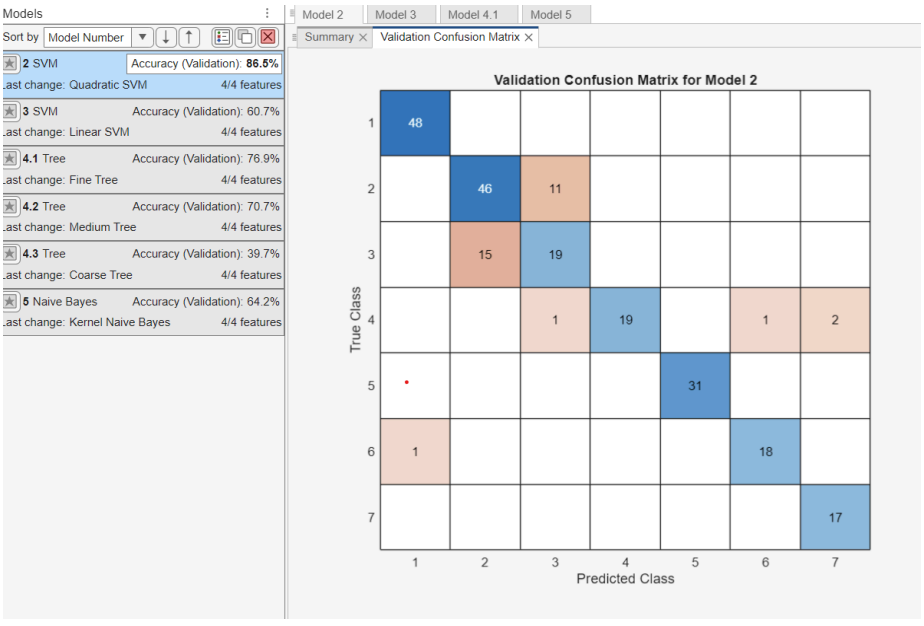
Annika Hoag

COM 322 - Final Write-Up

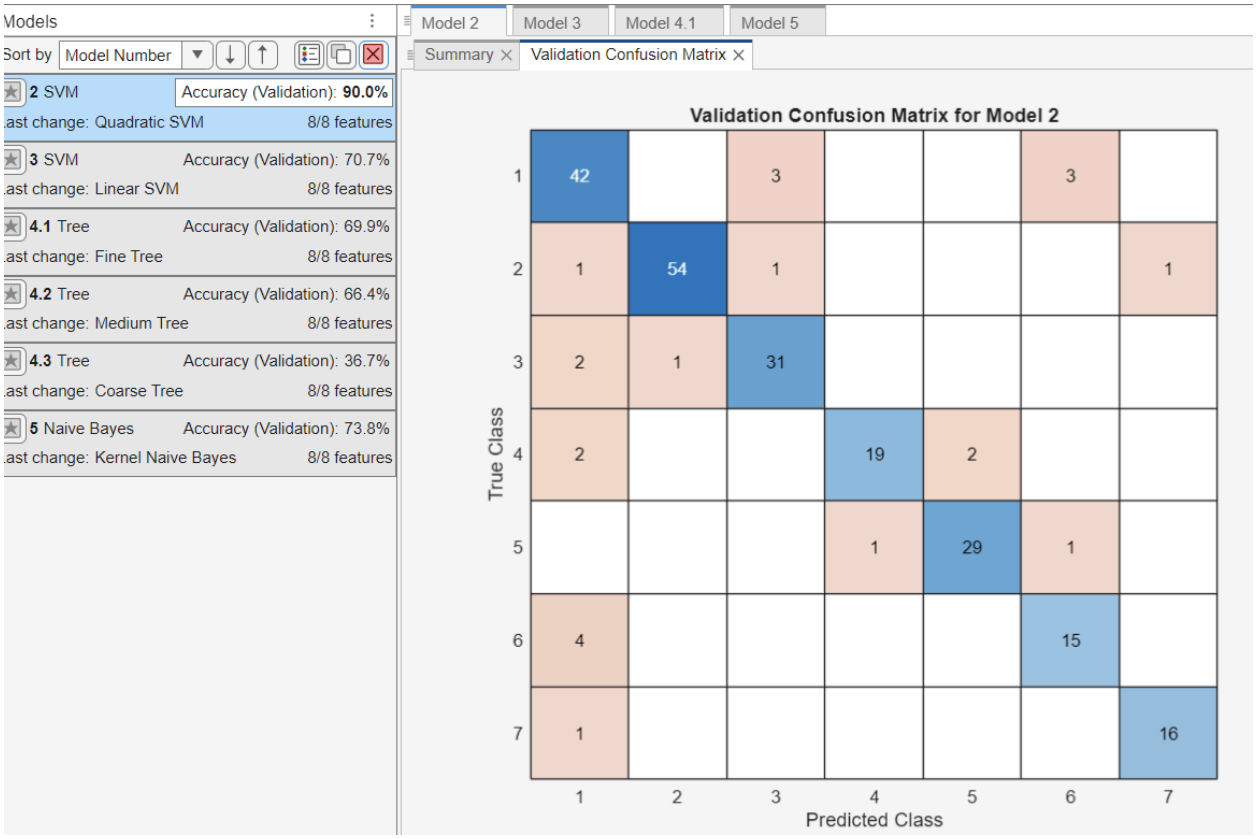
December 16, 2024

For my Final Project I am classifying seven ballet poses from images in a dataset that I created. There were two main methods that I used, OpenPose's joint position identification and MATLAB's Classification Learner app. The first step in my project was installing OpenPose and learning how to read the JSON files that OpenPose writes the joint position keypoint coordinates and confidences to. I checked my understanding by plotting the written coordinates onto my images in MATLAB. Then, I used MATLAB's plot function to determine what I would use as my features for classification. I decided to try examining the distance from the Mid Hip joint to both wrists and both ankles. I plotted these distances as lines in a graph