REPORT Homework 4

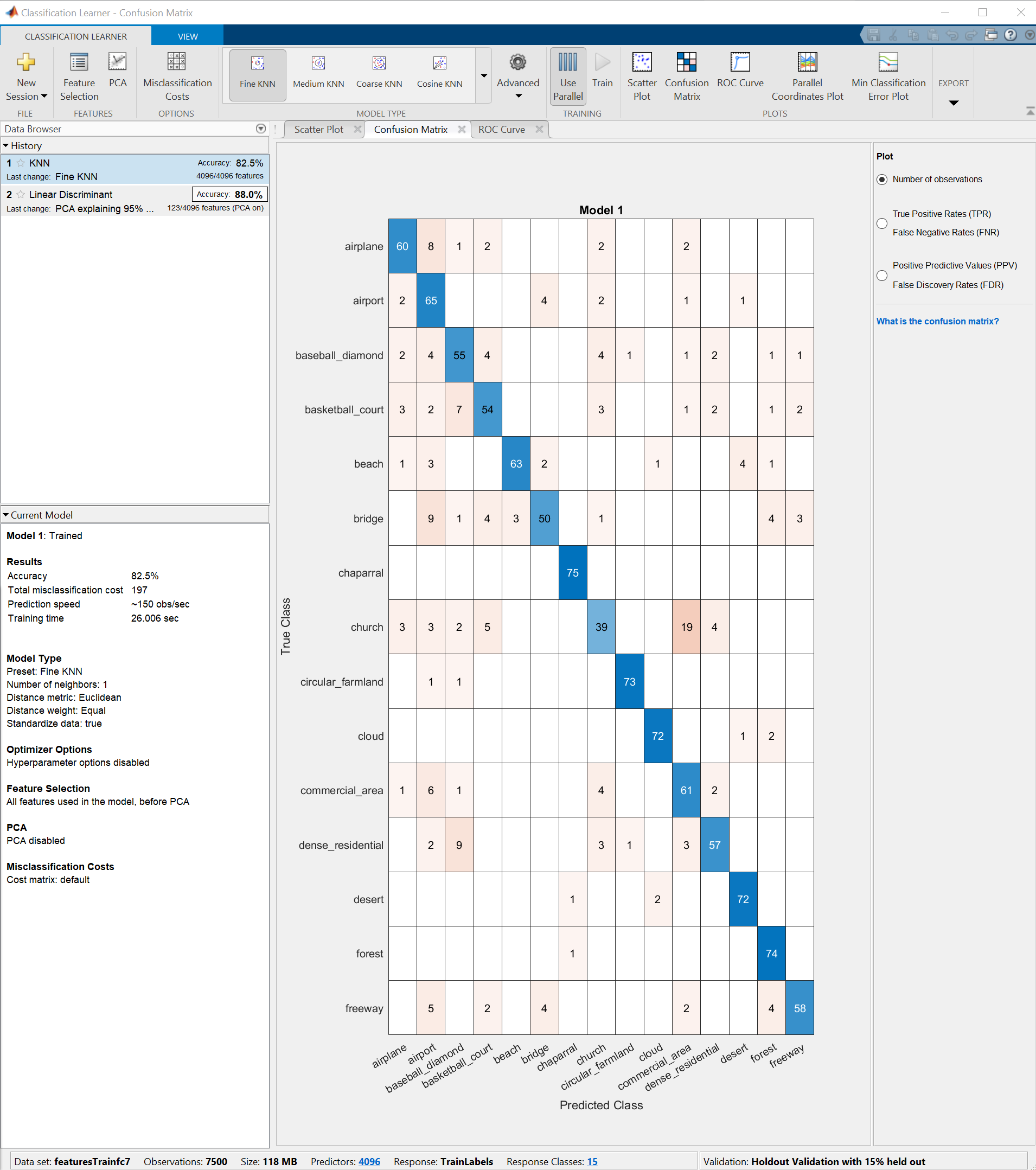
Vlad Dubrovenski

GitHub link to the project:

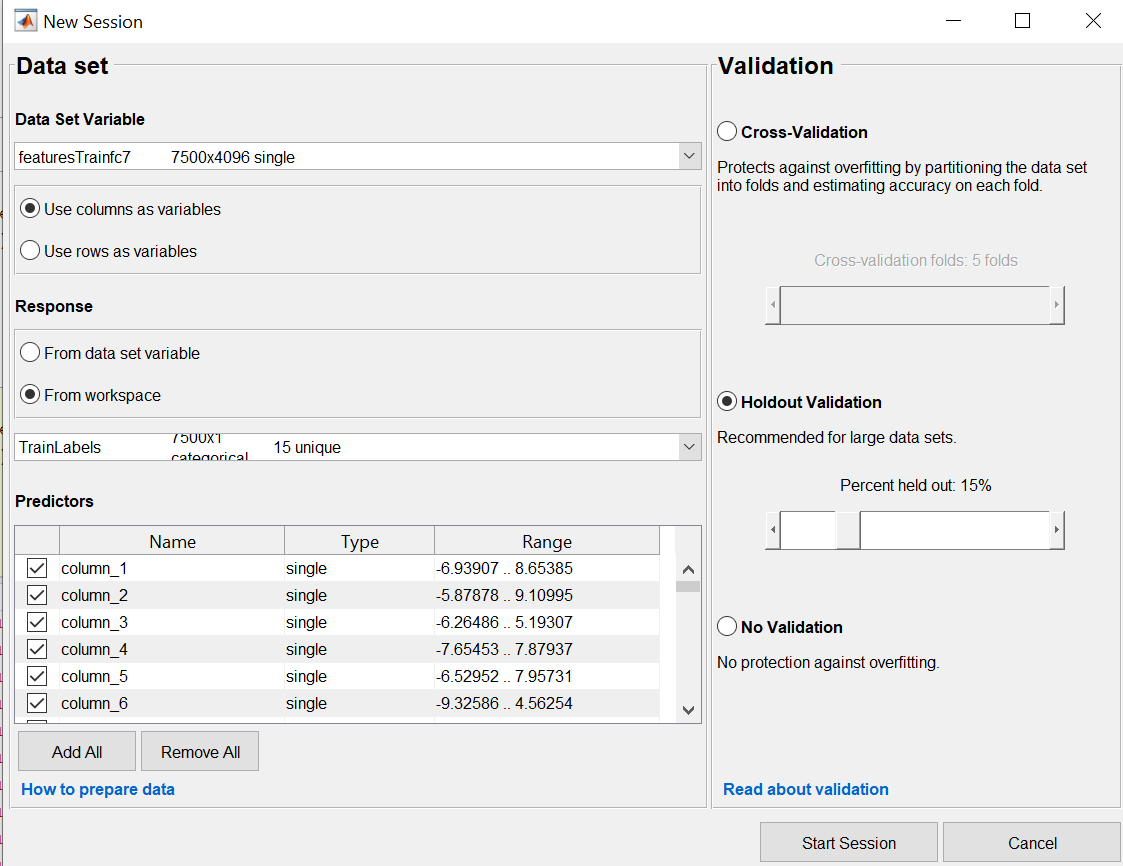
Homework-4: Classification of the remote sensing data set

1. Use pretrained VGG16 network FC features, find its low dimension embedding of fc1, fc2, and fc3 via PCA+LDA, and compute baseline accuracy with 1-NN classifier, plot the 100x100 affinity map also : (50pts)

