Assignment 5 - MNIST Data Set

Executive Summary

As a data science manager, we need to assess how well PCA and machine learning performs in computer vision analytics. Computer vision is an important scientific field that seeks to automate more advanced tasks, such as, digital image or video processing through the use of computer analytics. This is done by developing methods for acquiring, processing, analyzing and understanding of high-dimensional data from the real world in order to produce numerical or symbolic information. Given the complexity and size of the data, it's important from a management perspective that the predictive accuracy of models be weighed against the costs of model development and implementation in order for the model to be effective.

Research and Design:

We utilized data from the MNIST database of handwritten digits as a platform for experimenting with learning techniques and pattern recognition methods on a real life example where dealing with a much larger dataset. This data includes 70,000 observations (images of numbers) and 784 pixels (features). We split the data into a 60,000 train and 10,000 test set based on the assignment requirements. First, we fit the feature set to a Random Forest classifier and recorded its runtime and F1 score, the harmonic mean of precision and recall. We then assessed the results of a scaled PCA preprocessor that had 95% variance explained by fitting the resulting principal components into a RandomForest classifier. The runtimes and F1 score were recorded then compared to the original model using the full feature set.

Technical Overview:

We used Python for our analysis to build and run the models for assessing F1 score. First before building the models it's important to standardize the data and apply scaling on the features, to bring all features to the same level of magnitude so all variables are equally weighted in the model. Scaling was needed for PCA. F1 score along with runtime will be used for model comparison purposes. PCA was then applied to slim down the features to those representing 95% of the variability in our features and compared against the full model. Running a model with reduced features should reduce the duration than running a model on the all features and without sacrificing a score with 95% of the variability in PCA. We then made sure to account for the define flaw in the assignment and reran the experiment applying PCA transformation only on the training set to limit overfitting of the data and have a valid test set.

Findings:

Step #1: We ran a random forest classifier using the full set of 784 explanatory variables and training set. It took 0.0804 minutes to fit the model and resulted in an F1 score of 0.95 (very high as 1 represents a perfect precision recall score).

Step #2: We executed principal components analysis on the full set of 70,000, generating principal components that represent 95% of the variability in the features. It took 0.253 minutes to generate 95% of the variability in the explanatory variables. By running this, we reduced the components to 332 or a 58% reduction in the number of features, but still accounting for 95% of variance in the PCA model.

Step #3: Using the 332 principal components identified in the PCA, we ran another random forest classifier on the training set. Surprisingly, it took slightly longer to fit the model with fewer components versus using all the features. The time it took to fit the model is 0.271 minutes and resulted in an F1 score of 0.895 where a reduction in F1 score is expected as not all variance in features is accounted.

Step #4: In comparing the performance of the full model with all features versus the 95% PCA model, we noticed the full model resulted in a higher F1 score of 0.95 versus the PCA model at 0.895. We also noticed the full model took less time to run at 0.0835 versus performing the PCA and calculating F1 score at 0.525 minutes.

There is a flaw is in Step #2, when the PCA was fitted to the entire dataset - we typically would want the test set to be held out until the very end to limit overfitting of the data and have a "unseen" test set. You want to fit PCA to only the training set and transform the test set with that. Rerunning the revised model slightly decreased the F1 score from 0.895 to 0.885 but sped up the rain time from .525 minutes to .5145 minutes.

Conclusion and Recommendation:

Overall, yes we would recommend using PCA as a preprocessing step to machine learning classification because it significantly reduces cost by cutting the features we need to measure substantially while maintaining a high F1 score. However, there needs to be a balance between cost and time as using a complete feature set will take longer than running a model with reduced features. Management needs to pay close attention to what is asked in the problem set and if PCA is relevant and should be applied to machine learning, or if we only care about time & precision / recall score then it should be left out of consideration as a preliminary machine learning model.