to see if there was a distinguishable difference in the relationship between these distances for each pose. I also tried the distance between both ankles as an option. I ultimately determined that two of the poses were too similar and combined those two into one class.

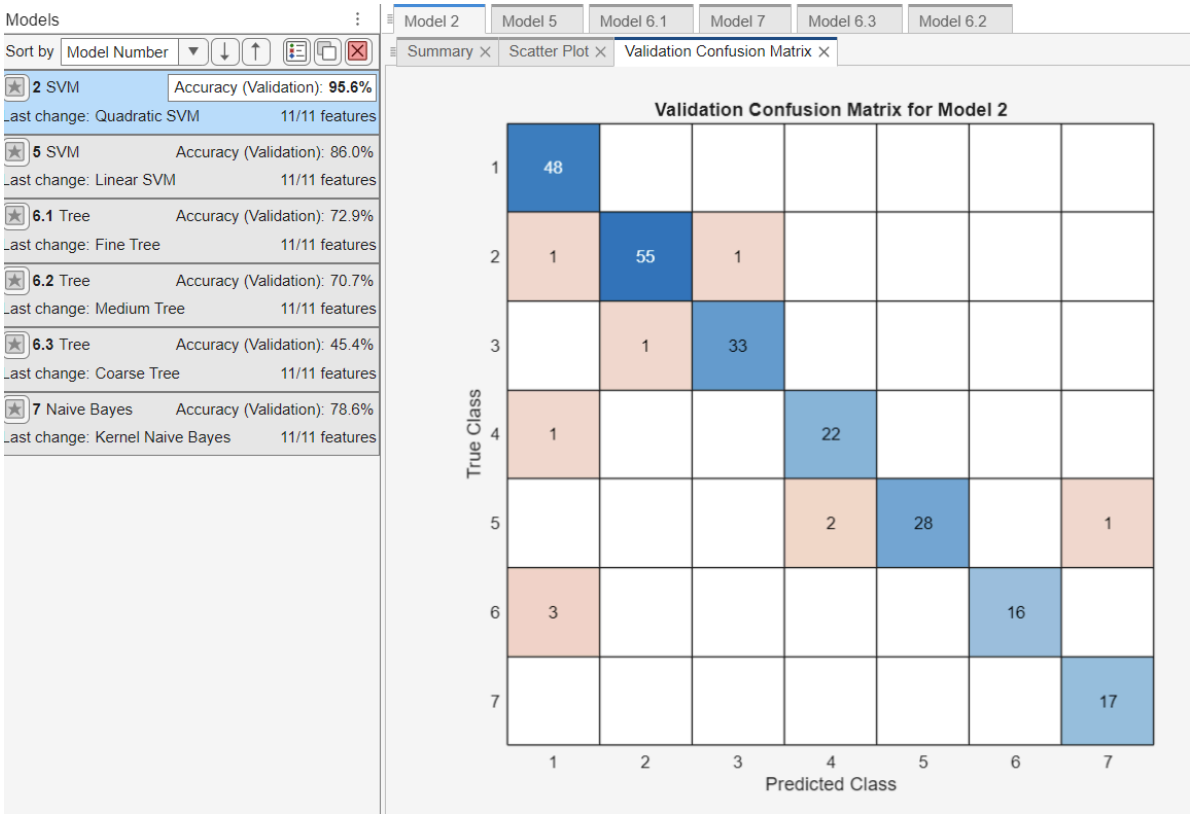
Once I had these features in mind, I created a matrix where each row is an image, and each column is a feature (Euclidean distance from Mid Hip to each wrist and each ankle). The last column of the matrix indicated which pose was in the image. I then ran this matrix through MATLAB's Classification Learner App, specifically Quadratic SVM, to see how those features performed. In all of the images here you can see the confusion matrix for Quadratic SVM, as well as the percentage of accuracy for Linear SVM, Fine Tree, Medium Tree, Coarse Tree, and



