IBM ML PROF. CERTIFICATION: FINAL PROJECT

JACKSON WALTERS

1. Introduction

This project is for the Exploratory Data Analysis for Machine Learning certification from IBM via Coursera. The long-term goal is to create a t-SNE plot to observe clustering of mental health disorders based on symptom data, as in the linked paper[1]. I would like to include additional data based on life factors such as income, housing status, and insurance status. The idea is that if someone has no income, is unhoused, and uninsured then they would be much more likely to be classified as having a mental illness. I want to go beyond correlation, and see clustering.

2. PCA and t-SNE for MNIST sample data

For now, I implemented code from here [2], and removed the SAS connection. I rewrote it in pure Python with sklearn. I did both PCA and t-SNE on the MNIST data, and plots are shown below.

All code is publicly available on GitHub [3].

Figure 1. PCA plot of MNIST handwritten digits data
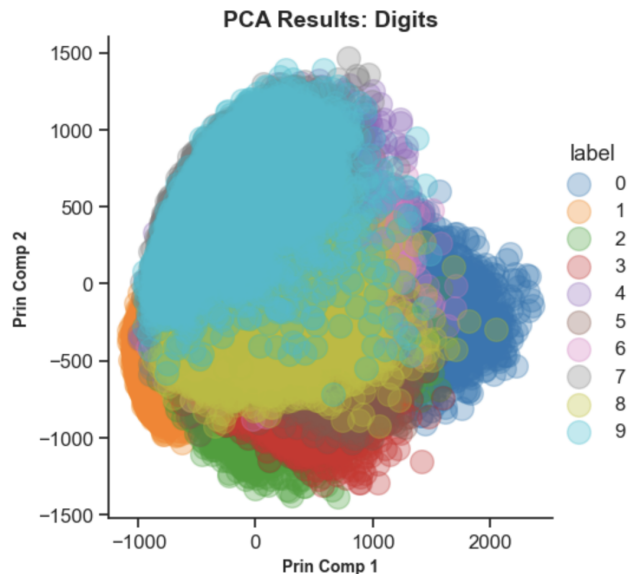
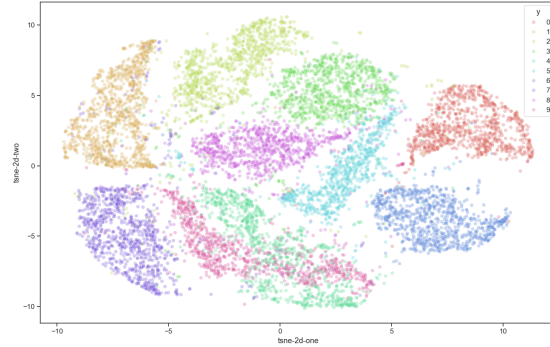## 3. Mental Health Client-Level Data

The data are 6.5mil rows and around 50 columns. There are 13 disorder codes listed, each a binary variable. The rest are life factors including employment, demographics like age, gender, ethnicity, geographic information, and more. I have dropped the irrelevant columns, one-hot encoded the categorical

---

[1]https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6783390/

[2]https://medium.com/@violante.andre/an-introduction-to-t-sne-with-python-example-47e6ae7dc58f

[3]https://github.com/jacksonwalters/ml-examples

FIGURE 2. t-SNE plot of MNIST handwritten digits data

columns, and scaled all columns to be in $[0, 1]$.

There are a few different ways to think of the labels, but they are all derived from the 13 binary diagnosis columns. This is an example of a multi-label problem, distinct from a multi-output or multi-class problem, though we may transform it into one of these. For example, in one labeling I bin the disorders as [#disorders = 0, #disorder = 1, #disorders > 1] giving 15 labels.

Another approach is to use a binary encoding so that each 13-tuple of binaries maps to a unique integer between 0-8191 since $2^{13} = 8192$. One could also count the number of disorders, yielding 13 labels, since the maximum sum is 13. Finally, one could use a more sophisticated approach like k-means, which finds a centroid of clusters and expands outwards. This is what we will typically use to color the t-SNE and PCA plots:

FIGURE 3. t-SNE plot of mental health data with k-means labeling
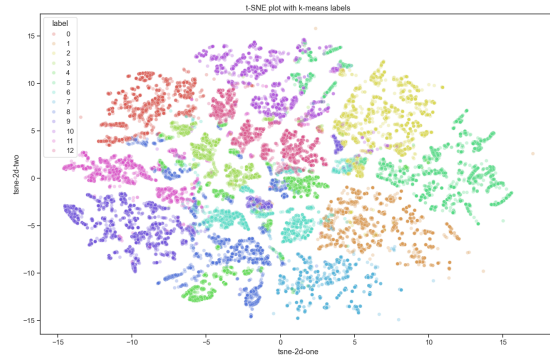
Here we choose $num_c lusters = 15$, and we can clearly see some clusters forming, likely from the demographic information but the diagnostic information is included. If we try to label with $unique_d isorders$, that is 13 categories for those diagnosed with a unique disorder and two for no disorders and multi-disorder, we obtain the following plot:

4. CLASSIFICATION

We can also attempt the classification problem. That is, given the demographic and life factor data, predict which disorder class the patient falls into. Since we have a number of labelings, we have a number of different problems to consider.

