Lab Class 5

index	Numbe	Question	Correct (a fraction)	Max Mark	Comments
1_1	1_1	Create a python function that computes $\boldsymbol{\Phi}$ for the linear basis, $$\boldsymbol{\Phi} = \begin{bmatrix} \mathbf{x} &\mathbf{1}\end{bmatrix}$$ Name your function `linear`. `Phi` should be in the form of a *design matrix* and `x` should be in the form of a `numpy` two dimensional array with n rows and 1 column Calls to your function should be in the following form: ```python Phi = linear(x) ```	2	2	
1_2	1_2	Create a python function that accepts, as arguments, a python function that defines a basis (like the one you've just created called `linear`) as well as a set of inputs and a vector of parameters. Your new python function should return a prediction. Name your function `prediction`. The return value `f` should be a two dimensional `numpy` array with n rows and 1 column, where n is the number of data points. Calls to your function should be in the following form:```python f = prediction(w, x, linear)```	2	2	
1_3	1_3	Create a python function that computes the sum of squares objective function (or error function). It should accept your input data (or covariates) and target data (or response variables) and your parameter vector `w` as arguments. It should also accept a python function that represents the basis. Calls to your function should be in the following form:```python e = objective(w, x, y, linear)```	3	3	
1_4	1_4	Create a function that solves the linear system for the set of parameters that minimizes the sum of squares objective. It should accept input data, target data and a python function for the basis as the inputs. Calls to your function should be in the following form:```python w = fit(x, y, linear)```	3	3	
1_5	1_5	Fit a linear model to the olympic data using these functions and plot the resulting prediction between 1890 and 2020. Set the title of the plot to be the error of the fit on the *training data*.	3	5	The value of error should be in the plot title.
2_1	2_1	Start by creating a python-function called `quadratic`. It should compute the quadratic basis. $$\boldsymbol{\Phi} = \begin{bmatrix} \mathbf{1} & \mathbf{x} & \mathbf{x}^2\end{bmatrix}$$ It should be called in the following form: ```python Phi = quadratic(x)```	5	5	
2_2	2_2	Use this to compute the quadratic fit for the model, again plotting the result titled by the error.	3	5	The value of error should be in the plot title.
3_1	3_1	For both the linear and quadratic models, fit the model to the data up until 1980 and then compute the error on the held out data (from 1980 onwards).	8	8	
3_2	3_2	Which model performs better on the validation data?	2	2	
4_1	4_1	Modify your code to pass these additional arguments to the python function for creating the basis. Do this for each of your functions `predict`, `fit` and `objective`. You will find `*args` (or `**kwargs`) useful.			
4_2	4_2	`predict`	5	5	
4_3	4_3	`fit`	5	5	
4_4	4_4	`objective`	5	5	
4_5	4_5	Write code that tries to fit different models to the data with polynomial basis. Use a maximum degree for your basis from 0 to 17. For each polynomial store the *hold out validation error* and the *training error*. When you have finished the computation plot the hold out error for your models and the training error for your p. When computing your polynomial basis use `offset=1956.` and `scale=120.` to ensure that the data is mapped (roughly) to the -1, 1 range.	7	7	
4_6	4_6	Which polynomial has the minimum training error?	2	2	
4_7	4_7	Which polynomial has the minimum validation error?	1	1	Good plotting
5_1	5_1	Write code that computes the leave one out validation error for the olympic data and the polynomial basis. Use the functions you have created above: objective, fit, polynomial. Compute the leave-one-out cross validation error for basis functions containing a maximum degree from 0 to 17.			
5_2	5_2	Looping structure correct	5	5	
5_3	5_3	Leave one out error normalised	5	5	
5_4	5_4	Correct removal of hold out point from data	5	5	
5_5	5_5	General other code correctness (right degree, right result	5	5	
6_1	6_1	Perform k-fold cross validation on the olympic data with your polynomial basis. Use k set to 5 (e.g. five fold cross validation). Do the different forms of validation select different models?			
6_2	6_2	Correctly randomizing partitions (not having partitions that are contiguous	5	5	
6_3	6_3	Correct loop structure	5	5	
6_4	6_4	Trial different runs and see that different model orders are selected according to different partitions.	0	5	This part of question is not answered and can not inferred from the code; Do the different forms of validation select different models? Does five fold cross validation always select the same model?" You are expected to repeat the 5-fold cross validation several runs and see whether it always select the same model.
6_5	6_5	General code correctness.	5	5	
		Mark is from 100	91	100	
		Mark is from 5	4,55	5	