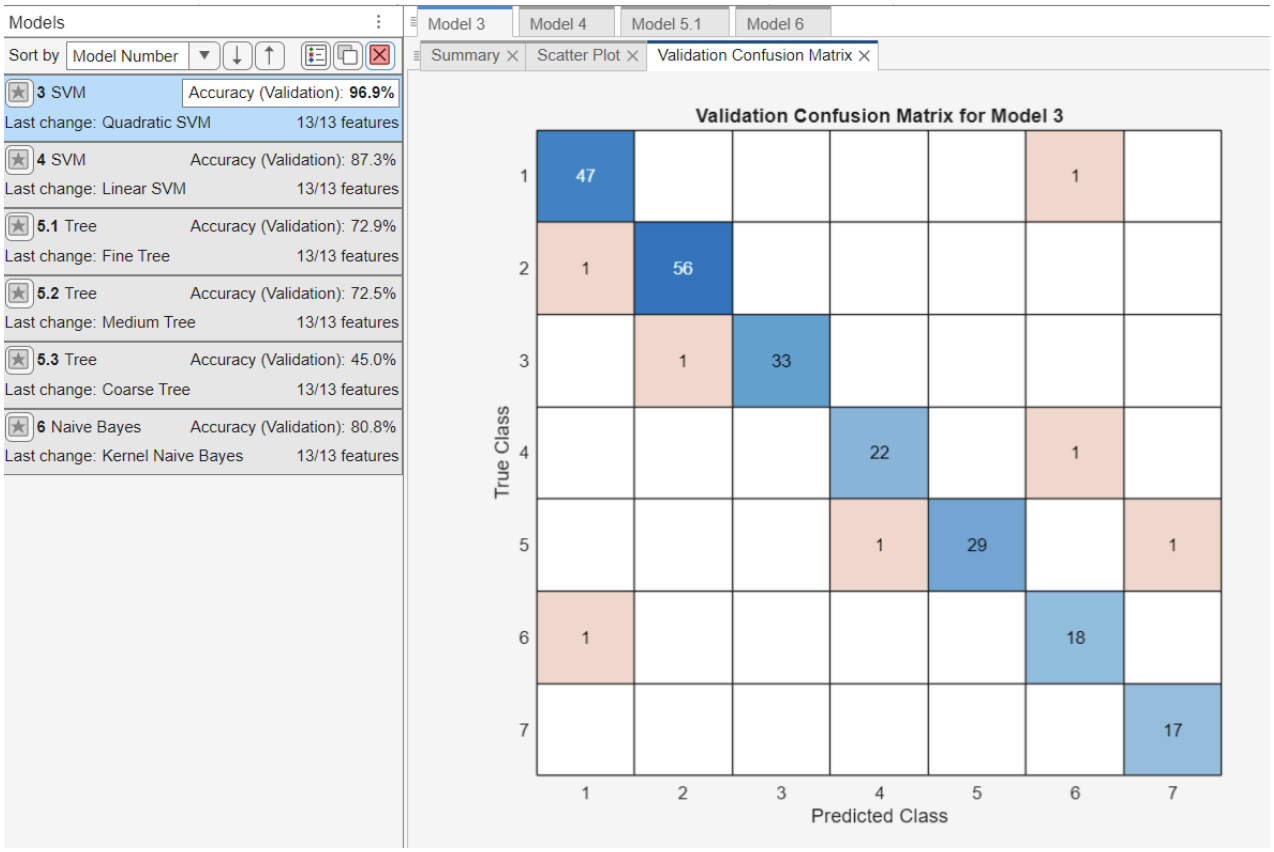
Naive Bayes classifiers. Also in this stage I cleaned up my data to only contain images with the dancers doing the pose (i.e. not in transition between poses) to eliminate noise. So I trained with around 230 images in the end. The image above shows the results of training Quadratic SVM with those features, the accuracies were consistently in the high 80s, usually around 86%. After seeing these results I also added the confidence of both wrists and both ankles as another feature to see if that data point made any difference. The below image shows the results from running that through Quadratic SVM, and I consistently got 90% accuracy, which was a slight improvement.



