MATLAB 2

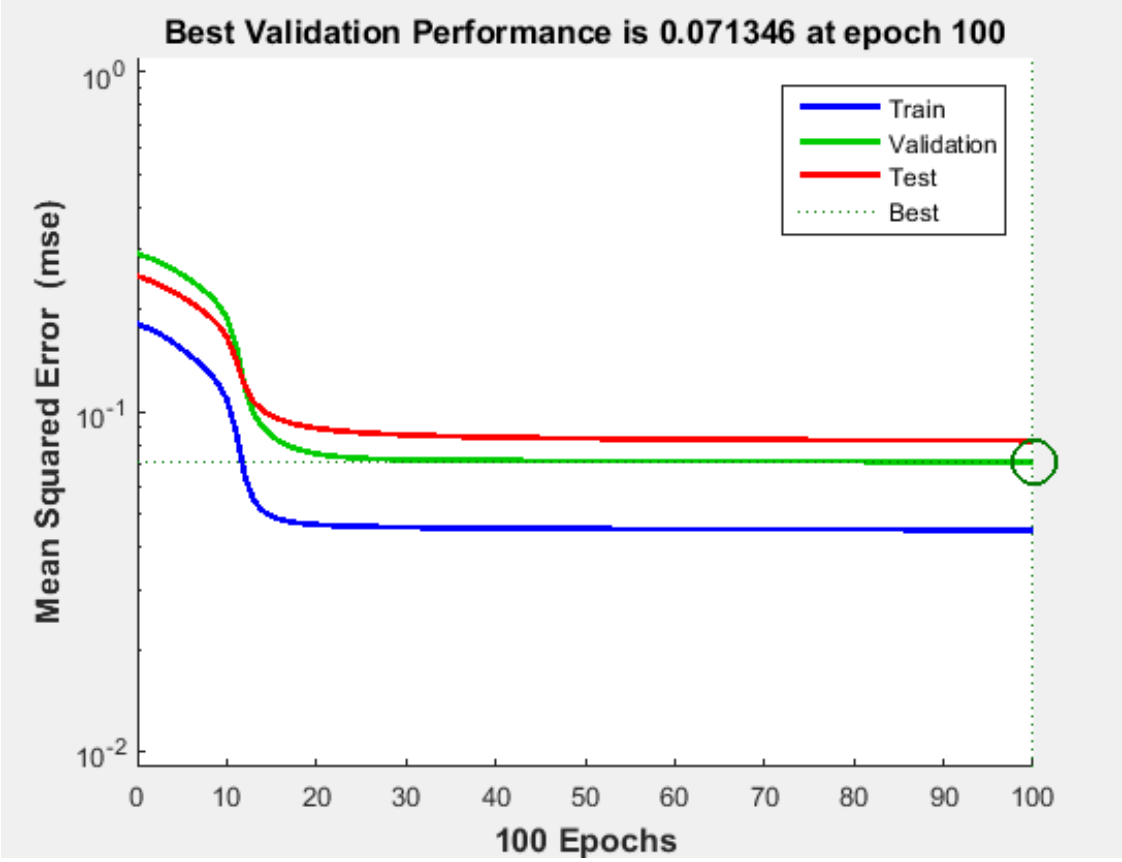
Predicting phenotypic and environmental characteristics

In this project we use a set of 223 transcriptional profiling samples from the gram-negative bacterium Escherichia coli, which is the well-studied organism with great importance to human health and biotechnology. The dataset contains 4502 features, the first 6 corresponding to gene ID, strain, medium, environmental and genetic perturbation, and information about the growth rate. The last entries correspond to the expression of all genes in the bacterium.

