Assignment 2: Machine Learning

Part B

Exercise 3: Extended Feature Set

Results

Table 1: Average Weighted Accuracy for Rubine vs Extended feature sets when using 10-fold cross validation

		TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
Naive Bayes	**Rubine**	0.777	0.009	0.790	0.777	0.779	0.773	0.947	0.796
	Extended	0.800	0.008	0.823	0.800	0.805	0.801	0.954	0.801
Random Forest	**Rubine**	0.865	0.005	0.869	0.865	0.864	0.861	0.990	0.926
	Extended	0.887	0.005	0.892	0.887	0.887	0.884	0.991	0.934
Random Tree	**Rubine**	0.683	0.013	0.687	0.683	0.682	0.671	0.835	0.492
	Extended	0.610	0.016	0.619	0.610	0.610	0.597	0.797	0.407

Top Features:

1. Angle of the BB Diagonal
2. Bounding Box Height
3. Convex Hull Area Ratio

Discussion

The extended feature set saw improved results in almost every measure when using a Naive Bayes algorithm and a Random Forest algorithm. Furthermore, two of the top three features were not in the original Rubine feature set. As stated in *The Power of Automatic Feature Selection: Rubine on Steroids*, adding the right features improves performance.

The Random Forest only weakly relies on each classifier because the prediction is the result of many different decision trees. It resists overfitting to anomalies in training data because individual trees do not use every feature. The decision trees making up the random forest are generally shallower and are less fitted to the data samples, so adding more features would yield more accurate results. However, Random Forest can be computationally expensive, especially when more features are added. This can be resolved by choosing the optimal features, as suggested by Blagojevic, Chang, and Plimmer.

The Naive Bayes classifier is a probabilistic model that can also be thought of as a decision boundary. More features create a more accurate bounding shape. It is a simple model that, like all models, can fall victim to underfitting if the number of features used is limited too severely. The relative absolute error for this model was 21.1984%. This could be improved by adding more features.

On the other hand, when using the Decision Tree algorithm, the Extended Feature set saw worsened results, likely due to overfitting. Adding the additional features made the model too complex, and worsen performance. Decision Trees are prone to overfitting when using an excessive number of classifiers and a small training set, such as in this exercise. In the extreme case, each leaf of the decision tree can represent a distinct entry of the sample set. Therefore, when using a Decision Tree algorithm, it is important to employ automatic feature selection techniques.

Additionally, instead of increasing the number of features and making the model more complex, the results could be improved by increasing the size of the training data set. More data would reduce the Decision Tree model's bias toward anomalies found in the training set. This principle holds true across machine learning and statistics – larger sample sizes yield more accurate results.

Exercise 4: Gesture Recognizer

Results

The 3 classifiers used on the gesture features data were Naive Bayes, Random Forest, and Multilayer Perceptron. The Naive Bayes classifier had f-measures of 1.0, 0.988, 1.0, and 0.987 for gestures 1 through 4 respectively. Random Forest had the exact same f-measures 1.0, 0.988, 1.0, and 0.987 while Multilayer Perceptron had an f-measure of 1.0 for all gestures.

The top three selected features were features 1, 10, and 11, the angle of the bounding box diagonal, cosine of first to last, and cosine of the initial angle.

Discussion

All three classifiers performed very well on the dataset. For both the Naive Bayes and the Random Forest classifier, there was only one instance where a gesture was misclassified. The multilayer perceptron performed perfectly as can be seen by the f-measure of 1.0. When discussing the use of heuristics versus machine learning I believe this case is a good example of not needing machine learning in order to achieve good results. All three classifiers, although very different, performed almost perfectly. Additionally, the feature selection only identified 10 important features out of 45. A lot of features can be removed showing that only a few features of the gesture are needed to correctly identify the gesture. This allows heuristics to be used to create a simpler way of classifying the gestures.

Looking at the top three features provided by the feature selection can allow us to identify which gesture (1-4) corresponds to which command. Using only the cosine of first to last it is possible to narrow down most of the gestures. Gesture1 is most likely the swipe down gesture that indicates play/pause since the initial angle is always near 0. Gesture2 is most likely the swipe left to right gesture to go back since its initial angle is always near 1. Similarly for Gesture4, its initial angle is always near -1 which makes it most likely the right to left swipe to go to the next song.

Using these ideas for identifying the corresponding gestures a few heuristics can be made to classify

gestures. Starting with using the position of the start and end points, if the gesture has the same start and end points, or they are within some set threshold, then the gesture can be classified as the circular gesture to repeat a song since a circle should be a closed shape. Additionally, to add an extra safeguard for classifying the circular gesture, we could also add a heuristic corresponding to the total time taken to complete the gesture. Since all the other gestures only consist of straight lines and we know that drawing a circle should take longer than drawing a line, a value could be set as the threshold for the minimum time it should take to complete the circular gesture.

For the next and back gestures, if the x value of the starting point is before x value of the the endpoint (when going from the left to right) then it is the back gesture. Similarly, if the x value of the starting point is after the x value of the endpoint, then it is the next gesture. The y values in for both gestures can also be checked to make sure that they are about the same.

Finally, for the play/pause gesture, if the start and end point are roughly the same in the x axis while having different y values within some set boundary, then it can be classified as the pause/play gesture.

Although there may be a few more heuristics needed to increase the accuracy of the classification, these proposed heuristics would allow there to be a baseline to start from.

Exercise 5: Children's Shape Drawings in Sklearn

Results:

Our classifier achieved 98.94% accuracy on our training set, and 95% accuracy on our test set. The F1 score is 0.86 for shape 1, 1 for shapes 2,3, and 4, and 0.91 for shape 5.

	precision	recall	f1-score	support
shape1	0.75	1.00	0.86	3
shape2	1.00	1.00	1.00	3
shape3	1.00	1.00	1.00	4
shape4	1.00	1.00	1.00	5
shape5	1.00	0.83	0.91	6
accuracy			0.95	21
macro avg	0.95	0.97	0.95	21
weighted avg	0.96	0.95	0.95	21

```
Training Accuracy 98.94179894179894 %
------
Test Accuracy 95.23809523809523 %
```

Discussion:

We initially tried training a multi-layer perceptron classifier, but quickly realized that its performance was extremely poor. When trained on all 45 features, the MLPClassifier had a test accuracy of ~45%. We then performed feature selection in order to remove as many false correlations in the dataset as possible. Sklearn has many feature selection algorithms implemented; we first sorted features using the Mutual Information algorithm. We selected the top 15 features using this algorithm and then re-trained our neural net. This gave us a big jump in performance-our test accuracy was now 71%.

Unfortunately, no amount of additional hyperparameter tuning helped improve accuracy. We then changed to a Random Forest classification model, and used SkLearn's SelectKBest function for feature selection. These 2 changes improved our accuracy to 95%.

Our final model used the following features: Density 1, Convex hull area ratio, Openness, Length perimeter ratio, Distance first to last, Total length / BB diagonal, Thinness ratio, Perimeter efficiency, Width to height ratio, Angle of BB diagonal, Point ratio, and Log aspect.