1. You are given a dataset \((X, y)\), \(X \in \mathbb{R}^{n \times m}\) (with \(n\) instances, each of dimension \(m\)), \(y \in \mathbb{R}^m\) (\(m\)-dimensional vector, with values corresponding to the instances in \(X\)).

(a) **Linear Regression:** Write a piece of Python code implementing the linear least squares regression algorithm. The program takes as input \((X, y)\) and produces as output a weight vector \(w \in \mathbb{R}^m\). Write a piece of Python code that reads the weight vector \(w\) (model) and predicts the corresponding value \(\hat{y}\) for a new test instance \(x\). Run your code on the dataset and report the squared loss obtained.

(b) **Ridge Regression:** Write a piece of Python code that takes \((X, y)\) and an additional parameter \(\lambda\) to implement Ridge regression on the same dataset. Experiment with \(\lambda = \{0.01, 0.1, 1, 10, 100\}\) and find the best \(\lambda\) by 5 fold cross validation (report the average cross-validation error with each \(\lambda\)) and report the squared error on the entire test data with the best \(\lambda\). (In order to perform cross-validation, first divide the entire training dataset into 5 equal parts (folds). Then, take fold-1 as your test subset and the remaining 4 folds as your train subset. Repeat the same for fold-2, fold-3, fold-4 and fold-5.)

(c) **Kernel Ridge Regression:** Implement the kernel ridge regression on the same dataset by writing a piece of Python code by adapting the previous piece of code for ridge regression. Run your program with linear, polynomial degree-2, polynomial degree-3, RBF width-1, RBF width-4 kernels. Experiment with \(\lambda = \{0.01, 0.1, 1, 10, 100\}\) and find the best \(\lambda\) by 5 fold cross validation (report the average cross-validation error with each \(\lambda\)) and report the squared error on the entire test data with the best \(\lambda\).