#LR with L1 accuracy

#LR with L2 accuracy (NOTE: change the penalty to "L2" in last section)

#load libraries

import numpy as np

from sklearn.linear\_model import LogisticRegression

from sklearn import datasets

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

#Create the Datasets

# Load the iris dataset

iris = datasets.load\_iris()

# Create X from the features

X = iris.data

# Create y from output

y = iris.target

#Display original data

print("Features: ", X[0:5])

print ("Target: ", y)

# Remake the variable, keeping all data where the category is not 2.

X = X[y != 2]

y = y[y != 2]

#display features and target data

#print("Features: ", X[0:5])

#print ("Target: ", y)

#split data into training/test sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=0)

#standardize features

# Create a scaler object

sc = StandardScaler()

# Fit the scaler to the training data and transform

X\_train\_std = sc.fit\_transform(X\_train)

# Apply the scaler to the test data

X\_test\_std = sc.transform(X\_test)

#Run LR with L1 at various strengths \*\*\*\*\*\*NOTE - change to L2 for second run!

C = [10, 1, .1, .001]

for c in C:

clf = LogisticRegression(penalty='l1', C=c)

clf.fit(X\_train, y\_train)

print('C:', c)

print('Coefficient of each feature:', clf.coef\_)

print('Training accuracy:', clf.score(X\_train, y\_train))

print('Test accuracy:', clf.score(X\_test, y\_test))

print('')