1001 Data Analysis Project 3 Report - gjd9961

January 27, 2022

Introduction

The purpose of this report is to explain the results obtained from the statistical analysis of the "Movie Ratings Data-Set". The methods used in this analysis include but are not limited to: PCA, KMeans, K-Nearest-Neighbors, and Nueral Networks.

Question 1: Personality Factors and PCA

Before beginnings the first part of my analysis, I decided to fill in missing values in the dataframe provided by using the mean of each column. After the data-set was mean filled, I applied PCA, and inspected the eigenvectors and eigenvalues of the resulting covariance matrix. Looking at the eigenvalues, the first 10 principle components explained roughly 55% of the variance of the dataset. The remaining principle components had eigenvalues less than or equal to 1. For the sake of interpritability and simplicity, I did not use the Kaiser Rule, and rather disregarded these less meaningful principle components, and only used the top 10.

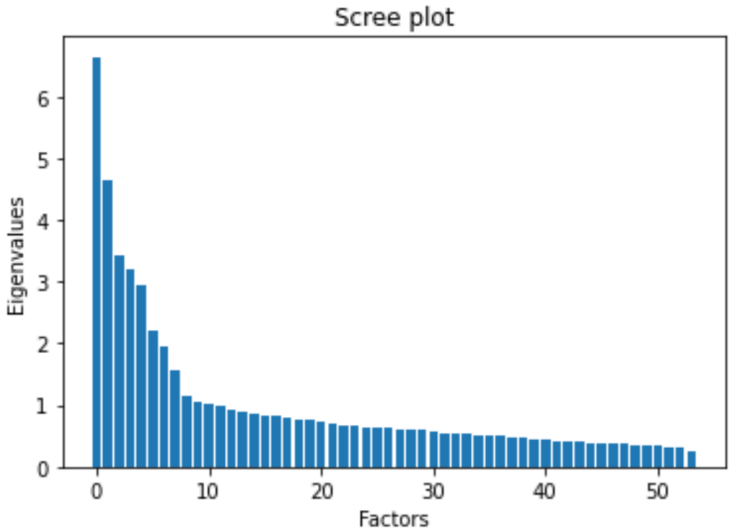

Figure 1: Scree Plot of Eigenvalues

In my python Notebook, I analyzed the loading matrix to identify which individual factors comprised each principle component by inspecting the loadings matrix. I then semantically interpreted the ten most meaningful principle components in the following way:

10 Most Meaningful Principle Components, Semantic Interpretation:

1. Enthusiastic vs negative mentality
2. Anxious/Impressioanble/Empathetic vs Not
3. Shy vs Assertive
4. Avoidant vs Confrontational
5. structured v unstructured personality
6. Type A vs Type B Personality
7. Casual Watcher vs Immersed in the Movie
8. Attentive and interested or disinterested and distracted
9. Artistic vs non Artistic
10. - Artistic vs Inventive

A more detailed breakdown of the individual components of the PCs is available in the code.

Question 2: Plot the PCs

As there were 10 principle components, and therefore 45 combinations of 2 PCs with 45 plots, I will omit the plots from this report. If you wish to see them, check out the code in the zip file.

Question 3: Identify Clusters in the New Space

Initially during my attempt to cluster individuals in the movie rating data-set, I tried finding clusters in pairwise comparisons between principle components. Essentially, I was looking at the plots derived in part 2, and trying to run KMeans clustering algorithm on these plots, and finding that in almost every pairwise comparison, the optimal clusters was 2 clusters. This was not very informative, as the minimum amount of clusters by definition of the KMeans algorithm is 2. So clearly something wasn't working.

I then tried clustering the users in the data-set by using their personality questions data in the PCA-altered form. This was much more informative, and as I ran KMeans in this higher dimensional space, I found that the optimal amount of clusters was approximately 12 (much more meaningful than 2). I used the silhouette method, to achieve the ideal number of clusters. The plots in Figure 2 illustrates how the KMeans sum of errors changed as the number of clusters varied.