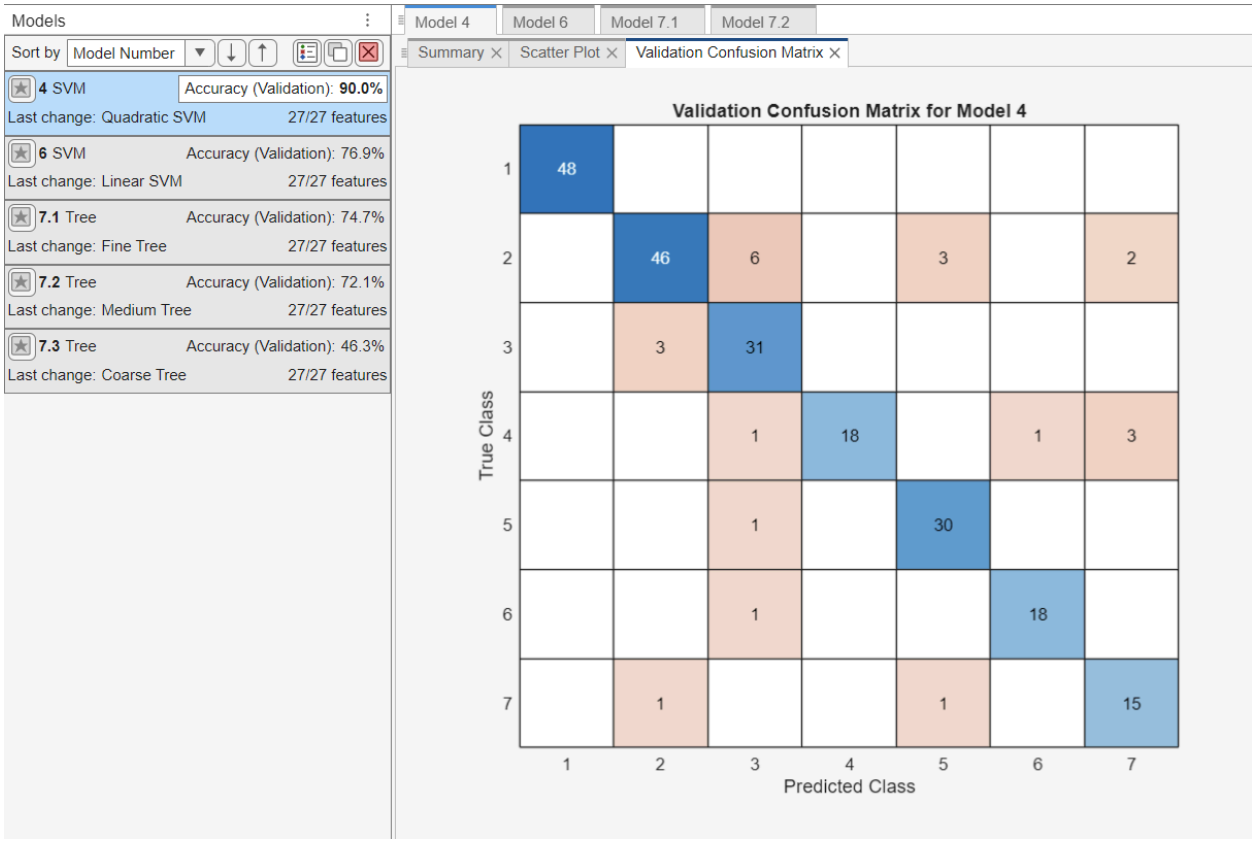
After looking at which poses the model got the most mixed up with Professor Izmirli, I decided to try and distinguish classes 2 and 3, as well as 4. So I decided to also add the Euclidean distance from Right Ankle to Right Wrist, as well as distance from Left Ankle to Left Wrist, which had the below results. Here I was consistently getting accuracies in the low to mid 90s for Quadratic