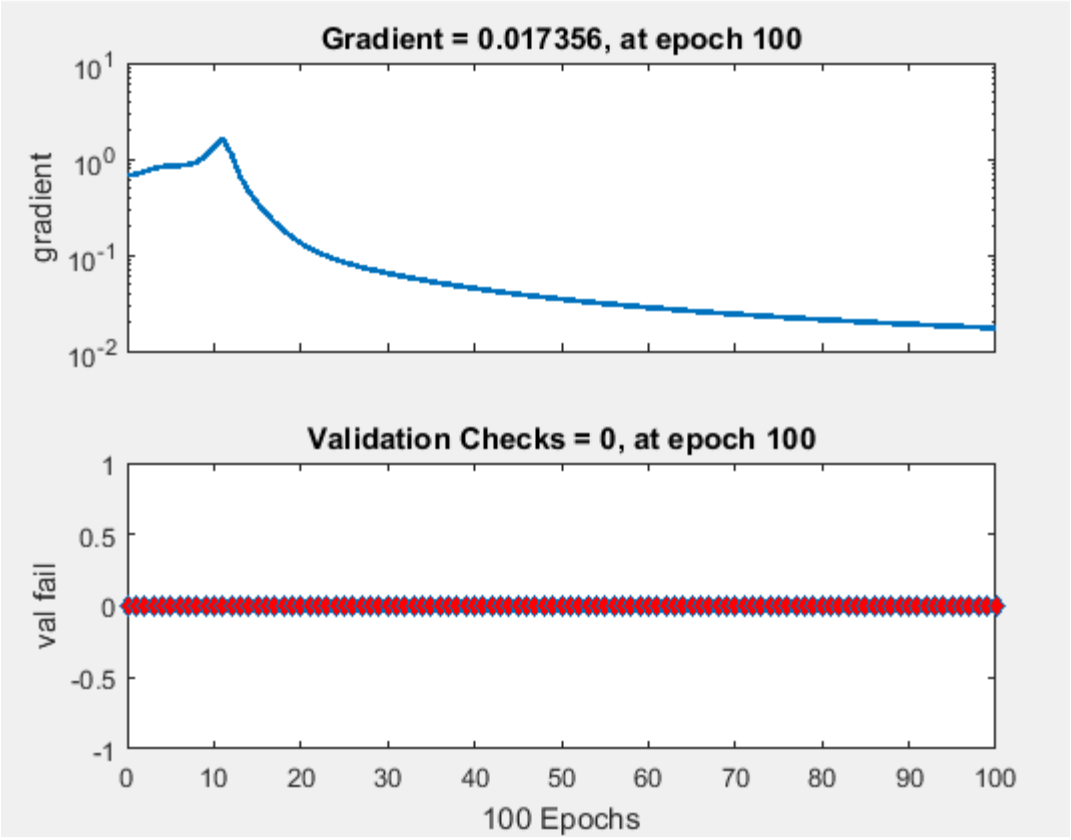
We have two excel files with the data and definition of the features and gene names.

First we created a predictor of the bacterial growth attribute by using only the expression of the genes as attributes.

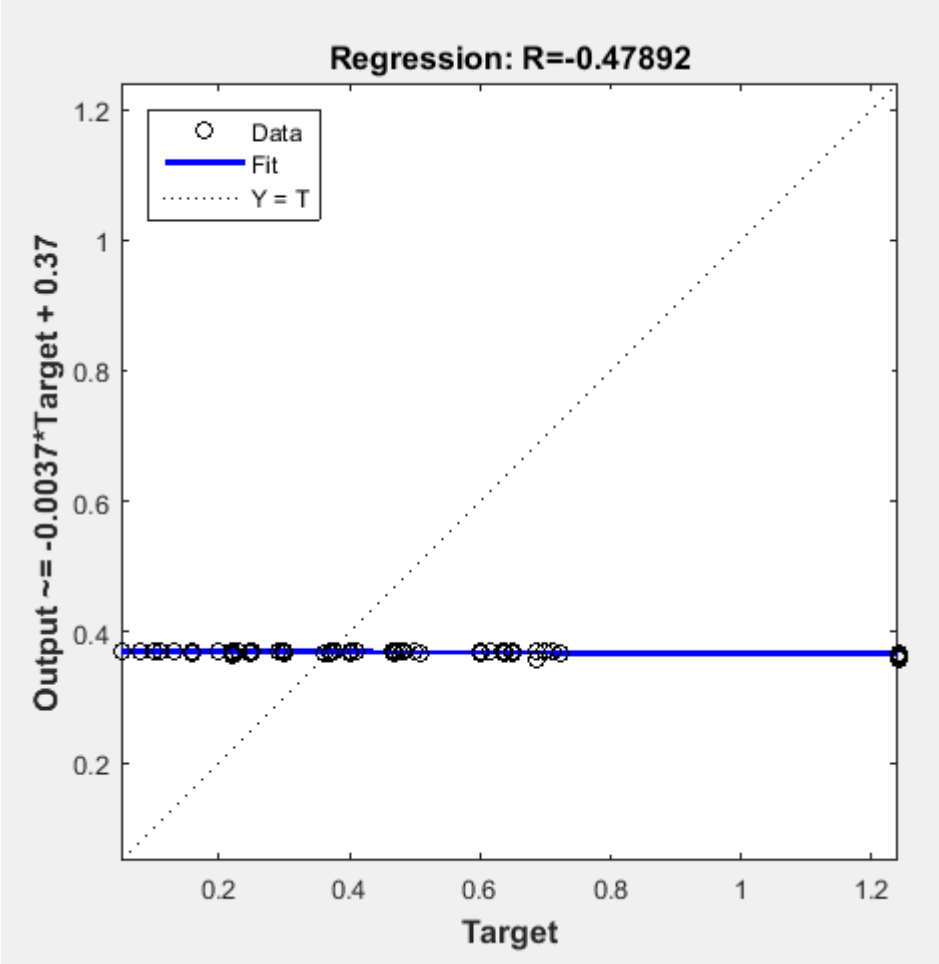
Best validation performance is 0.071346 as show below an it is at epoch 100.



