**Research Question:**

Use machine learning to recognize images of hand-drawn figures as numbers, as an extremely simplified version of computer vision.

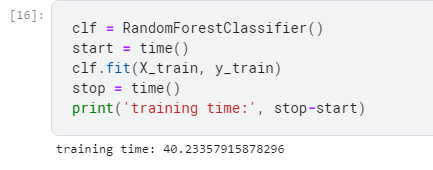
**Criterion:**

Categorization accuracy is used to score the models. In this case it is defined as the proportion of the predicted correct labels to the total number of labels (predicted and actual) for that class, then averaged across all classes.

**Model Development, Evaluation, and Results**

1. **Fit a random forest classifier using the full set of explanatory variables and the model training set (csv).**
2. **Record the time it takes to fit the model and then evaluate the model on the csvdata by submitting it to Kaggle.com. Provide your Kaggle.com score and user ID.**

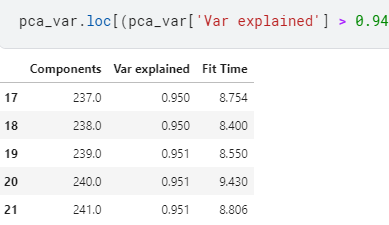
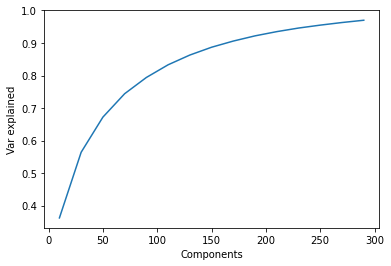
Time it took to fit a random forest classifier: 40.23s



1. **Execute principal components analysis (PCA) on the combined training and test set data together, generating principal components that represent 95 percent of the variability in the explanatory variables. The number of principal components in the solution should be substantially fewer than the explanatory variables.**
2. **Record the time it takes to identify the principal components.**
3. **Using the identified principal components from step (2), use the csv to build another random forest classifier.**
4. **Record the time it takes to fit the model and to evaluate the model on the csv data by submitting to Kaggle.com. Provide your Kaggle.com score and user ID.**

PCA: 240 PCs are enough to explain 95% of variance. 9.43s was needed for PCA fitting and transforming.

However, 147s was needed for the RandomForest fitting despite far less features.



1. **Use k-means clustering to group MNIST observations into 1 of 10 categories and then assign labels. (Follow the example here if needed: kmeans mnist.pdf).**
2. **Submit the RF Classifier, the PCA RF, and k-means estimations to Kaggle.com, and provide screen snapshots of your scores as well as your Kaggle.com user name.**

KMeans with 10 clusters: 96s fitting time

KMeans with 10 clusters and PCA: 38s fitting time

PCA seems to improve the score of a KMeans model, compared to Random Forests, where it lowered the score.

1. **The experiment we have proposed has a major design flaw. Identify the flaw. Fix it. Rerun the experiment in a way that is consistent with a training-and-test regimen, and submit this to Kaggle.com.**

None of the models were tuned or validated. We are essentially training on one set of data and using the hidden Kaggle public/private test set for testing, and the performance metrics are highly reliant on those two sets. The hyperparameters of the models are also not tuned for better results. In real world usage, the hidden Kaggle data set would essentially be deployment, and we have no idea how well our model is expected to perform.

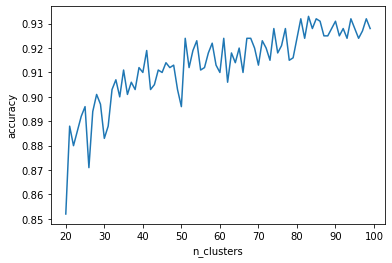
RFC:

Tuning hyperparameters on RFC did not give a test score improvement, despite a better CV score.

RFC with PCA: Slight improvement over untuned PCA scores.

KMeans:

Best CV scores at 81 clusters.



All models Kaggle results:

|  |  |  |
| --- | --- | --- |
|  | Public | Private |
| Untuned RFC | 0.9272 | 0.9238 |
| Tuned RFC | 0.925 | 0.9246 |
| Untuned PCA+RFC | 0.8964 | 0.8868 |
| Tuned PCA+RFC | 0.9008 | 0.8928 |
| Kmeans (10) | 0.6776 | 0.662 |
| PCA+Kmeans (10) | 0.6852 | 0.6694 |
| Kmeans (81) | 0.817 | 0.8164 |
| PCA+Kmeans (81) | 0.8268 | 0.821 |

**Insights:**

A simple random forest produced the best accuracy score, without any principal component analysis or any boosting factors. PCA did not reduce training time for random forests, despite only have a third of the total number of features. This is likely because now the features are all numerical, instead of binary, causing the model training time to increase. There is much more computational load since each feature can be split many different ways. None of the KMeans clustering results were as good as the simple RF, but they showed a big improvement with tuning for an optimum number of clusters, as well as with PCA involved. KMeans also did not show a big reduction in training time. In a tuned model with higher number of clusters, the time to train increased to more than 2 minutes.