function submit()

addpath('./lib');

conf.assignmentSlug = 'regularized-linear-regression-and-bias-variance';

conf.itemName = 'Regularized Linear Regression and Bias/Variance';

conf.partArrays = { ...

{ ...

'1', ...

{ 'linearRegCostFunction.m' }, ...

'Regularized Linear Regression Cost Function', ...

}, ...

{ ...

'2', ...

{ 'linearRegCostFunction.m' }, ...

'Regularized Linear Regression Gradient', ...

}, ...

{ ...

'3', ...

{ 'learningCurve.m' }, ...

'Learning Curve', ...

}, ...

{ ...

'4', ...

{ 'polyFeatures.m' }, ...

'Polynomial Feature Mapping', ...

}, ...

{ ...

'5', ...

{ 'validationCurve.m' }, ...

'Validation Curve', ...

}, ...

};

conf.output = @output;

submitWithConfiguration(conf);

end

function out = output(partId, auxstring)

% Random Test Cases

X = [ones(10,1) sin(1:1.5:15)' cos(1:1.5:15)'];

y = sin(1:3:30)';

Xval = [ones(10,1) sin(0:1.5:14)' cos(0:1.5:14)'];

yval = sin(1:10)';

if partId == '1'

[J] = linearRegCostFunction(X, y, [0.1 0.2 0.3]', 0.5);

out = sprintf('%0.5f ', J);

elseif partId == '2'

[J, grad] = linearRegCostFunction(X, y, [0.1 0.2 0.3]', 0.5);

out = sprintf('%0.5f ', grad);

elseif partId == '3'

[error\_train, error\_val] = ...

learningCurve(X, y, Xval, yval, 1);

out = sprintf('%0.5f ', [error\_train(:); error\_val(:)]);

elseif partId == '4'

[X\_poly] = polyFeatures(X(2,:)', 8);

out = sprintf('%0.5f ', X\_poly);

elseif partId == '5'

[lambda\_vec, error\_train, error\_val] = ...

validationCurve(X, y, Xval, yval);

out = sprintf('%0.5f ', ...

[lambda\_vec(:); error\_train(:); error\_val(:)]);

end

end