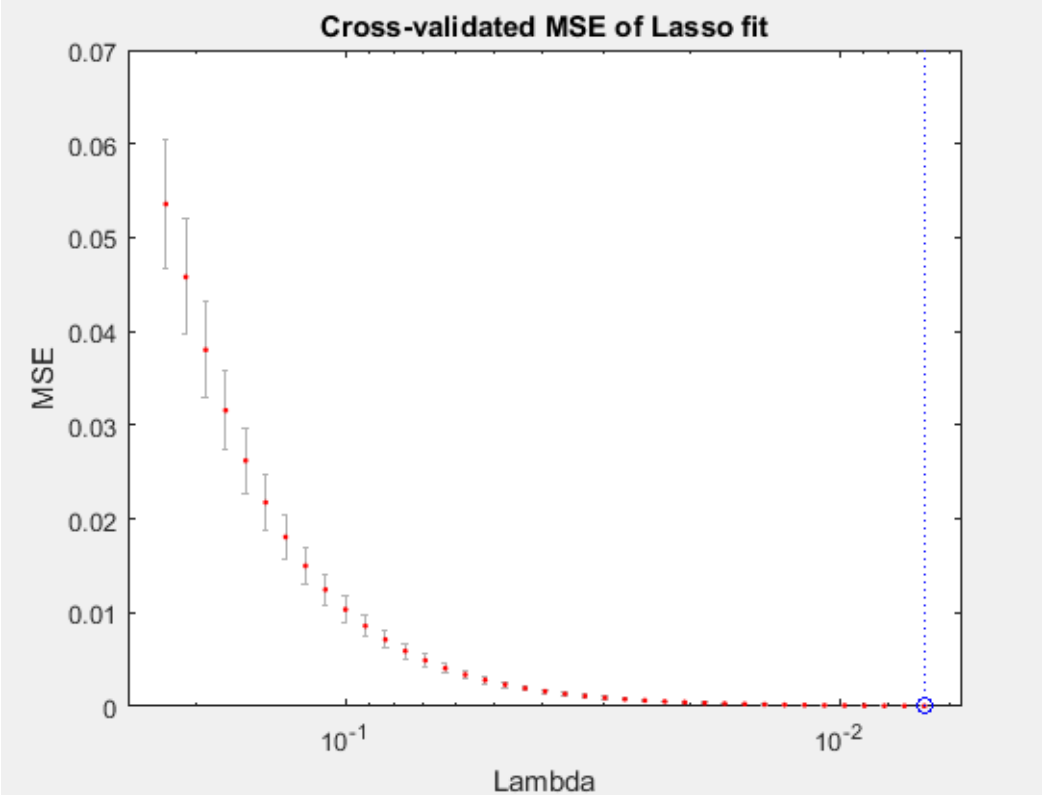
Gradient map shows at epoch 100 a gradient equal to 0.017356:



Regrassion plot with R=-0.47892 is shown below:



Using a regularized regression technique lasso (least absolute shrinkage and selection operator). This method uses the constraint that $\|\beta\|_1$, the L1-norm of the parameter vector, is no greater than a given value. This is equivalent to an unconstrained minimization of the least-squares penalty with $\alpha\|\beta\|_1$ added. In a Bayesian context, this is equivalent to placing a zero-mean Laplace prior distribution on the parameter vector.



You can see the optimal constrained parameter value (Lambda) circled on the plot.

10-fold cross-validation generalization error ***0.0083564***

Number of features that have non-zero coefficients ***38***

Second part. I expend my predictor to report the confidence interval of the prediction by using the bootstrapping method. We use bootstrapped confidence intervals because then we can basically forget about things like normality.

A confidence interval is an observed interval (i.e., it is calculated from the observations), in principle different from sample to sample, that frequently includes the value of an unobservable parameter of interest if the experiment is repeated. How frequently the observed interval contains the parameter is determined by the confidence level or confidence coefficient. It applied for a full assessment of the accuracy in comparison with the point estimate.

In applied practice, confidence intervals are typically stated at the 95% confidence level.

The bootstrap method is a technique used for determining, among other things, the accuracy of statistics. Traditionally, standard errors have been calculated using well

known formulae often based on assumptions that are not satisfied or only approximately satisfied. In essence, the bootstrap method relies on resampling with replacement from the given sample and calculating the required statistic from these repeated samples. The values of the statistic from the repeated sampling can then be used to generate standard errors and confidence intervals for the statistic like in our case.

In this project has been used function bootci in MAtlab, which is exactly constructs a bootstrap confidence interval. First step was Processing specifications (setting the maximal and minimal Growth Rate), than Processing capability and at last function gives us a confidence interval.