FIGURE 4. t-SNE plot of mental health data with $unique_disorder$ labeling
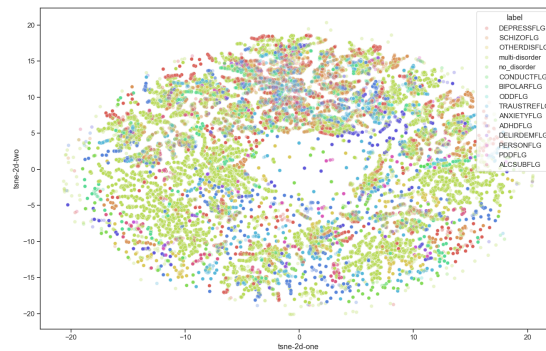
First we try the unique_disorders labeling and train a RandomForest classifier. This is a multi-class problem.

accuracy: 0.34965
precision: 0.30545804698445045
recall: 0.34965

Next up we try multi-class LogisticRegression. On the unique_disorders labeling, performance is similar to RandomForest. However, on the k-means labeling, we achieve much better performance:

accuracy: 0.88315
precision: 0.8819518553140321
recall: 0.88315

This reflects the much better clustering observed with the k-means labeling. Now the demographic and life factors are able to predict which k-means cluster a point will fall into, without diagnostic information.

We can try to drop 12/13 disorder columns and just do a logistic regression on the remaining column. This gives 13 separate logistic regressions, one for each disorder. We can then use it as a mutli-label classification, and compute the Hamming loss (a metric for distance between 0/1 datasets). In this case, we obtain hamming_loss=0.12.

Using MultiOutputClassifier or OneVsRestClassifier, we obtain hamming_loss=.1 in both cases. Using k-nearest-nearbors MLkNN from skmultilearn.adapt, we obtain hamming_loss=.11.

Finally, using MultiOutputClassifier with XGBoost, we obtain hamming_loss=.1. The accuracy score is modified a bit to "subset", and reads:

accuracy: 17.2%
precision: 46.8%
recall: 19.8%

It's worth noting that just computing the number of multi-labels that are correctly predicted with XGBoost is 3%, whereas a random guess would be $1/8192 = .01$ per data point, so 12.2%.

5. Unsupervised Learning

I decided to look at the disorder_cols of the data, disregarding all other columns. This is a Boolean matrix with a lot of structure to uncover. I used a number of clustering algorithms to try to understand which combinations of disorders were most common. I used Hierarchical Agglomerative Clustering, DBSCAN, and k-means. I used non-negative matrix factorization, PCA, and t-SNE for dimensionality reduction.

FIGURE 5. clustering results using DBSCAN

```python
from sklearn.cluster import AgglomerativeClustering
agg = AgglomerativeClustering(n_clusters=10, metric='euclidean', linkage='ward')
agg.fit(data[disorder_cols])
agg_label_df = pd.DataFrame(agg.labels_,columns=['label'])
```

```python
[84]: NUM_CLUSTERS_AGG = 10
```

```python
[19]: #use DBSCAN
      from sklearn.cluster import DBSCAN
      db = DBSCAN(eps=.1, min_samples=2, metric='euclidean')
      db = db.fit(data[disorder_cols])
      db_label_df = pd.DataFrame(db.labels_,columns=['label'])
```

```python
[20]: NUM_CLUSTERS_DB = len(set(db_label_df['label']))
```

```python
[67]: codebook_descriptions['OTHERDISFLG']
```

```
[67]: 'Other mental disorder reported'
```

```python
[37]: db_label_df.value_counts()[:10]
```

```
[37]: label
      0       9194
      1       7142
      23      3856
      16      3702
      18      3417
      9       3351
      20      2728
      2       2264
      34      1074
      19       872
      dtype: int64
```
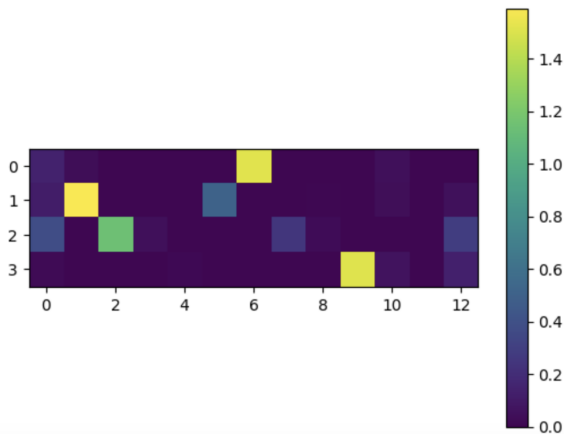
FIGURE 6. matrix factorization plot of H

```python
[23]: #perform non-negative matrix factorization
      from sklearn.decomposition import NMF
      nmf = NMF(n_components=4, init='random', random_state=0)
      W=nmf.fit_transform(data[disorder_cols])
      H=nmf.components_
      print(W.shape, H.shape)
      plt.imshow(H)
      plt.colorbar()
      plt.show()
```

```
(50000, 4) (4, 13)
```