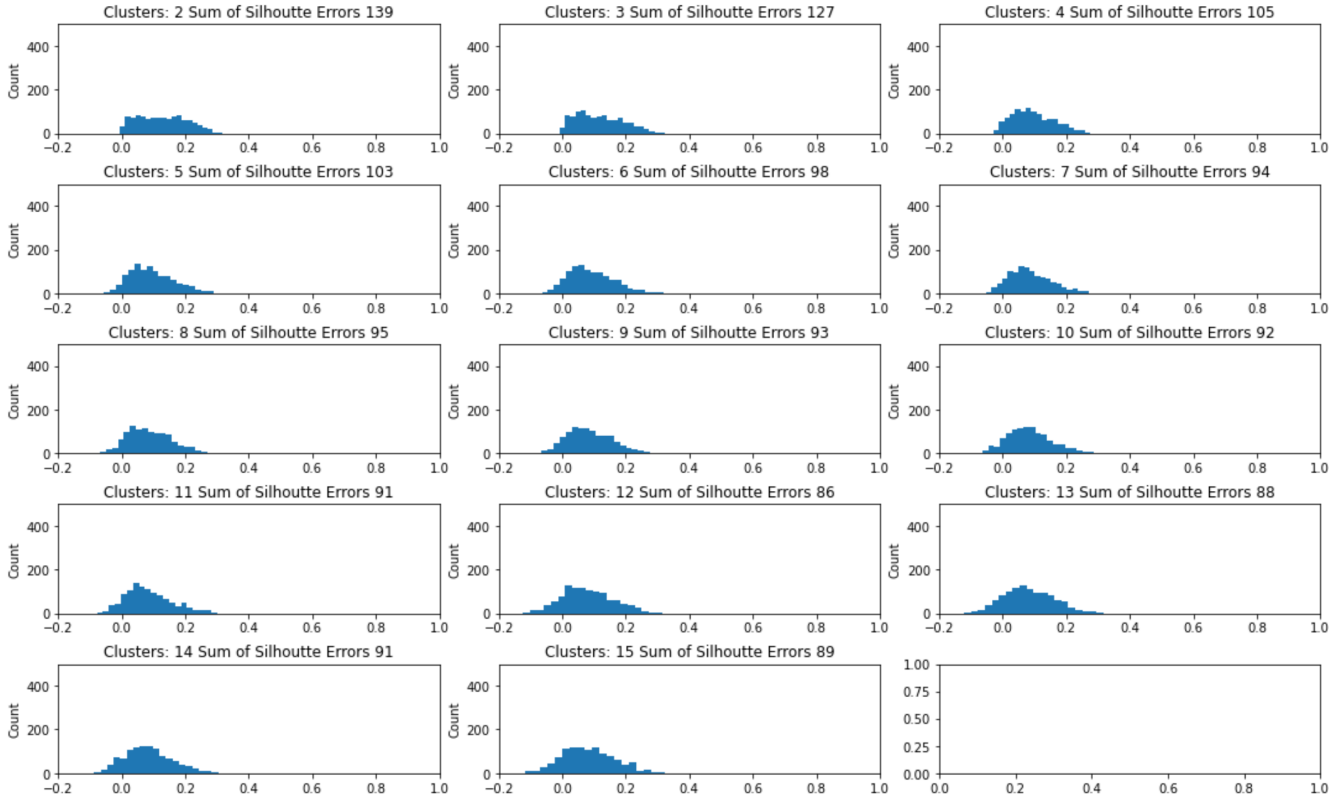

Figure 2: Plot of Kmeans Sum of Errors by Number of Clusters

Question 4: Use PCA Clusters to Perform Classification

Using the clusters I identified in Question 3, I implemented the KNearestNeighbors scitkit-learn algorithm to predict movie scores based on personality traits identified through PCA. I used KFolds cross-validation to split my dataset into 10 different train/test splits. For each of these train/test splits, I tried varying inputs of how many neighbors the KNearestNeighbors function should use when classifying. I did so to try to maximize the AUC score.

What I found is that beyond using 20-25 neighbors to classify which cluster a user belonged to based on movie ratings, there were diminishing returns associated with the AUC, with the average AUC score of each train/test split for each number of neighbors maxed out around 95-97%.

Question 5: Neural Network Prediction of Movie Ratings

I chose to use a simple Neural Network model from Sci-Kit Learn called MLP CLassifier. The model would use all but 1 column, to predict the excluded column in the dataset. In other words, the model would include all possible data points, (personality questions, movie scores, etc.) to predict any given column of movie ratings.

I used a simple train/test split, with a 70/30 train testing ratio. I printed the confusion matrix and classification report to evaluate the results for predicting the movie ratings of the

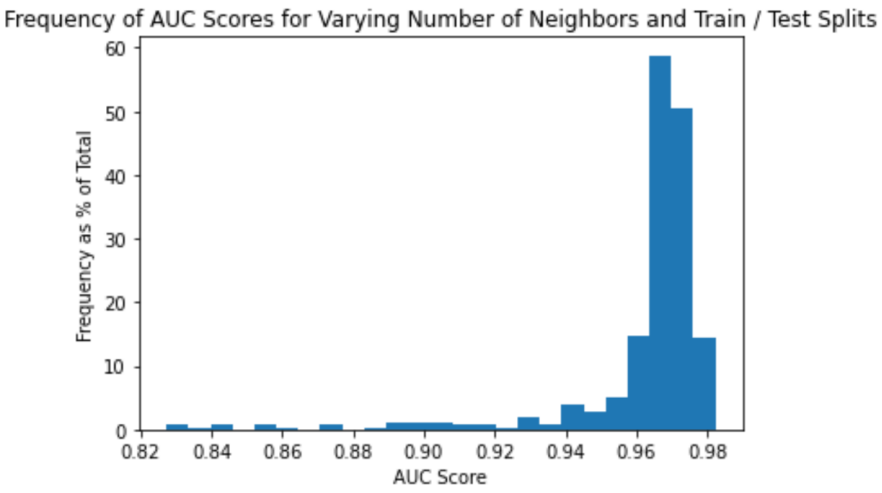

Figure 3: Average AUC Scores For Varying Of Neighbors and Train / Test Splits

movie "Anaconda (1997)". On 330 observations worth of training data, the model performed relatively well, with an weighted average precision of 80%, recall of 85%, and f-1 score of .82. However, this is a bit misleading as many of the values in the train and test splits were ratings of value "2" which was the rounded down mean-filled value. Thus, the model had exceptional recall and precision for movie ratings of "2", but very poor recall for ratings of other values. In other words, the model had learned that it was very efficient to simply guess the most common rating.

```
from sklearn.metrics import classification_report,confusion_matrix
print(confusion_matrix(y_test,predict_test))
print(classification_report(y_test,predict_test))
```

executed in 28ms, finished 18:37:03 2021-12-21

```
[[  2   0   4   1   0]
 [  1   0   8   3   0]
 [  1   0 276   8   0]
 [  0   0  18   4   0]
 [  0   0   2   2   0]]
              precision    recall  f1-score   support

           0       0.50      0.29      0.36         7
           1       0.00      0.00      0.00        12
           2       0.90      0.97      0.93       285
           3       0.22      0.18      0.20        22
           4       0.00      0.00      0.00         4

    accuracy                           0.85       330
   macro avg       0.32      0.29      0.30       330
weighted avg       0.80      0.85      0.82       330
```

Figure 4: Confusion Matrix and Classifiation Report of "Anaconda (1997")