We receive a result with the:

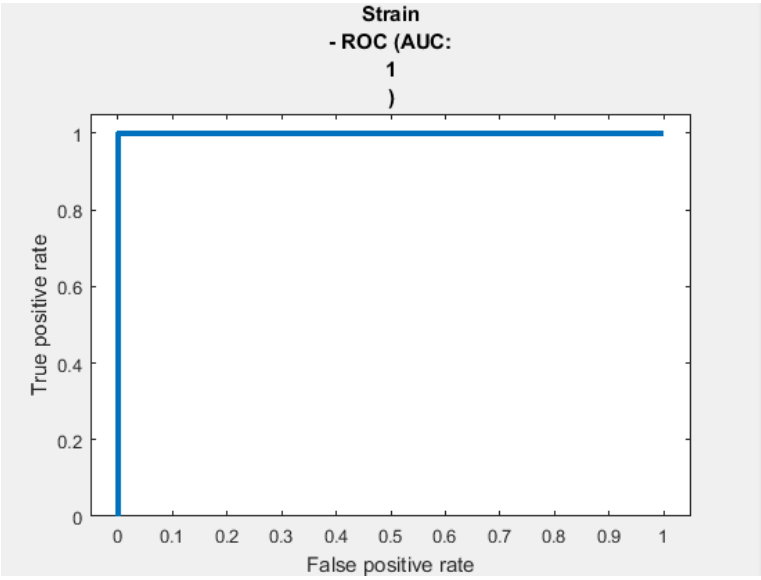
Confidence interval: 0.73142; 0.97752

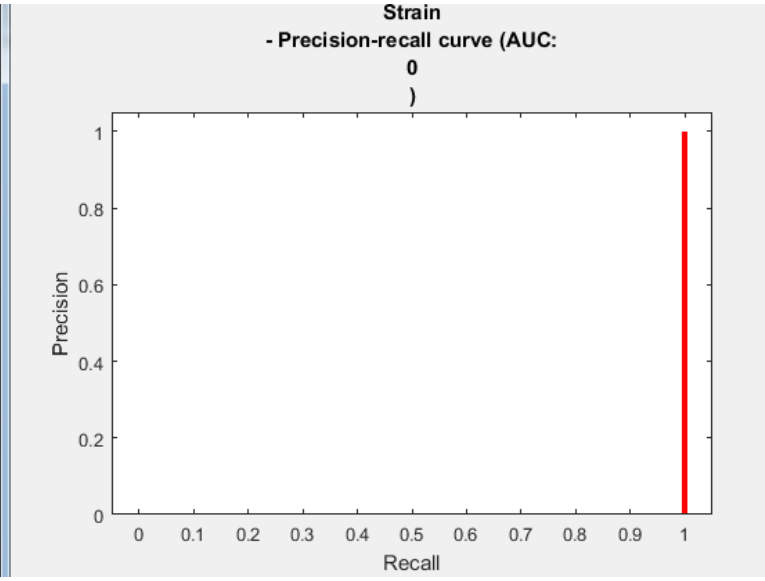
Third part. For a bacterium whose genes are expressed exactly at the mean expression value the predicted growth is ***0.38751.***

In **Fourth part** I created four separate SVM classifiers to categorize the strain type, medium type, environmental and gene perturbation, given all the gene transcriptional profiles. The classifier selects as features a small subset of genes by using non-zero weighted features from the regularized regression technique (lasso) of the first aim. The results for each classifier are presented below.

'Strain'