6. Deep Learning and Reinforcement Learning

I implemented a three layer deep neural network to perform classification on the 14 diagnostic boolean labels. The output of the classifier is 14 numbers which are then rounded to 0 or 1.

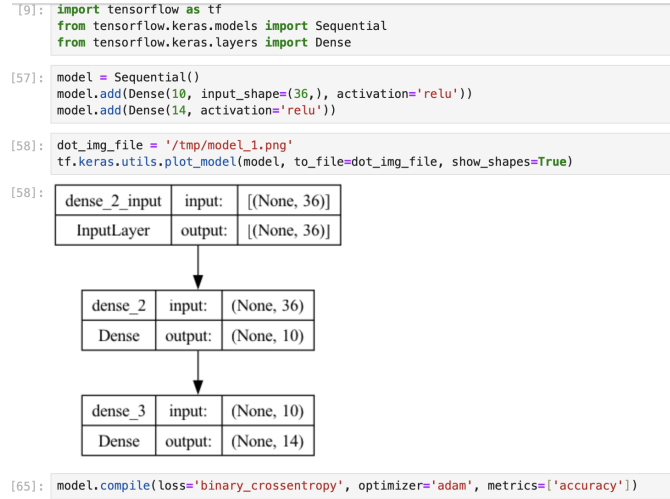FIGURE 7. Architecture of the deep learning model with three dense layers

FIGURE 8. Predictions of the model had 7% accuracy

7. Hierarchy of Multi-Disorders

Since each disorder is represented by a Boolean value, and there are fourteen different labels, we may view the vector of labels as a single multi-label. These are identified with subsets of 2^14. For example, suppose anxiety is the third disorder and ADHD is the fourth disorder, and someone is diagnosed with only these two disorders. Then the encoded vector would be $[0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$. Suppose depressive disorder is eighth. The vector would be $[0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]$. In this case the double disorder ADHD+anxiety is a subset of the triple disorder ADHD+anxiety+depression. We can map all of these with this relationship. We find that some multi-disorders occur much more often than others, and keep only the top fifty for visualization purposes. Generally there are no more than five disorders per person reported.

It is particularly illuminating to see that some doubles and triples occur with low or zero frequency. This eliminates pathways to developing a multi-disorder, assuming someone would develop the syndrome with 2-disorders before developing the one with 3 containing it, etc.

FIGURE 9. Hierarchy of multi-disorders, top 50


```
{'A': 'Substance use diagnosis',
 'B': 'Trauma- or stressor-related disorder reported',
 'C': 'Anxiety disorder reported',
 'D': 'Attention deficit/hyperactivity disorder reported',
 'E': 'Conduct disorder reported',
 'F': 'Delirium/dementia disorder reported',
 'G': 'Bipolar disorder reported',
 'H': 'Depressive disorder reported',
 'I': 'Oppositional defiant disorder reported',
 'J': 'Pervasive developmental disorder reported',
 'K': 'Schizophrenia or other psychotic disorder reported'
 'L': 'Personality disorder reported',
 'M': 'Alcohol or substance-related disorder reported',
 'N': 'Other mental disorder reported'}
```

8. CONCLUSION

We explored and analyzed the data using a variety of techniques. To detect clustering, we used the unsupervised k-means method and found that it was best, revealing clear clusters when viewed through the lens of a t-SNE plot.

DBSCAN revealed essentially only the relative percentages of each multi-disorder. We learned that this is in fact a multilabel problem, as opposed to a mutliclass problem. However, one can phrase it as a multiclass problem by looking at the $2^14 = 16,384$ distinct multi-disorder classed. Those are reduced to about 400 actual observed multi-disorders, most of which are single, double, or at most triple diagnoses.

The goal of this project was to do classfication, and we were able to use a variety of techniques including RandomForest, k-means, and logistic regression. It appears that RandomForest and LogisticRegression are both around 28% accuracy and 19% precision, which beats guessing the majority class which would be 16% accuracy. We don't expect great classification with no symptom-level data, which would be a future direction for this project.

(J. Walters) ARLINGTON, VA
Email address: jacksonwalters@gmail.com
URL: https://jacksonwalters.com