**EX-6**

1. **dataset3Paramas**

%% %%%%%%%%%% WORKING: SOLUTION1 %%%%%%%%%%

% C\_list = [0.01 0.03 0.1 0.3 1 3 10 30]';

% sigma\_list = [0.01 0.03 0.1 0.3 1 3 10 30]';

%

% prediction\_error = zeros(length(C\_list), length(sigma\_list));

% for i = 1:length(C\_list)

% for j = 1: length(sigma\_list)

% C\_test = C\_list(i);

% sigma\_test = sigma\_list(j);

% model = svmTrain(X, y, C\_test, @(x1, x2) gaussianKernel(x1, x2, sigma\_test));

% predictions = svmPredict(model, Xval);

% prediction\_error(i,j) = mean(double(predictions ~= yval));

% end

% end

%

% % Finding row and col corresponding to min(prediction\_error)

% [values, row\_index]=min(prediction\_error);

% [~ ,col] = min(values);

% row = row\_index(col);

%

% % C and sigma corresponding to min(prediction\_error)

% C = C\_list(row);

% sigma = sigma\_list(col);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% %%%%%%%%%% WORKING: SOLUION 2 %%%%%%%%%%%%%%

C\_list = [0.01 0.03 0.1 0.3 1 3 10 30]';

sigma\_list = [0.01 0.03 0.1 0.3 1 3 10 30]';

prediction\_error = zeros(length(C\_list), length(sigma\_list));

result = zeros(length(C\_list)+length(sigma\_list),3);

row = 1;

for i = 1:length(C\_list)

for j = 1: length(sigma\_list)

C\_test = C\_list(i);

sigma\_test = sigma\_list(j);

model = svmTrain(X, y, C\_test, @(x1, x2) gaussianKernel(x1, x2, sigma\_test));

predictions = svmPredict(model, Xval);

prediction\_error(i,j) = mean(double(predictions ~= yval));

result(row,:) = [prediction\_error(i,j), C\_test, sigma\_test];

row = row + 1;

end

end

% Sorting prediction\_error in ascending order

sorted\_result = sortrows(result, 1);

% C and sigma corresponding to min(prediction\_error)

C = sorted\_result(1,2);

sigma = sorted\_result(1,3);

1. **emailFeatures**

%% WORKING: SOLUTION 1 %%%%%%

% for i = 1:length(word\_indices)

% x1 = ([1:n] == word\_indices(i));

% x = x | x1';

% end

%% WORKING: SOLUTION 2 %%%%%%

for i = 1:length(word\_indices)

x(word\_indices(i)) = 1;

end

1. **gaussianKernel**

sim = exp(-1\*sum(abs(x1-x2).^2)/(2\*sigma^2));

1. **processEmail**

%% %%%%% WORKING: SOLUTION %%%%%%%%%%

% find index of the word in vocabList (if Exist)

index = find(strcmp(str,vocabList),1);

% Add the index in the vector word\_indices

word\_indices = [word\_indices; index];