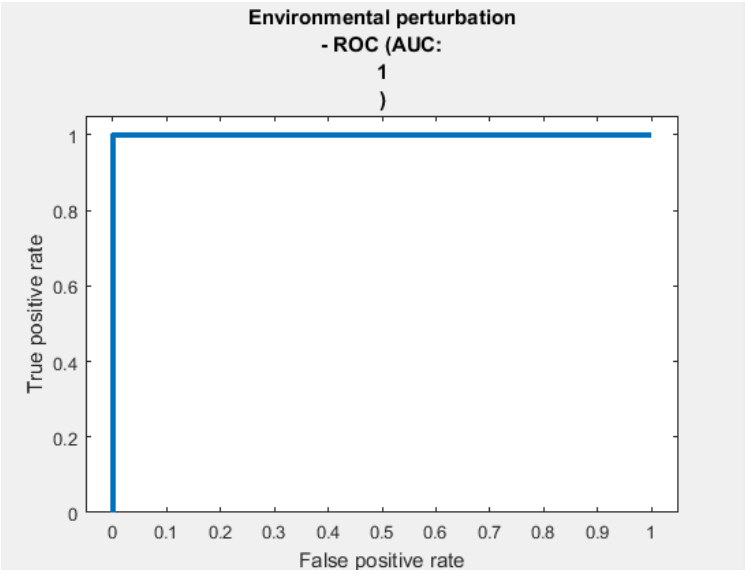
Cross-validation loss: 0.13402

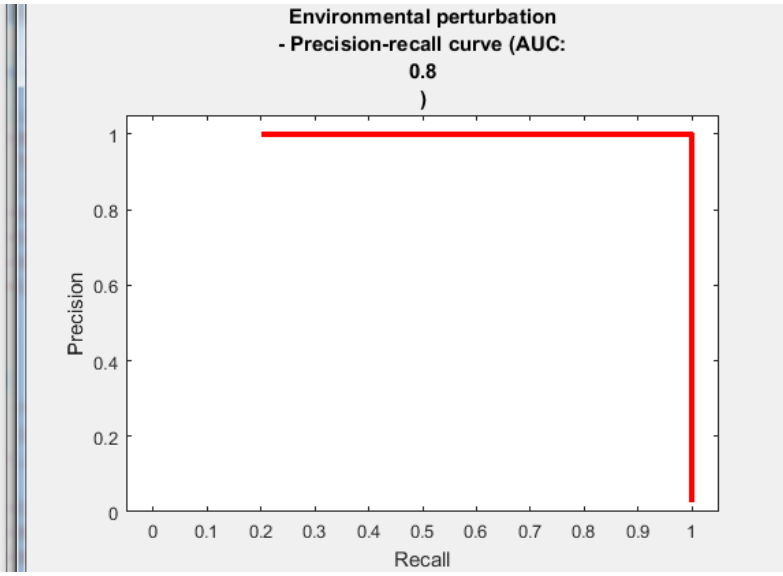




'Environmental perturbation'

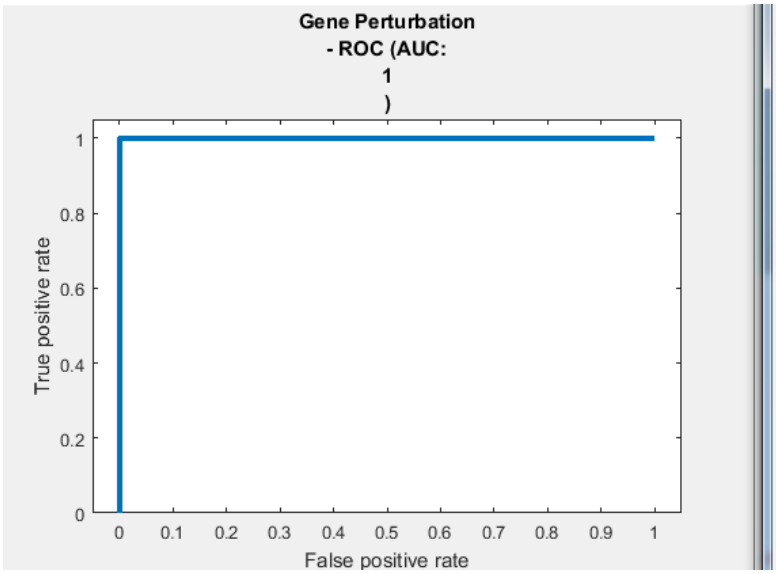
Cross-validation loss: 0.19072

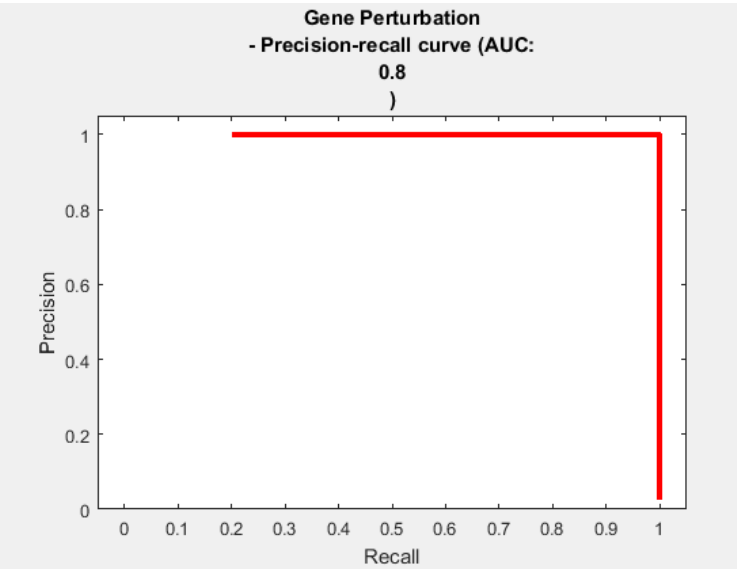




'Gene Perturbation'