SVM, with this being the best confusion matrix I saw. The most mixed poses now were 1 and 6, so I added Euclidean distance from Right Ankle to Left Wrist and distance from Left Ankle to



Right Wrist, which gave the above results, the best ones I saw. Training with Quadratic SVM gave a consistent 96% accuracy, and I found the confusion matrix to be very impressive. To add one last point of discussion, I tested out throwing every possible feature at the Classification Learner and seeing how it did. I had the features be the distance from the Mid Hip joint to every other joint, giving the below results. The models did quite well, but not as well as with my very specifically



thought out features.

All of the code to run every part of the program is in the file **AnnikaHoag_FinalProject.m** (inside of **final_project_code** and **openpose**). To run the program where you train the data within the Classification Learner app, you should make sure lines 16 and 17 of the file are not commented out. Lines 32 through 40 should remain commented out. If you want to train using my handcrafted features uncomment line 23 and comment out line 24. Do the opposite if you want to train with all possible features. Leave line 25 uncommented for either option. Once you have done that, run the file as usual. Then switch the blue tab at the top of the screen to “App”, and choose “Classification Learner”. Choose “New Session”, and then “From

Workspace". When that window opens up, click on the dropdown menu under "Data Set Variable" and choose **features**. The dropdown menu under "Response" should automatically fill but if not choose the last column for this (column 14 if using handcrafted, column 28 for all joints). Then click "Start Session. You can choose whatever model you would like to train with here by clicking the upside down triangle next to the box with "All Quick-To-Train", "All", and "All Simulink Supported". Once you have chosen your model, click the green triangle button that says "Train All" to see the validation accuracy percentage and the confusion matrix. You can use this to evaluate the approaches I used. You can close the window when you are done and do this as many times as you would like.

The functions I wrote to plot the keypoints on the images and to plot potential features as line graphs are also in the file **AnnikaHoag_FinalProject.m**. To run the first function (plotting keypoints on the images) comment out all lines up to 31 and lines 38 through 40. Uncomment lines 32 through 34 and run the file as usual. For the second function (plotting potential features as line graphs) comment out all lines up to 37 and uncomment lines 38 through 40. Run the file as usual.