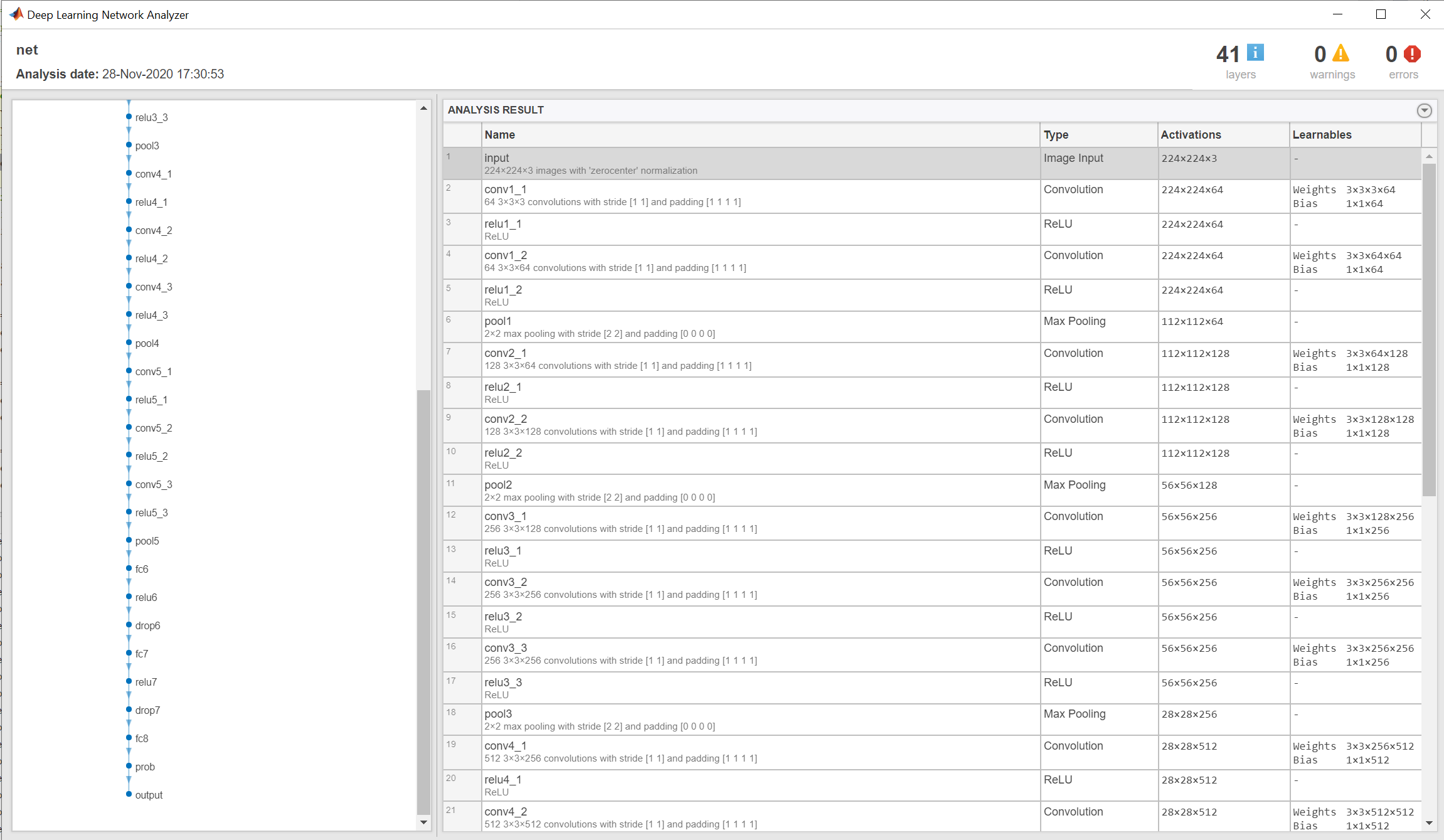
In this work, I explored the features of Classification Learner application of Matlab. With this application, I was able to train the classifiers for all the tasks, generate the code for the classifiers, as well as get the characteristics of such models. Below is the User Interface of this application.



As you may notice, I could just use the validation dataset directly from this application with holdout validation. I also was able to include the PCA with this app. That improved my development speed by a lot, however, I still had to modify the code of the resulting classifier to accomplish most of the tasks. Below is how to set the validation set to 15%. I did, of course, split the data in 3 sets as well with code, but I found that using GUI is a lot faster for development and lets me try a lot of possible classifiers.



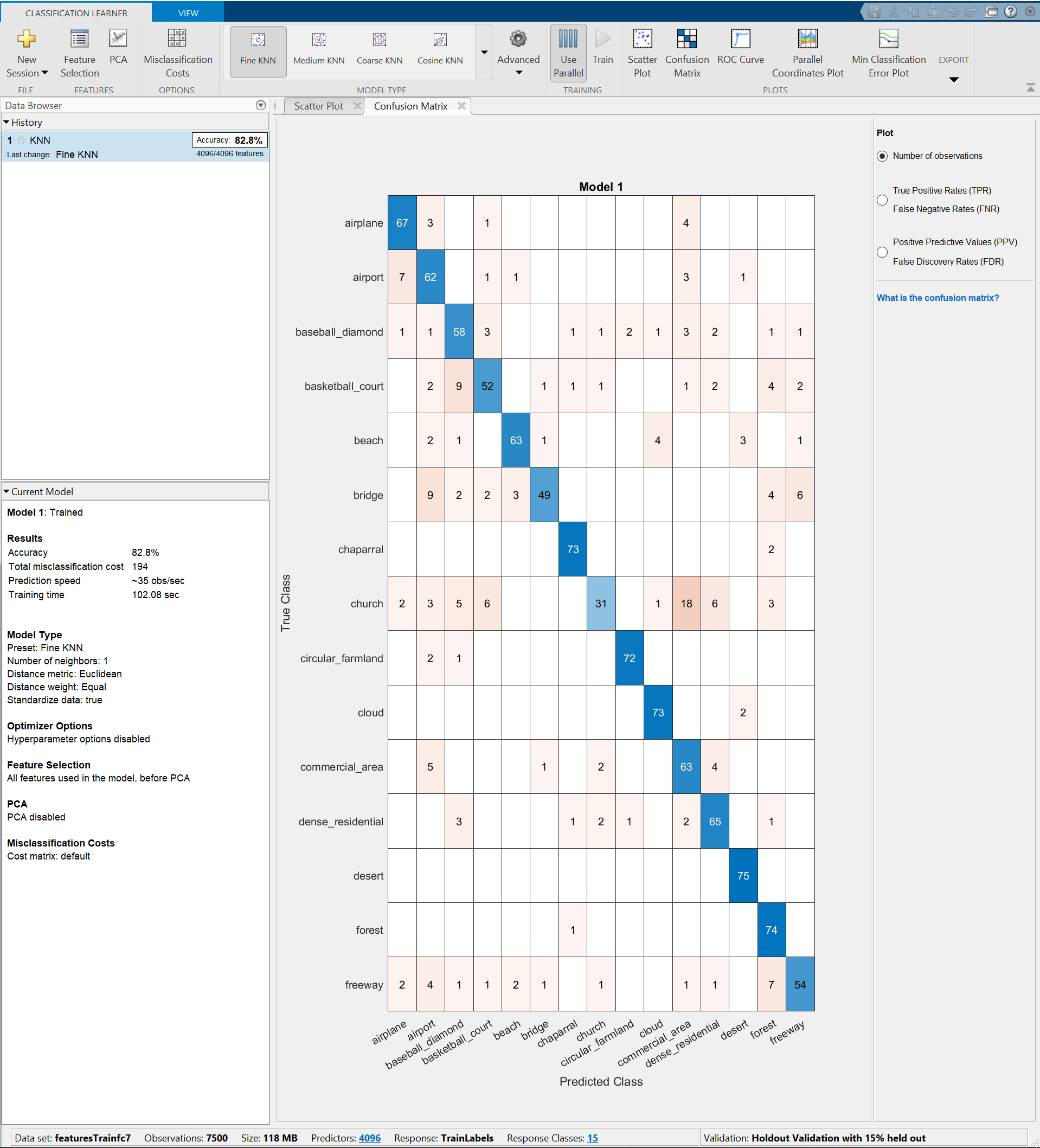
To accomplish the following, I used the deep learning network toolbox, VGG16 trained on image net, and standard functions the Matlab provides with version 2020b. Below is the scheme that I obtained with matlab analyzeNetwork() function which shows every layer and its properties.



Below are the model features for each FC and both the validation and testing confusion matrices. Also, I of course included the accuracy on testing set.