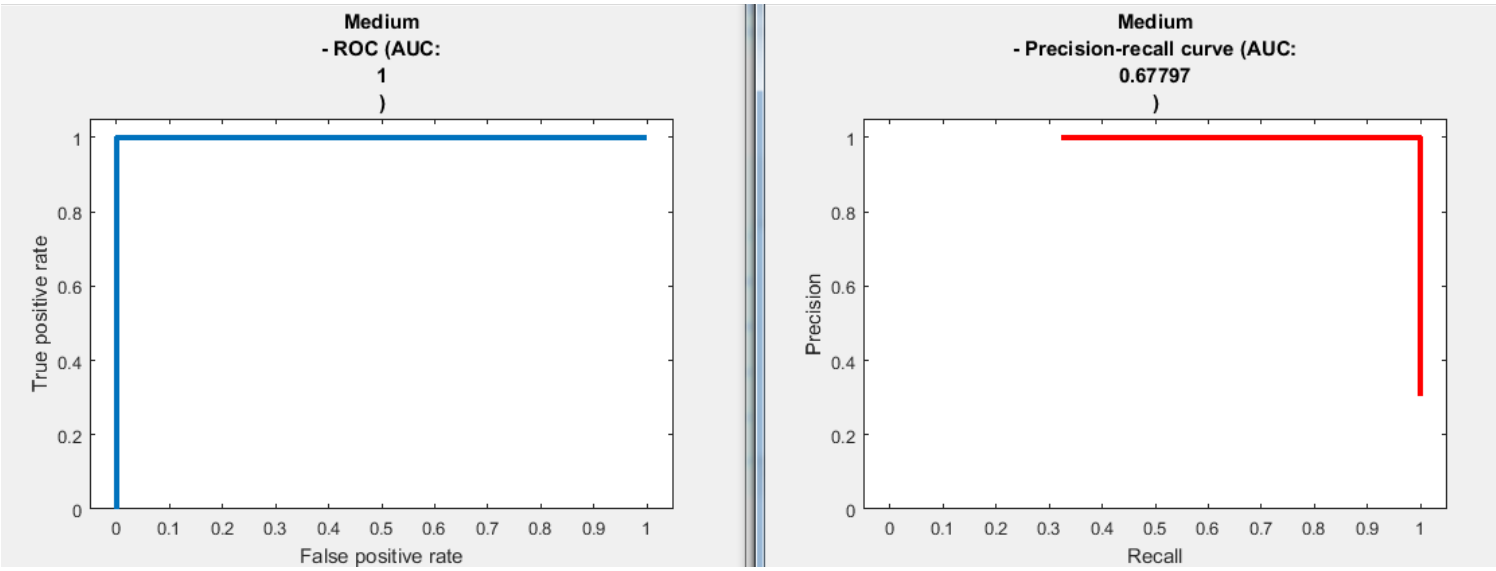
Cross-validation loss: 0.15464





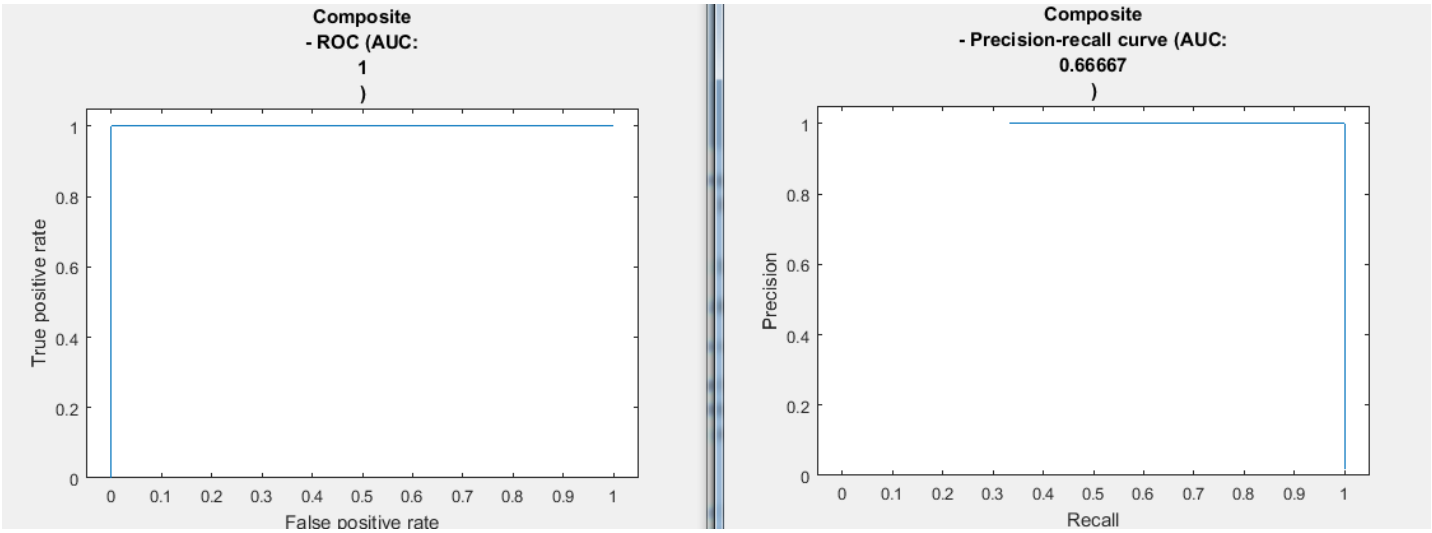
'Medium'

