ML Findings from Published Movie Data (Wallisch & Whritner, 2017)
as reported by: Guilherme Albertini

Summary

I continue from where work was left off on Project 2; we are interested in applying machine learning methods to the movie rating and personality data. Once again, to cut down on false positives, the per-test significance level α is set to 0.005 (as per Benjamin et al., 2018). Some key findings include:

- PCA identified 8 factors that greatly reduced the dimensionality of the data (using Horn's method)
- Two optimal cluster groupings identified using the silhouette model
- The silhouette model gave a more convincing argument than elbow method for cluster grouping determination
- kMeans clustering alongside PCA produced a logistic regression model with AUC of 0.89
- Neural network (multi-layer perceptron regressor) with 100 hidden layer units yields coefficient of determination of 0.45
- The sklearn library is one of the most capable when dealing with machine learning methods

Discussion

First I used an overall PCA on columns 421-474 that, using Horn's algorithms, managed to identify 8 factors that were above noise and are considered throughout the rest of the analysis. These factors are shown in **Figure 1**. Note that the features consisted of responses to questions posed to the 1097 participants. Further information can be found in the notebook. Data was normalized (missing values used column averages) and fit to the PCA model. The transformed graph is made by plotting the first 3 main components in **Figure 2**.

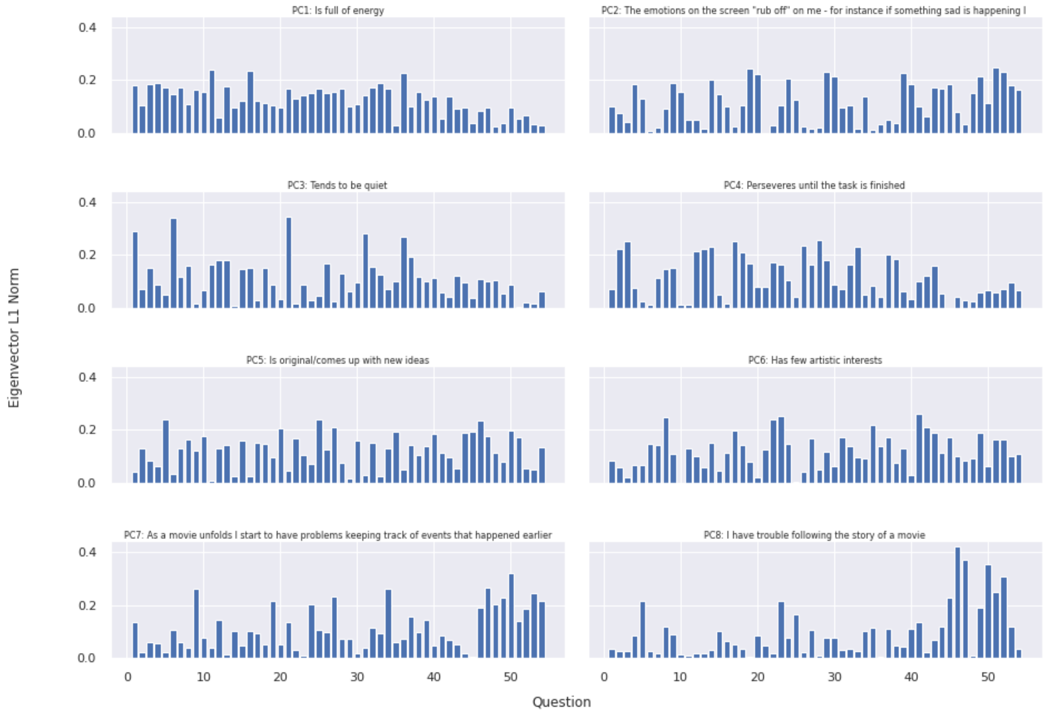

Figure 1: *Principal components identified in PCA. The L1 norm of the vector was considered when identifying these factors. \.*

PC1 explained the most variance in the Scree plot (consult notebook) and naturally showed the greatest magnitudes in all the directions of its eigenvector. The rest of the factors did not make the most convincing groupings when factoring in the kMeans algorithm; though the first 3 principal components explained roughly 26% of the data, they show the greatest cluster separation using kMeans along the principal components eigenvector axes (directions of greatest variance). Their centroids are colored in yellow in **Figure 2**. Both cluster centroids are roughly adjacent for the other PCA plots, suggesting that misclassification portion found in the score plot (the negative silhouette scores) come from components explaining relatively little of the data. This would prompt us to reconsider how many factors we should consider without making our model overly complex in future iterations. We can also decide to use another clustering algorithm in future work to compare against kMeans performance.

The silhouette method is chosen over the elbow method in determining the ideal number of clusters as the "joint" at the elbow plot was not easily discernible; the value of 2 and 3 for the number of clusters looked to be optimal. The silhouette score for each cluster was above all other average silhouette scores. While the fluctuation in size is similar, the thickness is more uniform with 2 clusters than that with 3. We also see a greater proportion of misclassification possible with 3 clusters and thus 2 groupings are chosen. See **Figure 3**.

Using the classifiers for the data from kMeans clustering, I go on to develop a logistic regression model to predict movie data (the predicted weights are found in the methods to the logistic regression function). kFold (10-fold) cross validation to avoid overfitting the model to the training data. An L2 penalty term is added to the logistic regression model when considering the previous results, whereby only the first 3 PCA components demonstrated clear clustered groups and other features were less predictive. The resulting AUC was 0.89, indicating a fairly good logistic regression model for this dataset.

