Predicting and Profiling Mental Health Risk

Samantha Roska, Thomas Hogan, Nial Alwash

Introduction

Mental health conditions are both prevalent and impactful for adults. Understanding the long term influence of childhood experiences, backgrounds, and behaviors is essential for developing intervention strategies and improving care. Our project aims to analyze and predict mental health diagnoses using the National Comorbidity Survey: Reinterview (NCS-2), 2001–2002 dataset, a nationally representative survey that captures adult respondents' self-reported past experiences and mental health histories.

The team is passionate about understanding mental health diagnoses at a deeper level, and by uncovering insights in the dataset we seek to provide clinical and policy-related findings. If we can accurately identify patterns in an individuals' behavior or in their past that are associated with mental health diagnoses, we can inform screening procedures, guide outreach strategies, and help improve select individuals' qualities of life.

The most interesting findings from our supervised results were that the logistic regression and random forest models performed very well in predicting negative cases, and the gradient boosting model performed well with positive cases. Given the nature of the dataset, we believed the gradient boosting model would be the best choice; however, it lacked sufficient accuracy and generated too many false positives. Our top feature was PH4 - Medication taken in the past 12 months. Overall, our results were inconclusive (too much variability), and we did not feel confident that we could appropriately predict Major Depressive Disorder with the dataset alone.

The analysis of the top three diagnoses across clusters reveals overlapping patterns of mental health conditions. Nicotine Dependence (DSM_TBD) and Panic Attack (DSM_PAT) consistently appear among the top diagnoses in Clusters 1, 2, 3, and 5, suggesting these conditions are prevalent across a wide range of individuals. Alcohol-related disorders represented by Alcohol Abuse (DSM_ALA) and Alcohol Abuse With Hierarchy (DSM_ALAH) are also prominently featured in the same clusters, indicating a potential co-occurrence of substance use and anxiety-related conditions in those groups.

Top 3 diagnosis by cluster				
Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5

DSM_TBD Nicotine Dependence	DSM_TBD Nicotine Dependence	DSM_PAT Panic Attack	DSM_TBD Nicotine Dependence	DSM_PAT Panic Attack
DSM_ALA Alcohol Abuse	DSM_ALA Alcohol Abuse	DSM_TBD Nicotine Dependence	DSM_PAT Panic Attack	DSM_ALA Alcohol Abuse
DSM_PAT Panic Attack	DSM_ALAH Alcohol Abuse With Hierarchy	DSM_SO Social Phobia	DSM_MJD Major Depressive Disorder	DSM_SP Specific Phobia

Table 1

Related Work

Several studies have looked at connections between early life experiences and mental health outcomes.

1. When Emotional Pain Becomes Physical: Adverse Childhood Experiences, Pain, and the Role of Mood and Anxiety Disorders. While the study also uses the NCS-2 dataset and focuses on adverse childhood experiences (ACEs), its primary goal is to examine how specific types of abuse relate to chronic pain, with mood and anxiety disorders acting as mediators. In contrast, our project investigates the broader impact of general childhood experiences on the likelihood of developing particular mental health diagnoses. Additionally, the approach in the study centers on mediation analysis, whereas our project applies machine learning to explore patterns and associations across a wider range of childhood factors and diagnoses.

2. Social criticism moderates the relationship between anxiety and depression 10 years later. The study also draws from the NCS dataset (both Baseline and Reinterview) and investigates the long-term relationship between anxiety and depression. It focuses on how perceived social criticism from friends and relatives influences the later comorbidity of these conditions over the ten year period. Unlike our project, which examines how past experiences related to later mental health diagnoses using machine learning, this study uses structural equation modeling to test moderation effects. It is more narrowly focused on interpersonal context as a variable, rather than looking at several potential predictors or diagnoses.

3. Negative Life Events and Incident Alcohol Use Disorders Among Ethnic Minorities. This report uses the NCS dataset to examine how recent negative life events are associated with the development of alcohol use disorders across racial and ethnic groups. This study found that while Black participants reported more negative life events

than White participants, incident AUD rates did not differ significantly across groups. Importantly, these negative life events were predictive of AUD amongst Whites and Hispanics but not for Blacks. We will not be looking at specific ethnic groups in our analysis and anonymize racial features in our analyses.

Data Source

Our project uses the National Comorbidity Survey: Reinterview (NCS-2), 2001-2002. This dataset is available through ICPSR and was downloaded in a CSV format. It contained survey responses from 5,000 adults aged 25-65 and included one thousand categorical and ordinal fields for each respondent. These features included behavioral, emotional, socioeconomic, and diagnostic domains. Most of the questions relating to past experiences were self-reported. The dataset also included diagnosis indicators based on DSM criteria.

We focused on a subset of features related to early life experiences and later mental health diagnoses. After filtering out administrative and checkpoint variables, we retained 891 columns. These columns also contained specific negative values representing different kinds of missing answers. For example, -6 represented situations where the respondent “didn’t know” the answer to the question being asked whereas -8 represented situations where the respondent “refused” to answer. These columns would end up being retained because they provided valuable information during clustering and other later analyses.