**FC1**

KNN Base Line Validation Accuracy + Confusion Matrix + other information

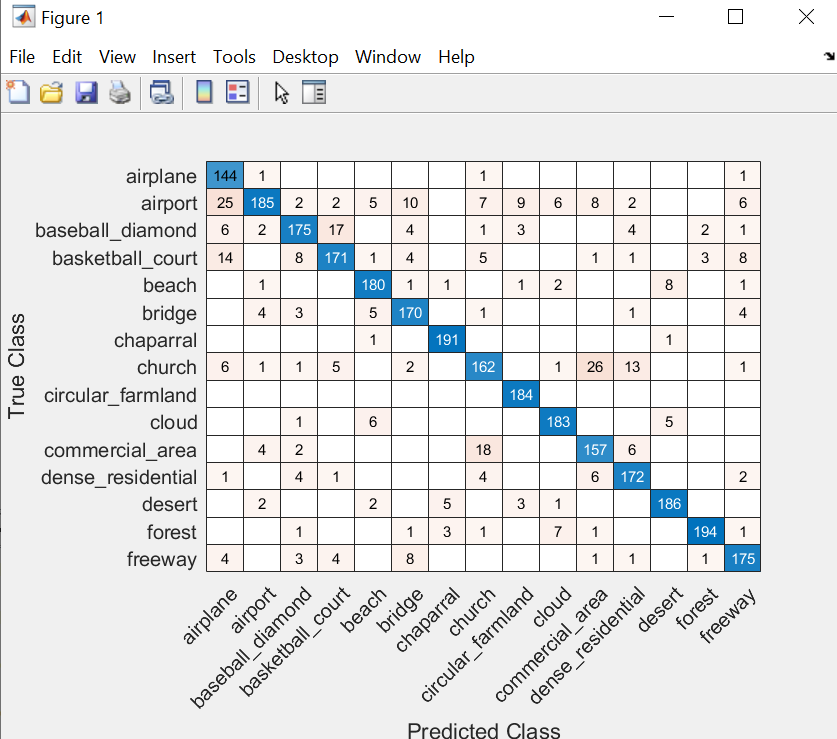


KNN Base Line Testing:

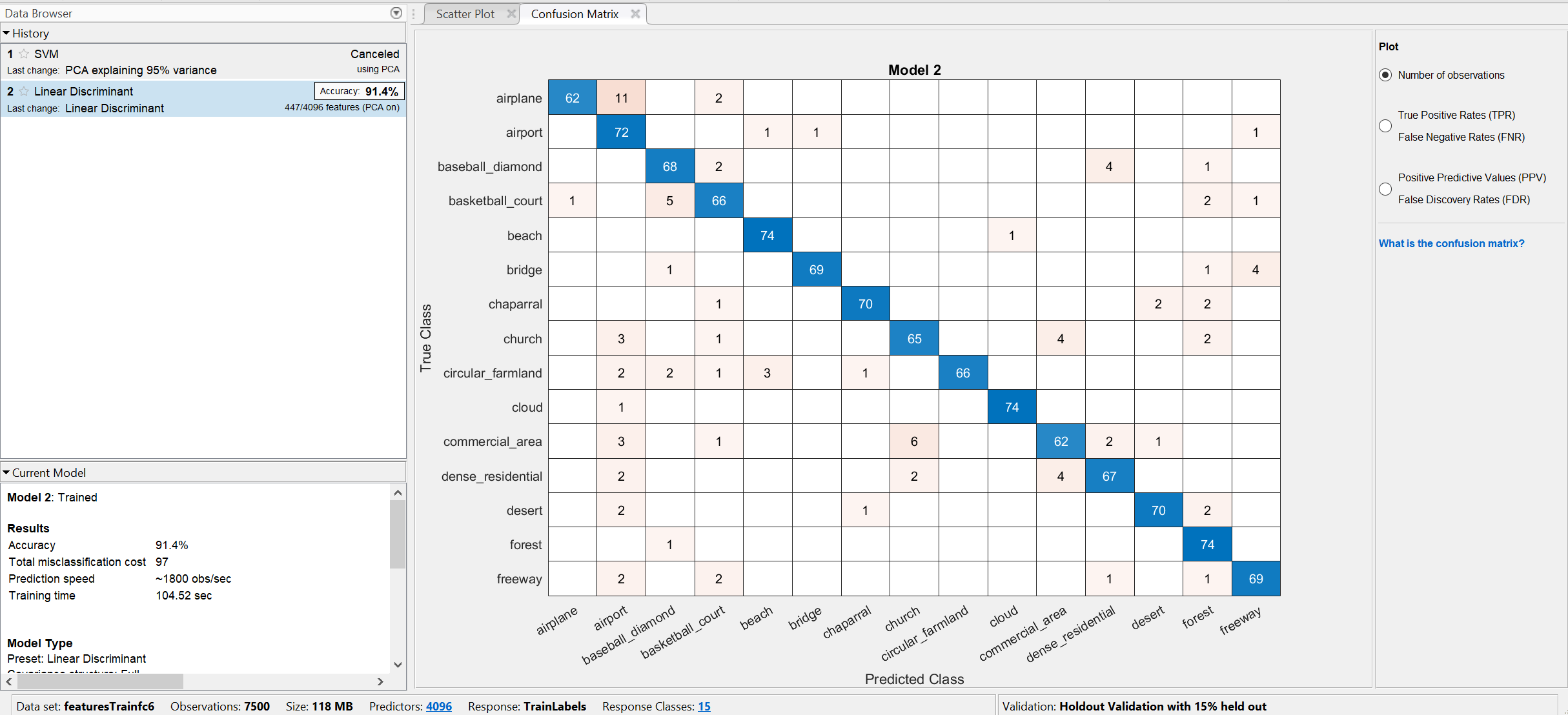
Accuracy:



Affinity Map:

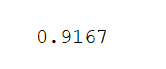


LDA+PCA Validation Accuracy + Confusion Matrix + other information

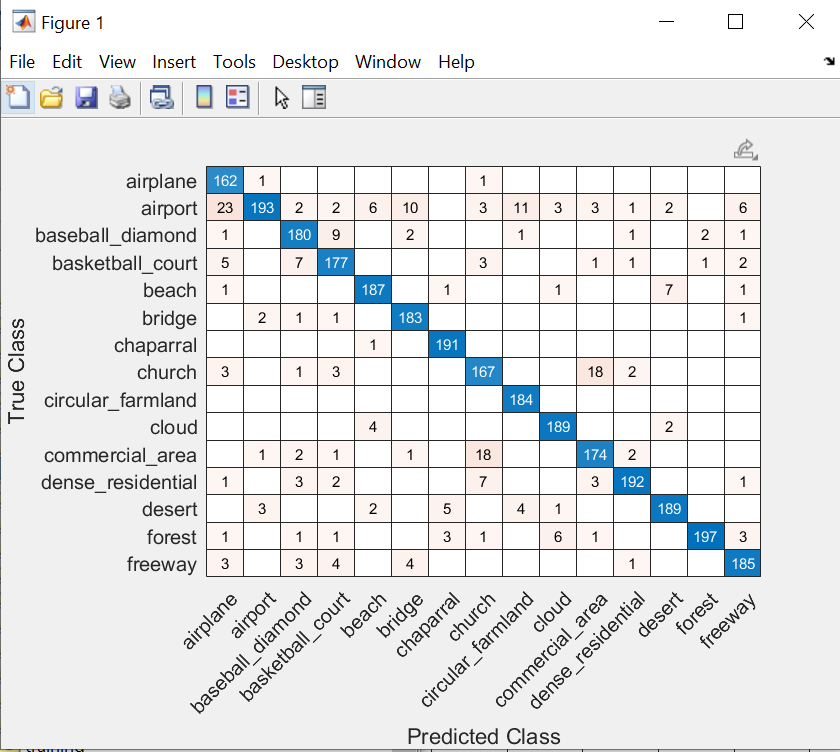


LDA+PCA Testing

Accuracy:

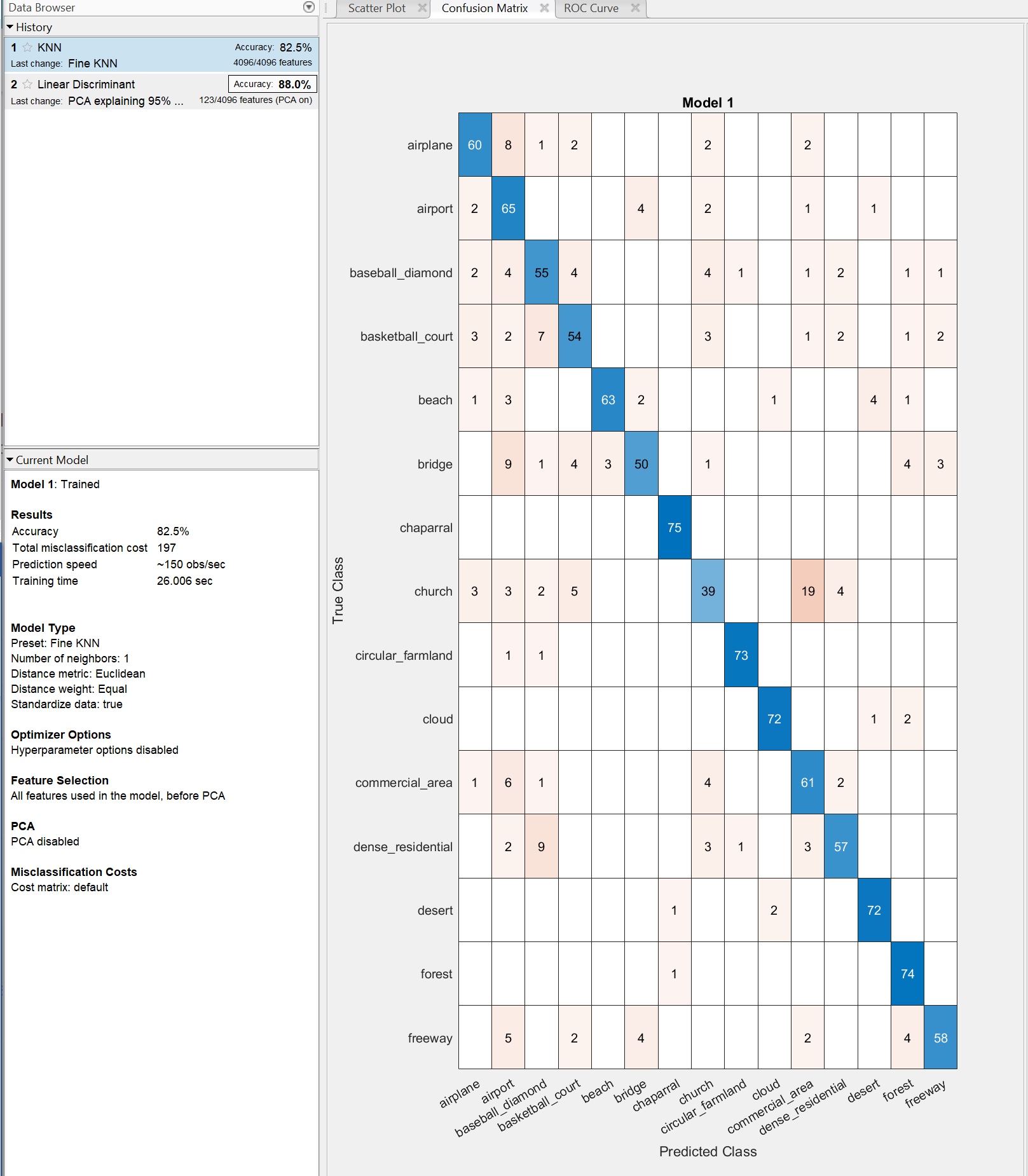


Affinity Map:



**FC2**

KNN Base Line Validation Accuracy + Confusion Matrix + other information

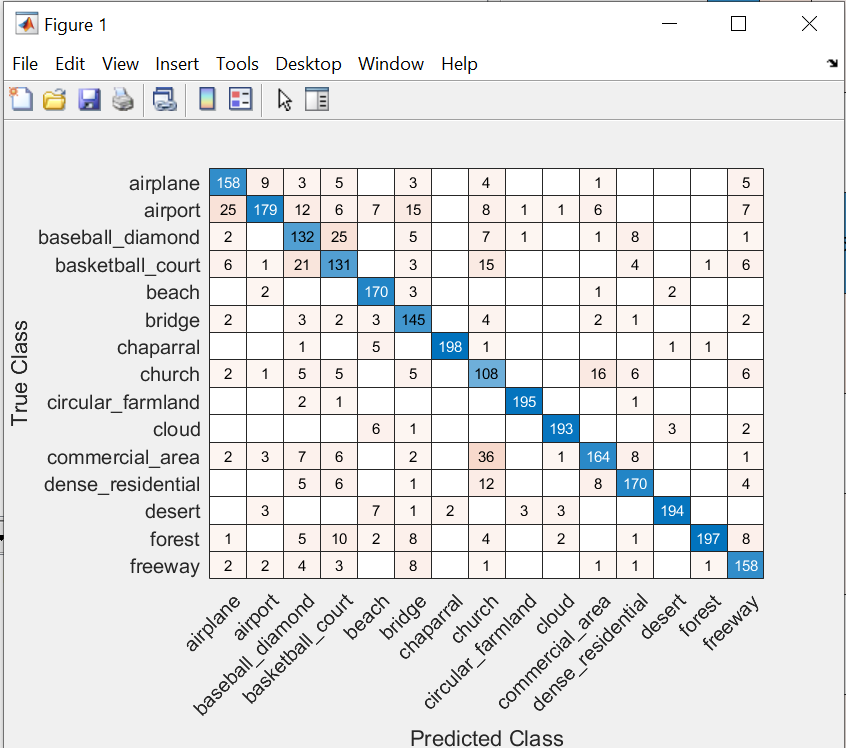
: 

KNN Base Line Testing:

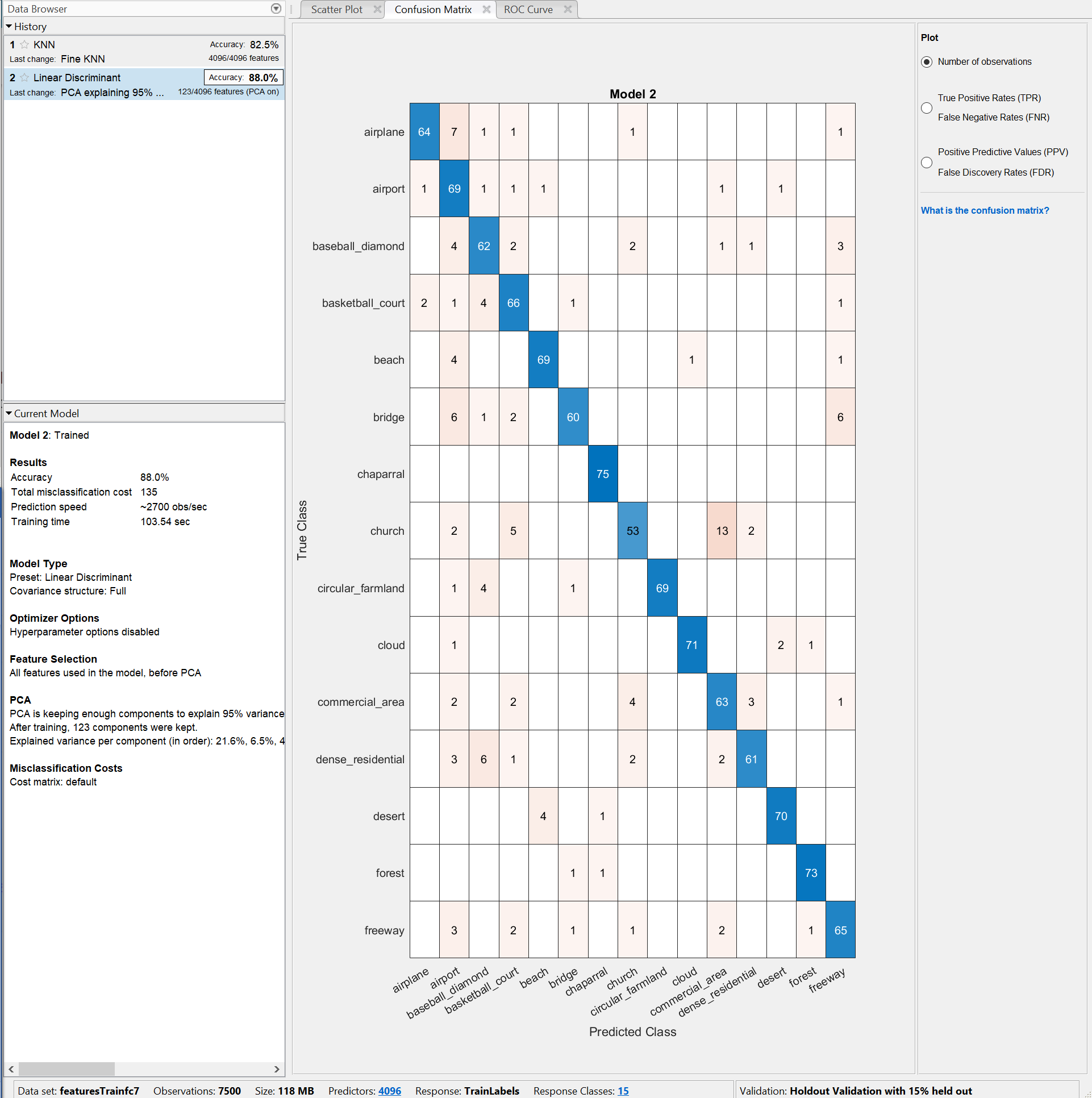
Accuracy:



Affinity Map:



LDA+PCA Validation Accuracy + Confusion Matrix + other information

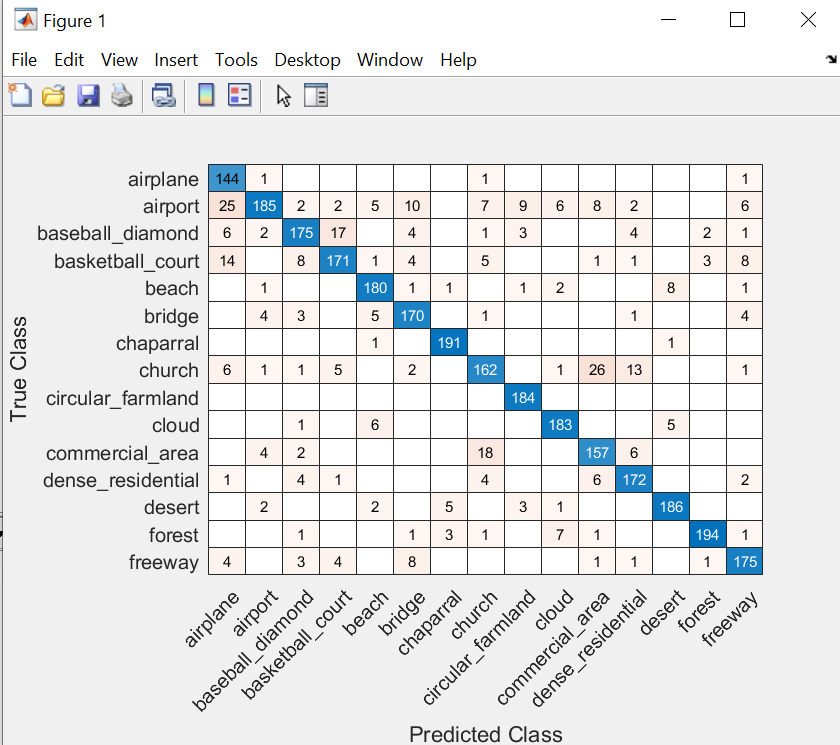


LDA+PCA Testing

Accuracy:

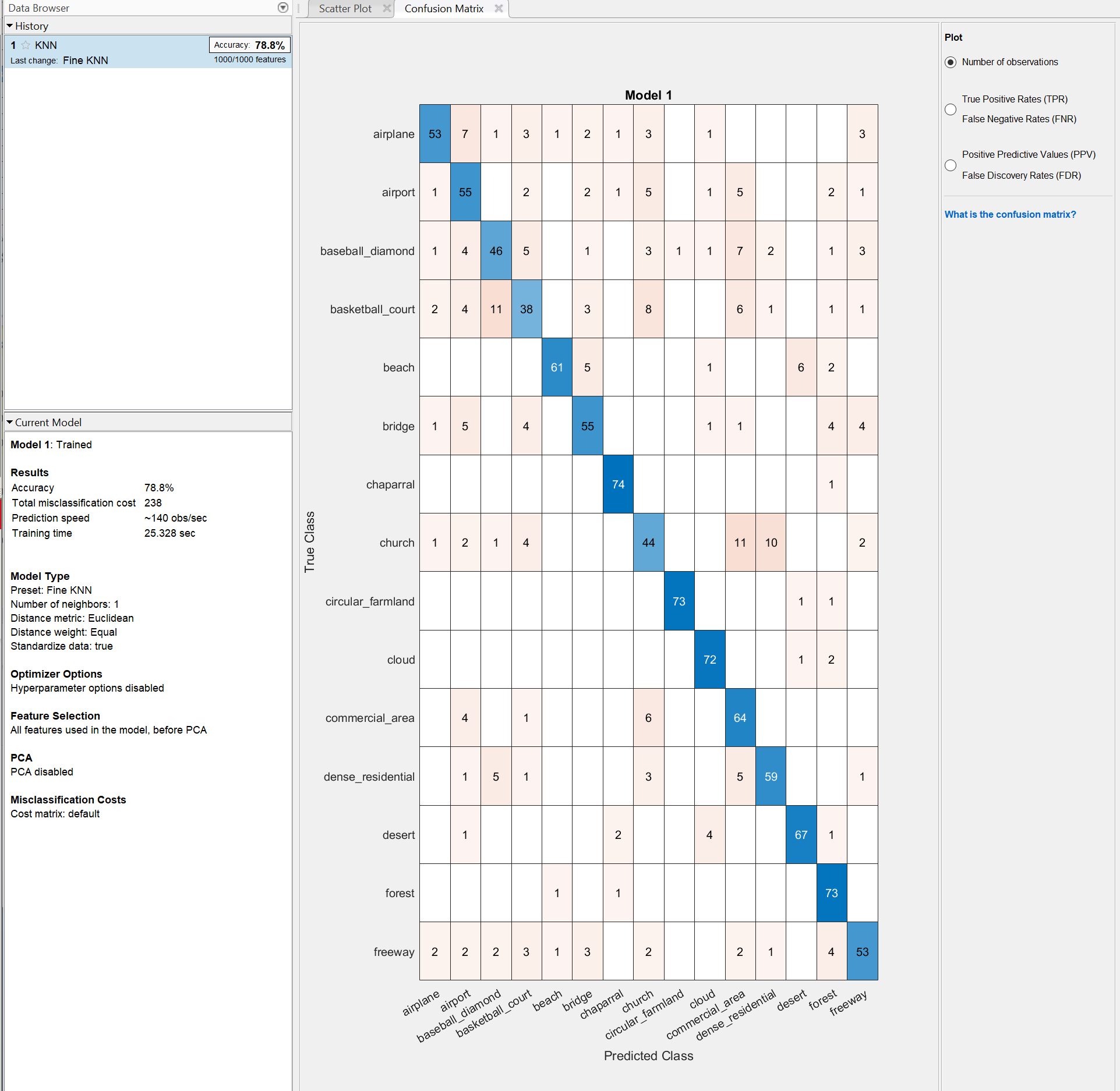


Affinity Map:



**FC3**

KNN Base Line Validation Accuracy + Confusion Matrix + other information

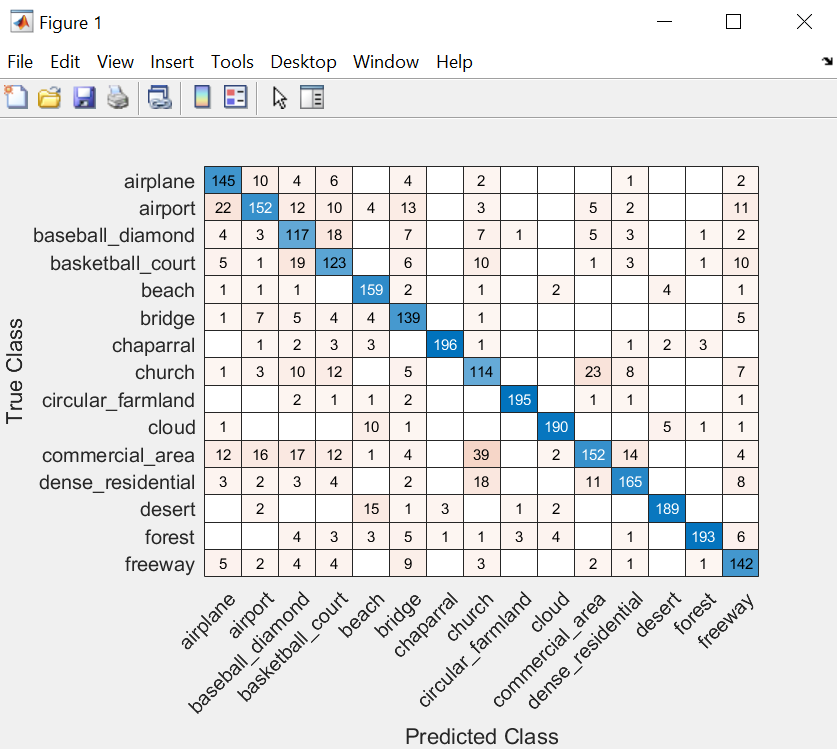


KNN Base Line Testing:

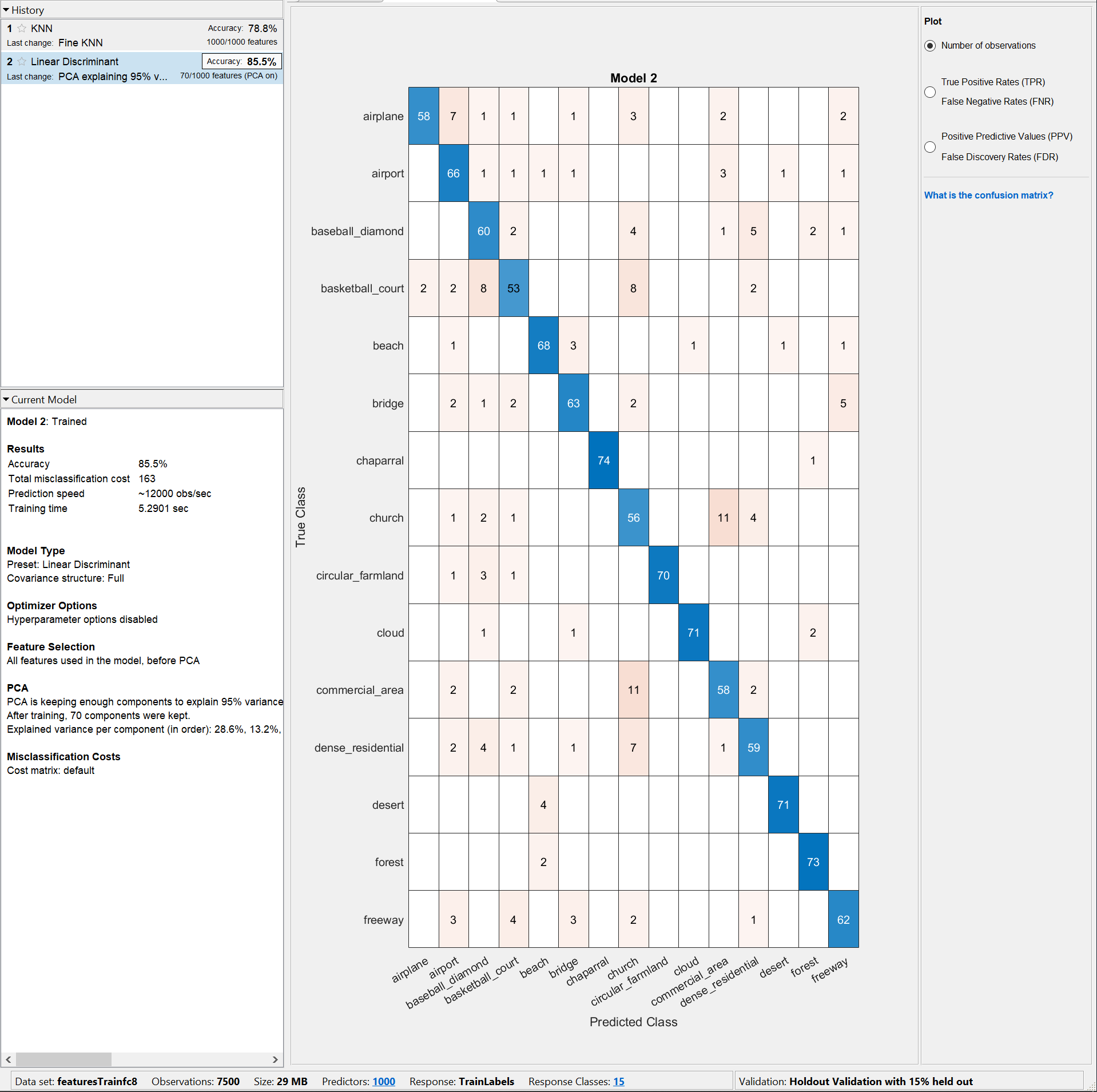
Accuracy:



Affinity Map:



LDA+PCA Validation Accuracy + Confusion Matrix + other information

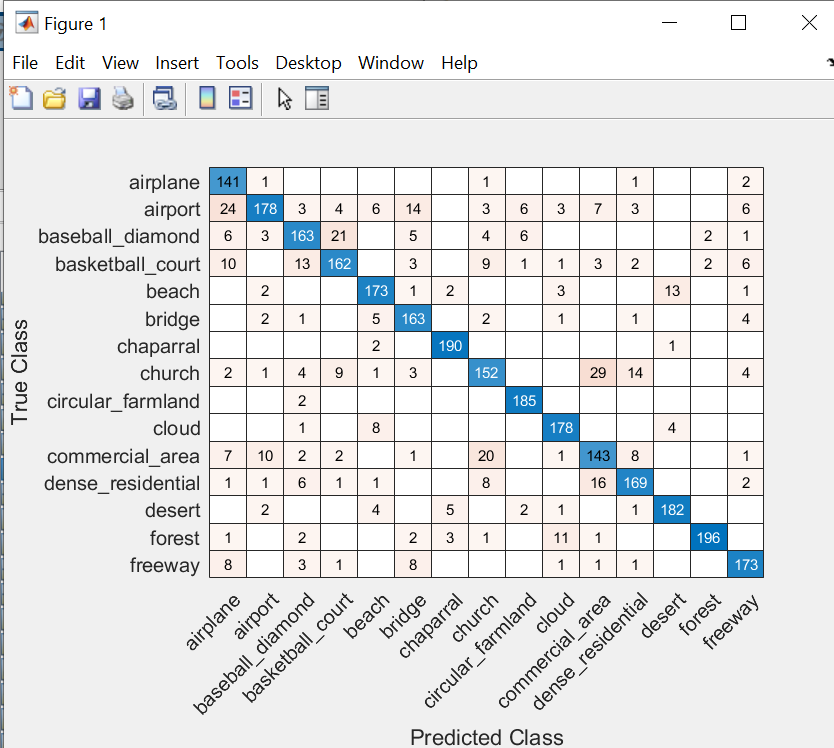


LDA+PCA Testing

Accuracy:



Affinity Map:



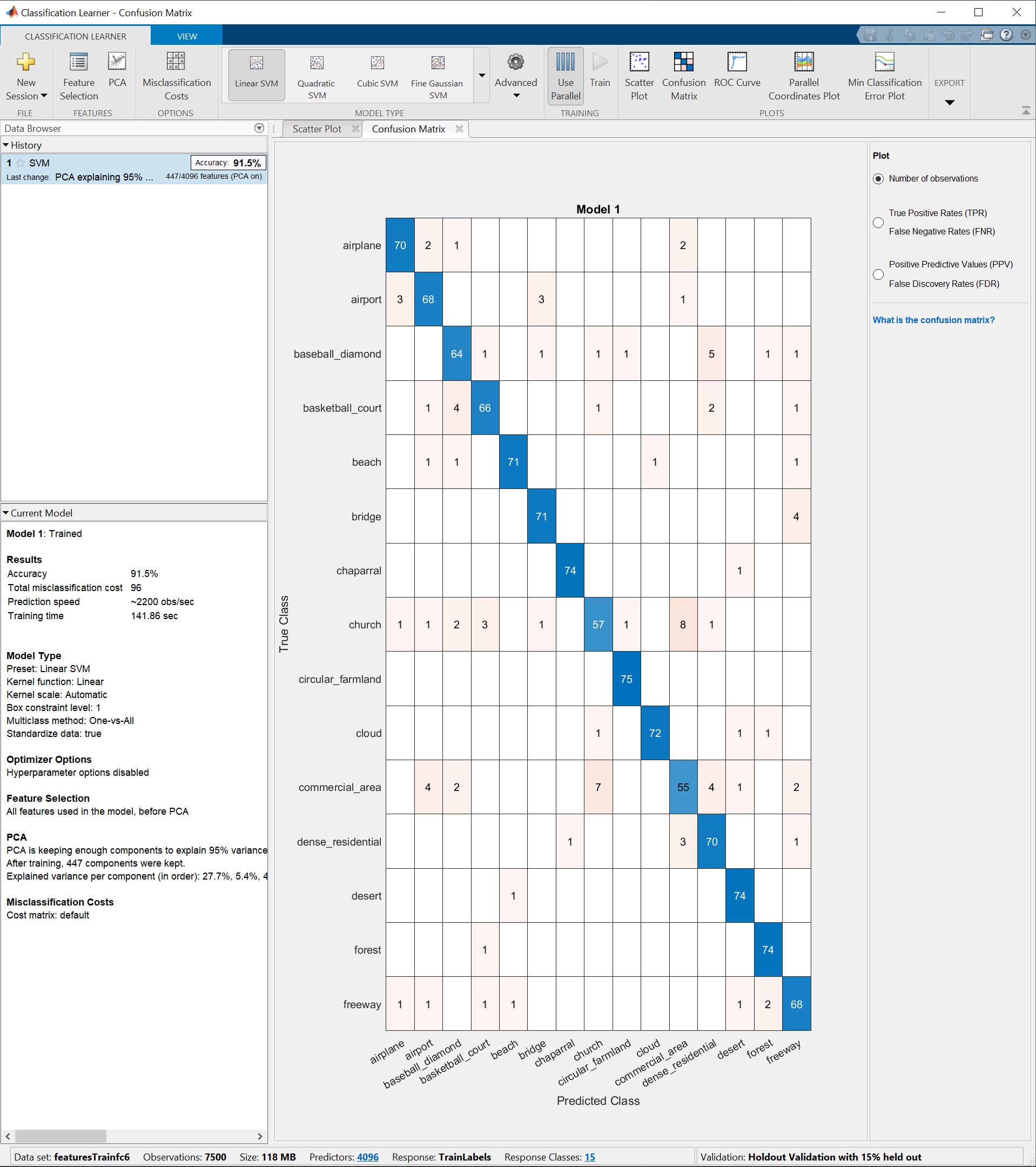
As we can see, the model with fc7 LDA+PCA performed the best.