Cross-validation loss: 0.25258

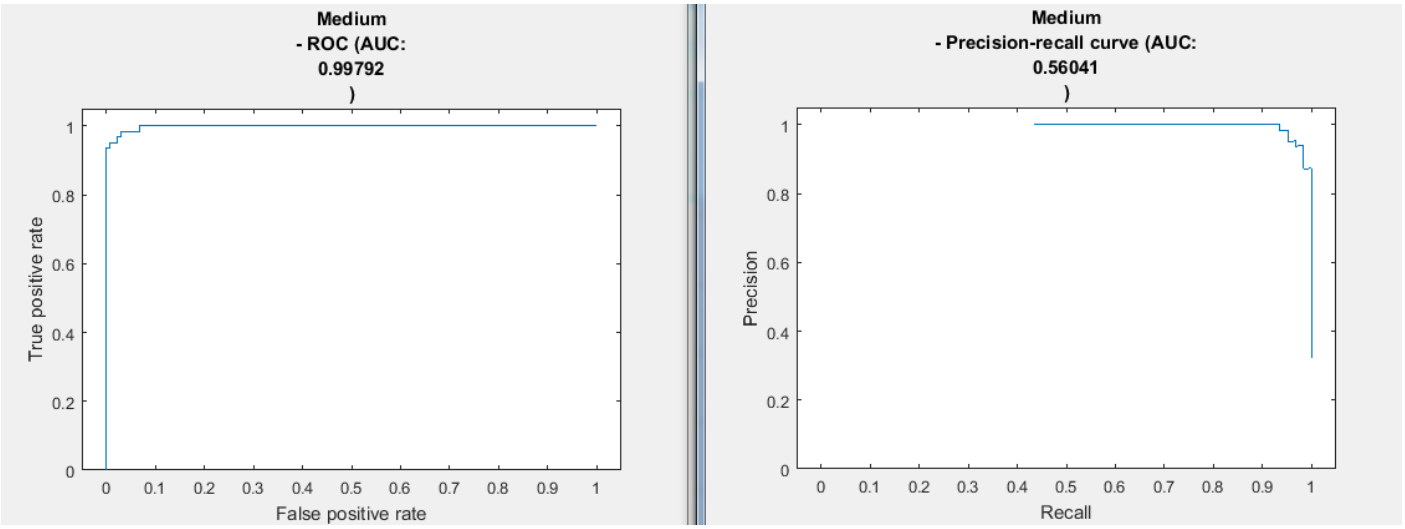


Part five. There was created one composite SVM classifier to simultaneously predict medium and environmental perturbations. The 10-fold cross-validation AUC/AUPRC value is shown below

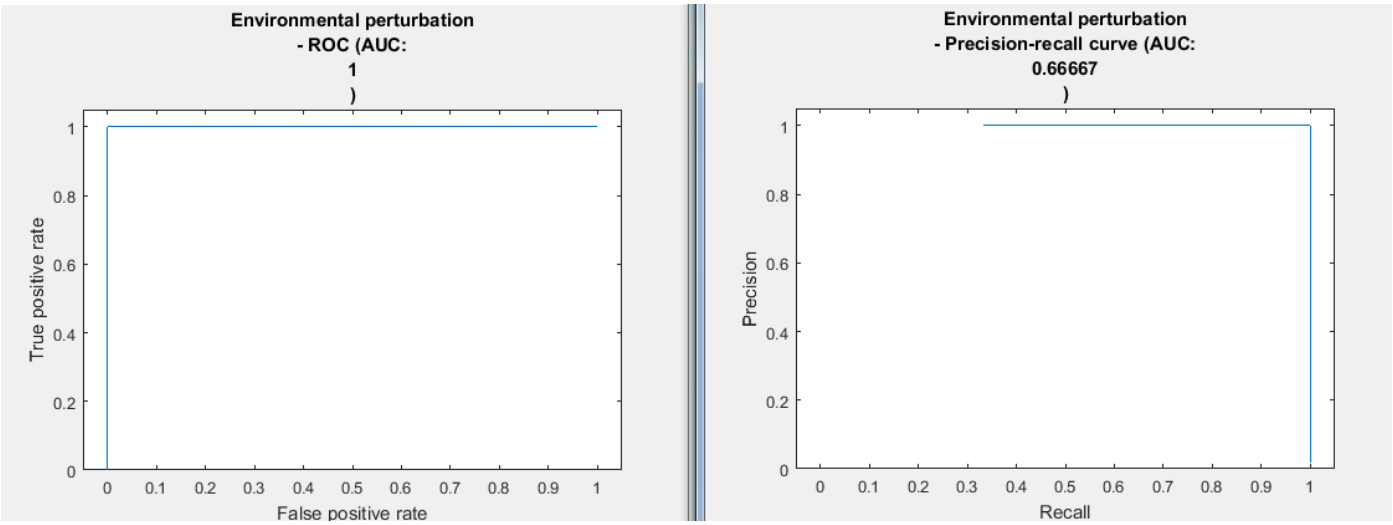
'Composite' Cross-validation loss: ***0.35567***



'Medium' Cross-validation loss: ***0.28866***



'Environmental perturbation' Cross-validation loss: ***0.18041***



This classifier performs worse that the two individual classifiers together for these predictions. So we are better off building two separate classifiers to simultaneously predict these two features.

