1.32065210e+01
 -2.54937340e+01 -1.05338892e+01 -2.41322832e+02  3.56179018e+00
 -1.73811867e+01  9.06668561e-01 -3.07433712e+00  5.39195618e+00
 -3.98198063e+00  1.48220882e-01 -2.48527299e+01  5.17121871e+00
 -3.39910077e-02 -2.93538824e+00 -6.30548757e+00 -8.76834752e-01
 -1.02209639e+01 -1.99993029e+01 -2.44886003e+00 -1.39517555e+01
 -5.89012495e+00 -3.10470618e+01 -4.86008017e+00 -2.63777340e+01
 -1.93263437e+00 -6.93942269e+00 -9.21458597e-01  1.41905769e+01
  2.27312042e+01  2.11574143e+01  6.38421440e-01  6.27437861e+01
  5.42504837e+00]
3 most significant coefs pos:  [ 3 52 55]
```

Matching the indices of the 3 most significant coefficients to the list of names given reveals 3d, $, and longest capital run length are the most significant ones for the svm.

3d was the 2nd most influential one.

Running the lrm gave an accuracy of about 83%:

```
Accuracy on test set w/ LRM: 0.8957654723127035
coeffs:  [-0.52517087 -0.83278314  1.26655681  1.35798234  3.82397464  2.47262334
  6.54840504  3.14769596  2.39646863  1.52239017  0.98483045 -1.68910329
  0.80754411  0.36334477  1.72100571  5.27115345  3.41876643  2.50022714
  1.89781101  2.44894677  2.95969025  2.80655686  6.1293604   3.48301855
 -5.52857882 -3.09454724 -4.37447084 -0.21252293 -1.5928283  -2.42795676
 -1.5274959  -0.69472481 -1.95011883 -0.79422026 -2.34745102 -0.086693
 -2.07095358 -0.69138184 -1.40661999 -0.52366894 -1.9077419  -3.25446443
 -1.89753907 -1.85025206 -3.58888881 -3.1146118  -0.77440536 -1.24077088
 -1.60338267 -0.54197231 -0.67315393  3.68240408  5.69193145  0.83533246
  0.48766235  4.5045547   3.28170378]
3 most significant coefs pos:  [ 6 22 52]
```

Matching the indices of those coefficients, the most significant factors were if the email contains remove, 000, and/or $

Task 3: Accuracies of my neural networks:

```
model 1 acc. on test set: 0.9250814332247557
model 2 acc. on test set: 0.3941368078175896
model 3 acc. on test set: 0.3941368078175896
```

The only one that got close to the others is the first one. This is likely because, for linearly separable data, neural networks typically don't perform any better than traditional models.