1. Use (PCA +) Laplacian embedding and 1 vs the rest SVM and compute the top-1 accuracy for fc1, fc2 and fc3, find out which combination gives the best results. (50pts)

To accomplish this task, I again used the Classification Learner to speed up the development. I translated all the code as well. The was one vs all SVM, and I choose linear SVM with added PCA. With a slight modification, I was able to add the Laplacian as well. Below are the results.

**FC1**

Validation:

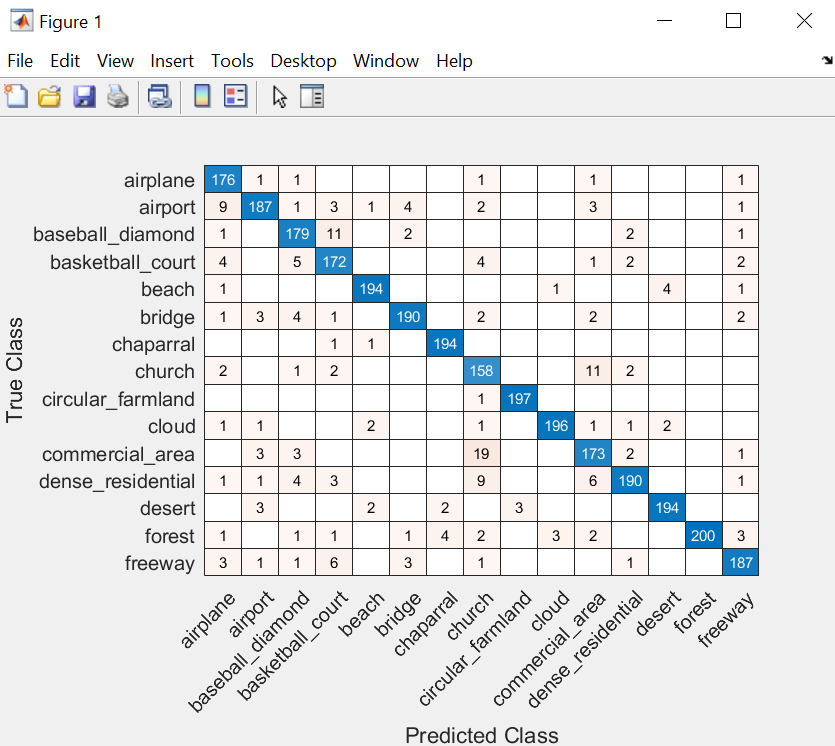


Testing:

Accuracy:

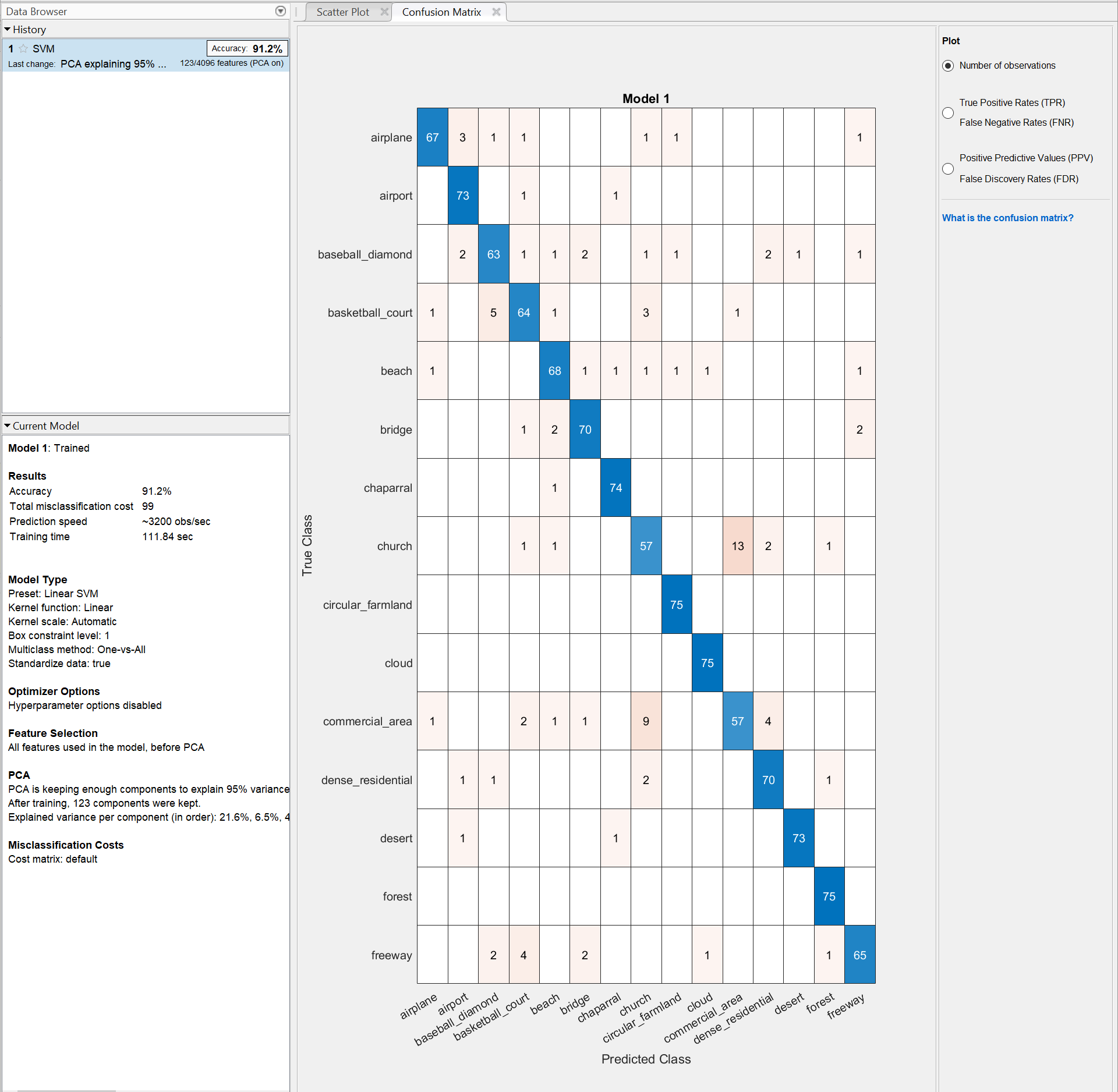


Affinity Map:



**FC2**

Validation:

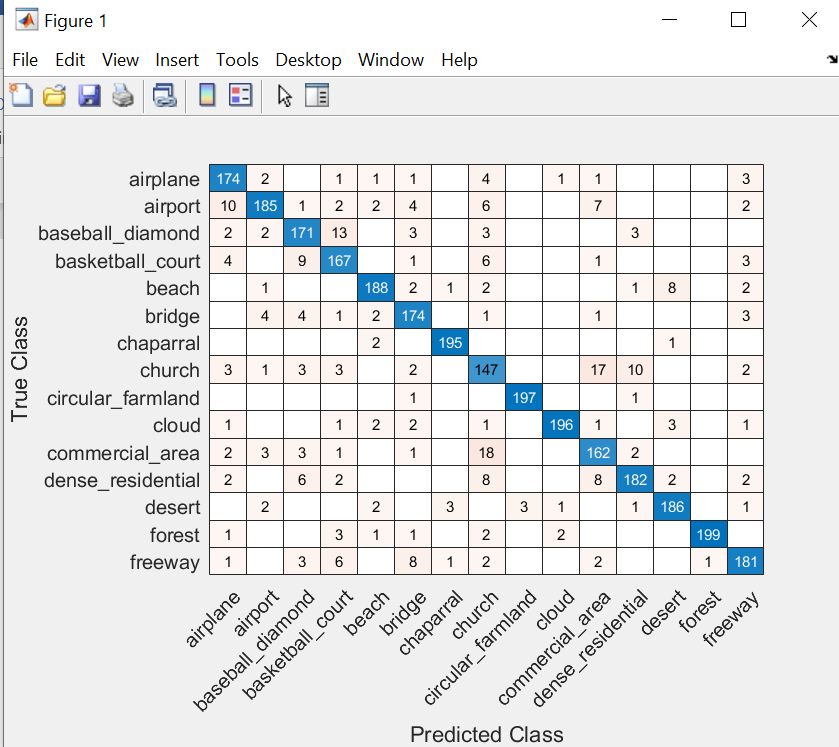


Testing:

Accuracy:

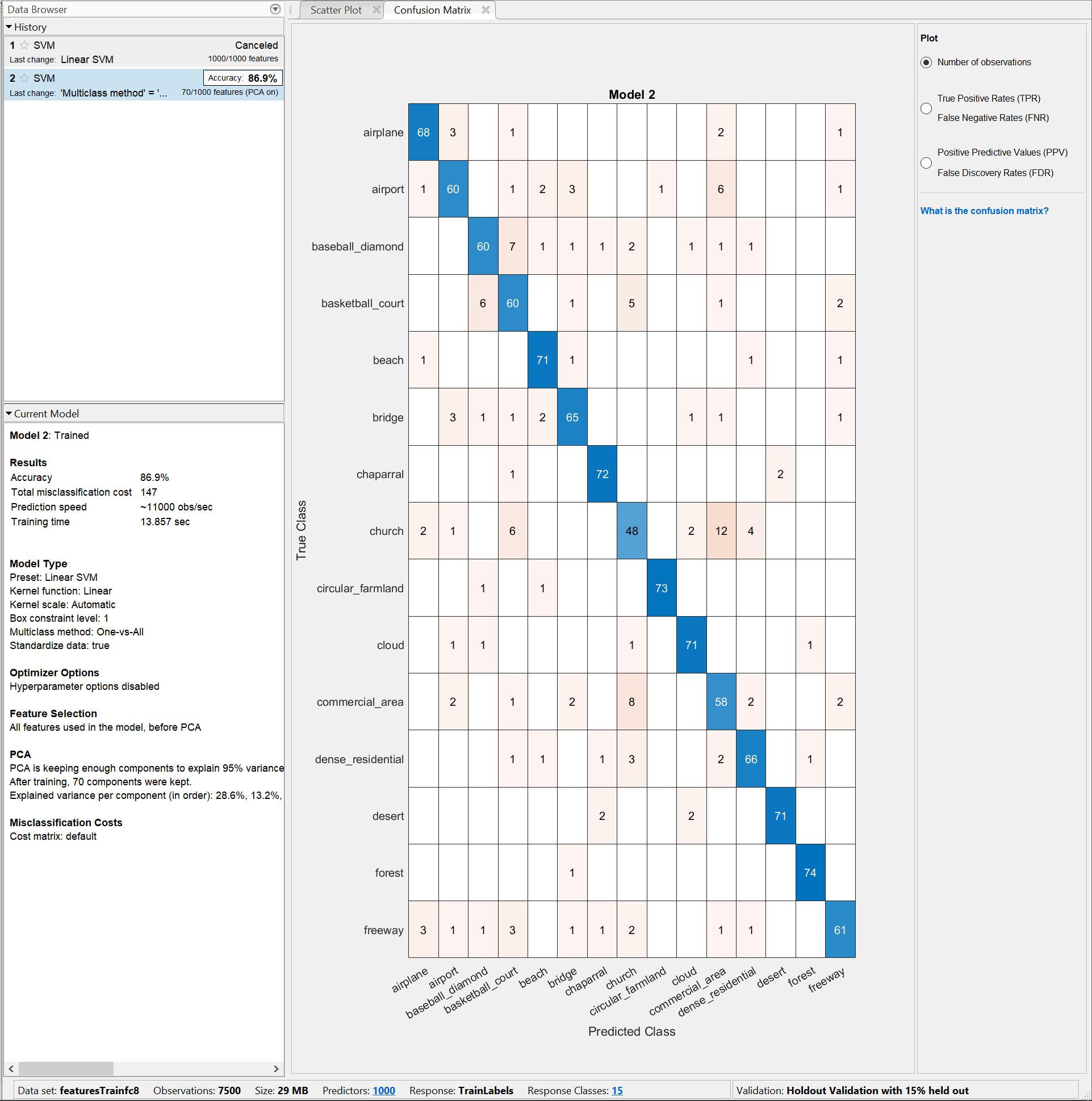


Affinity Map:



**FC3**

Validation:

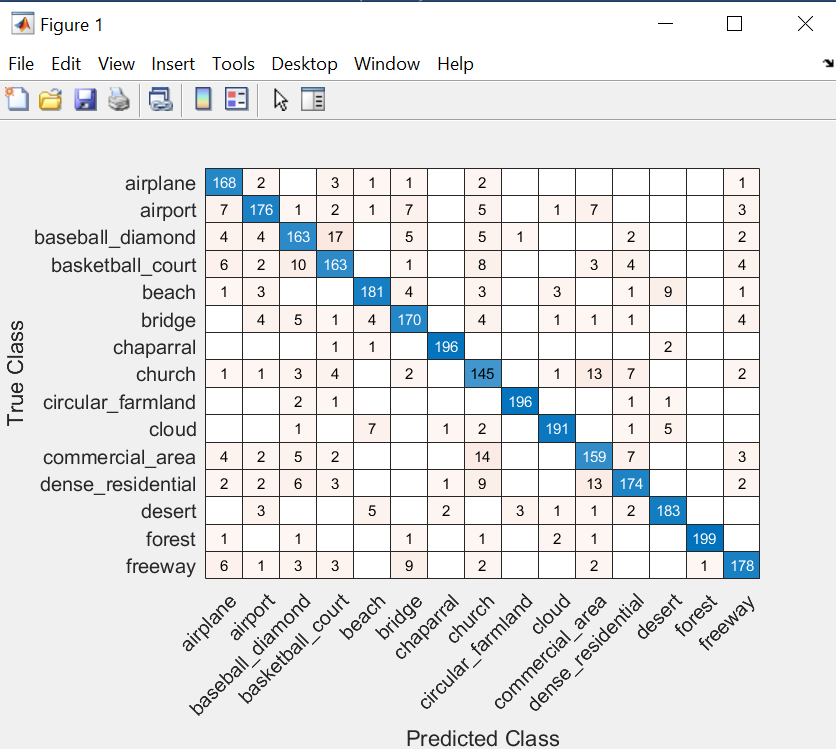


Testing:

Accuracy:



Affinity Map:



As we can see, SVM+PCA with FC1 showed the best result out of all competitors.

Running instructions:

Requirements: Matlab 2020b, Deep Learning license, Statistics license, and discrete video card is preferable.

1. Run the splitter.
2. Import vgg16 pretrained. Most likely, you don’t have that model downloaded, so just use the link from the error message. After that, just click on it and follow downloading instruction. You might need an account, it doesn’t have to be attached to a license (gmail worked for me). You may see the architecture to make sure it is the correct model.
3. Extract the features (need some toolbox license, not sure which one, most likely DL again).
4. Run models to see the results of testing. You may also print the validation scores if you would like.