A neural network is then developed using the sklearn multi-layer-perceptron regressor with 100 neurons in its hidden layer (the default). The data is normalized here again as neural networks of this sort are sensitive to feature scaling. Many of the default parameters were chosen for this model. The computed coefficient of determination for this model was 0.45, which indicates a fairly poor predictive capability. Further discussion of the tweaks made to the model are discussed below.

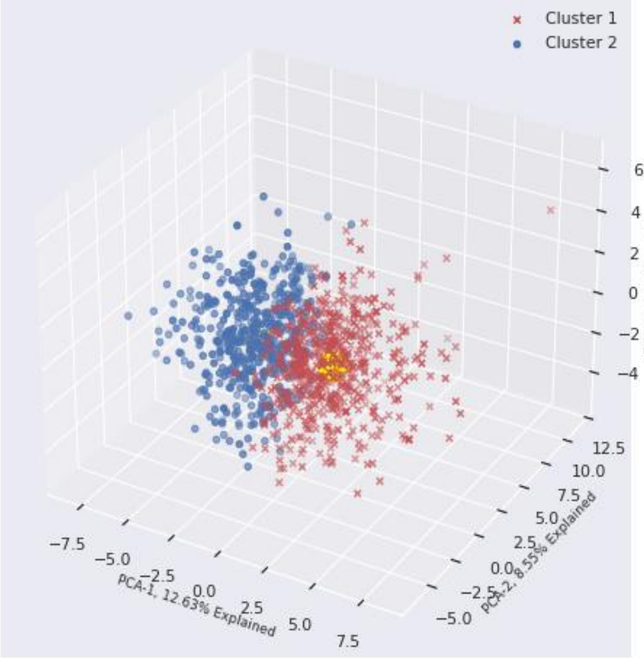

Figure 2: *First 3 principal components*

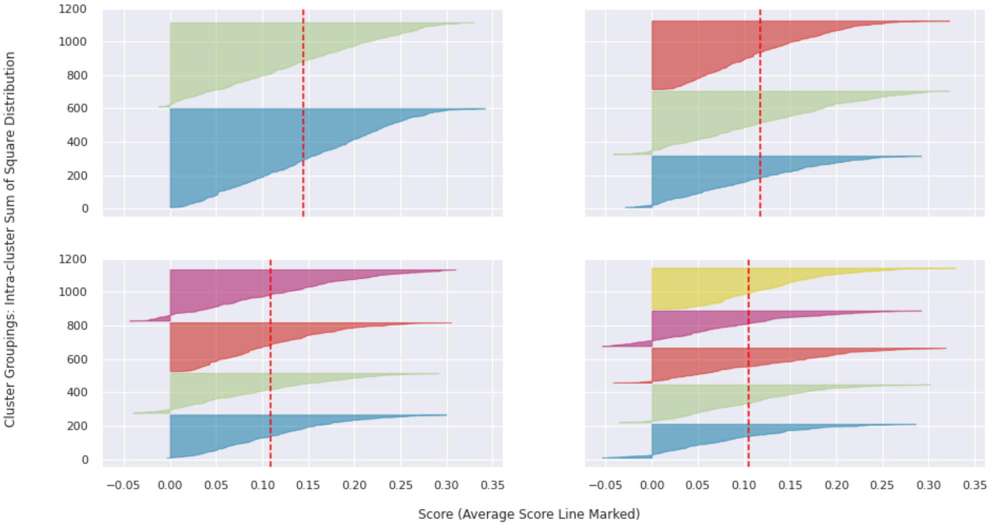

Figure 3: *Silhouette scores used to determine the optimal cluster groups of 2 considering misclassifications and average score line.. \.*

Further Discussion – Extra Credit with MLP

It was quite the challenge modifying the neural network architecture for the MLP regressor model. Many times, additional units and additional hidden layers would produce a coefficient of determination that was near 0 or negative, indicating that the fit was very poor when the sum of squares from the model (residual errors) are greater than the total residual errors from the mean; the model is no better than using a constant (mean) in these cases. Many architectures were tried in terms of modifying the number of hidden layers and neurons within them. The resulting chart shows the hidden layer (1 node is reserved for the input and output nodes and is not shown) and the corresponding coefficient of determination.

Hidden Layer Architecture (Hidden Layer 1, Layer 2,…, Hidden Layer n)	Coefficient of Determination (COD)
100 (Default)	0.45
100, 100	-2.08
40, 20	-0.51
80, 40, 5	0.04
80, 40, 10	0.48
80, 45, 10	0.46
80, 50, 10	-2.85
90, 15, 2	0.48
88,15,2	-1.46
10, 12, 15, 4, 3	-0.00
100,40,22,10,3	0.00
100, 80, 25	-3.50

The chart above shows just how sensitive adding nodes or hidden layers altogether is on the efficiency of neural network. A architecture of (80, 40, 5) has a horrid COD of 0.04 but by adding 5 nodes to hidden layer 3, it's COD skyrockets to 0.48. In contrast, Adding 5 nodes to the middle layer of this architecture decreases the COD to 0.46, and adding a further 5 neurons to the middle hidden layer yields an abysmal model!

These just show the many curiosities of neural networks. We scratch the surface here and would obviously not be using such a barbaric brute force approach when tuning the model; there were also many parameters that were kept as defaults and merit further investigation (using SGD solver vs ADAM vs LBFGS, determining max iterations, tolerances, etc.) and analysis of these parameters go beyond the scope of the material at hand.