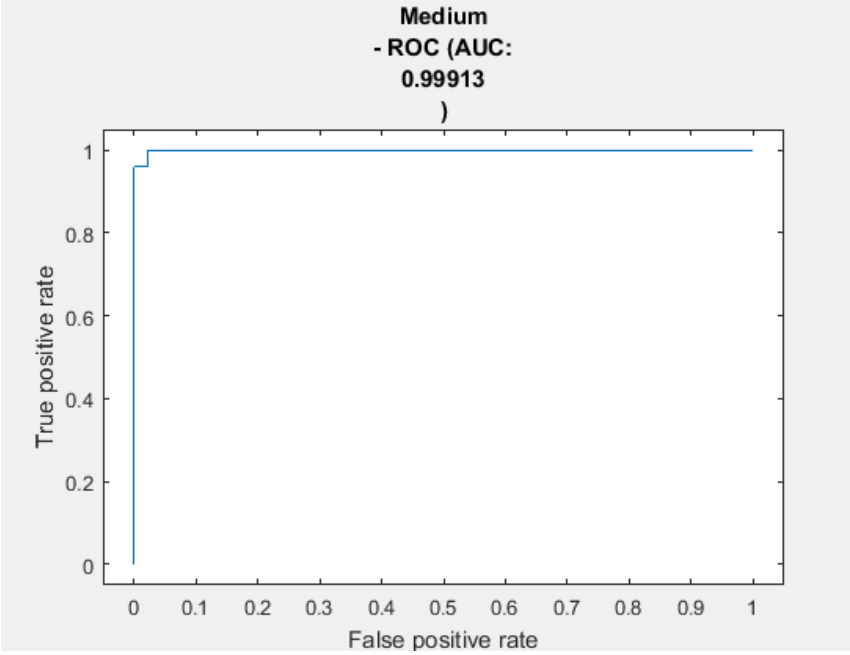
The work was to compare the speed and classification accuracy using a conventional classifier and a combination. That is, it was necessary to find out which method gives better accuracy.

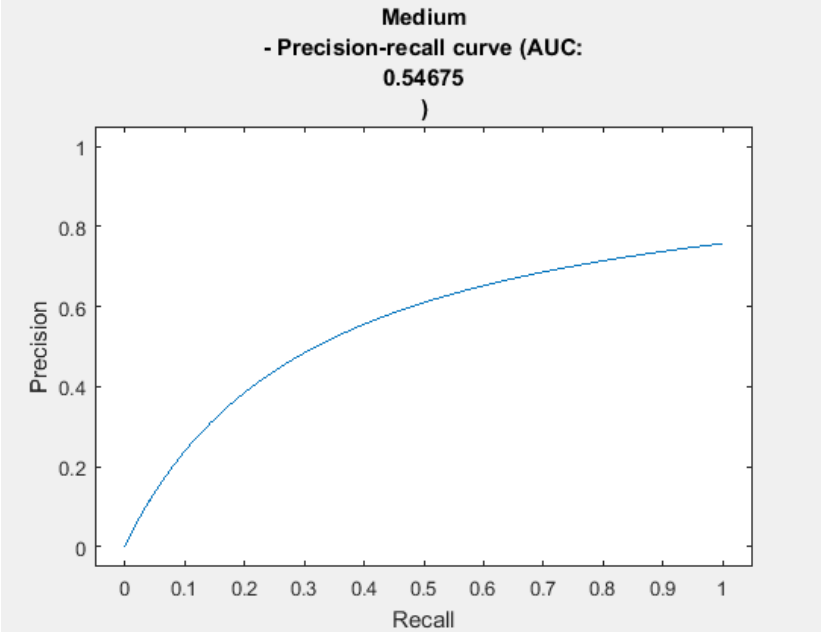
'Composite' 0.0722, Cross-validation loss: 0.36598
Single:
'Medium' error 0.0567, Cross-validation loss: 0.2732
'Environmental perturbation' error 0.0464, Cross-validation loss: 0.18557

So we see that error in composite method is bigger, but because the training takes place once for the two parameters, we get a speed boost.

Part six. Here is a performance of Principal Component Analysis, keeping only 3 Principal Components as features for the SVM classifier. The 10-fold cross-validation AUC/AUPRC value is

'Medium' Cross-validation loss: 0.63402





'Environmental perturbation' Cross-validation loss: 0.33505