1. Filter1 should be returned to the user.   
   The initial loop has helped us find the best C (cost of constraints) value by recording the validation errors. All the filters then make use of the minimal validation error as the C value.  
   Filter0 is trained on the training data and then tested on the validation data. As we already made use of the validation data to find our C, this does not really help us see how well the model generalizes to new data in comparison to if we used the test data for testing.  
   Filter2 uses both training and validation data during training, then is tested on the test data. This is likely one of the better approaches, but since we’ve already made use of the validation data to obtain our C this model might also be showcasing some bias towards the validation data.  
   Filter3 uses the entire spam dataset for training and test data for testing. As it will have seen the entire dataset, this model will more than likely be overfitted to the data. This is likely the worst filter to choose if we want to be able to generalize to new data.  
   Filter1 is trained on the training data and tested on the test data. It has, like the other filters, been provided with a C value that corresponds to the minimal validation error. This filter runs the least risk of introducing bias and is likely going to be the best one to pick.
2. Filter0 error = 0.0675  
   Filter1 error = 0.08489388  
   Filter2 error = 0.082397  
   Filter3 error = 0.02122347  
   Filter3 error being the lowest is likely due to overfitting to the data, as this model got to see the entire dataset when it was trained.   
   Filter0 being the second lowest can likely be explained by overfitting to the validation data. Validation errors were used to choose the C parameter in training and then the same validation data were used for testing.   
   Interestingly, filter2 and filter1 provide a very similar error, even though filter2 was trained on both the training and validation data while filter1 only used the training data. Both were evaluated on the test data. It seems like the additional data did not provide any noticeable difference in performance. Perhaps the training data is already big enough, so adding the validation data to it did not make enough of a difference.
3. Predictions:   
   [1] -1.998999 1.560584 1.000278 -1.756815 -2.669577 1.291312 -1.068444

[8] -1.312493 1.000184 -2.208639