This dataset worked well for our project as it is extremely comprehensive. While other datasets may have diagnostic information, NCS-2 provides us with deep background on each subject, allowing for prediction modeling. The ability to analyze mental diagnoses from diagnostic, behavioral, and experiential lenses really separates this dataset from others. The main issue in our dataset is the limited number of participants, as it only focused on 5,000 individuals which made it difficult to generalize our findings to a population.

Data Filtering

We developed a pipeline to clean and prepare the dataset for both our supervised and unsupervised methods. Our raw data had 1000 columns in it, several of which contained a high number of null values. This was mainly due to two factors: one being that individuals who were self reporting did not exhibit the behaviors or traits that they were being asked about or individuals were unwilling to answer the questions. To

conduct accurate and meaningful analysis, we had to identify which columns we could remove to tighten our dataset.

We first highlighted the columns that were purely administrative. These columns contributed nothing to our analysis, so they were perfect to be removed. These columns were things like Respondent ID, Case ID, and Interviewer Checkpoint Questions. Next, we wanted to understand what kind of values were permitted in our dataset. All questions had their missing data encoded with different categorical variables being -9 (Missing), -8 (Don't Know), -7 (Not Applicable), -6 (Refused), and -5 (No More Mentions). To handle these, we converted most of the negative values to nulls, except for -8 and -6. We retained these two values because they still may reflect some level of familiarity with the question, but the respondents were uncertain or unwilling to answer.

Different questions had different answer options. These different options were encoded differently. For example, questions that had the traditional yes/no answering options had their answers encoded as 1 (Yes) or 5 (No), but questions regarding severity of behaviors had a scale of 1 thru 5, with 5 being the most extreme. As a result, these types of questions needed to be grouped separately.

We grouped the remaining columns by value patterns. These value patterns allowed us to see which types of questions (type determined by different answer options) were the most prevalent. We found that questions with the Yes/No or True/False encodings were the most common (encoded as 1/5 respectively). We found 253 distinct answering patterns across the 891 remaining columns. We decided to select answering patterns that had four or more columns in their type (for some analyses we extended this threshold to six instead of 4). After filtering these columns with this logic, we were left with 26 different answering types and 648 columns.

Preparing Data for Diagnoses

Next, we wanted to build a function that would prepare the remaining data for specific supervised and unsupervised tasks. We knew that different columns were useful for analyzing different diagnoses. As a result, we developed a framework for identifying which diagnoses we should target and which columns we should remove when targeting the selected diagnoses.

First, we examined the 43 different diagnoses in our dataset. These had two types of response formats: either binary (1/5) or ordinal (0/1/2). In the binary format, 1 indicates an Endorsed diagnosis and 5 indicates Not Endorsed. In the ordinal format, 0/1/2 represents Non-Case, Possible Case, and Probably Case, respectively.

A diagnosis is considered Endorsed when all DSM-IV criteria for that condition are met. On the other hand, Possible and Probably cases reflect partial fulfillment of the criteria. This partial fulfillment is enough to raise clinical suspicion but not enough to fully endorse the diagnosis. In other words, Endorsed diagnoses meet strict diagnostic standards, while Possible and Probably labels represent varying levels of clinical judgment.

After reviewing how diagnoses were grouped in the dataset, we assessed which diagnoses had the most non-null values. This helped us identify which ones had sufficiently rich data to support our analysis. From this, we selected a handful of target diagnoses to focus on. We then created a function that accepts three inputs: the filtered dataset, a selected diagnosis, and a minimum threshold. The function isolates the target diagnosis (removes the others) and keeps only the features where the proportion of non-negative values meets or exceeds the specified threshold.

For example, if a user selected Major Depressive Disorder with a minimum threshold of 30%, the function will retain only the columns where at least 30% of the values for that diagnosis are non-negative.

Our diagnosis focused preprocessing ensured that our feature selection was tailored, relevant, and consistent across both supervised and unsupervised learning tasks. It allowed us to preserve data quality while maintaining flexibility in targeting specific mental health conditions. A complete list of the final features used in our analyses is included in the appendix.

Part A: Supervised Learning

Methods Description

For the supervised learning section, we selected three different learning methods.

- **Logistic Regression:** We started with logistic regression as our baseline because it is a probabilistic method that is easier to interpret. First, we loaded different modules from scikit-learn, including train_test_split, GridSearchCV, Pipeline, and StandardScaler. Then, we loaded the filtered dataset created from the preprocessing and removed all diagnostic columns that start with the prefix “DSM-” besides DSM_MJD - Major Depressive Disorder, which was our target output variable. All the other columns would be used as our features. Next, we split our data into a test set and a training set, with the test set representing 20% of the total data. Then, we created our logistic regression pipeline using the

liblinear solver because the dataset is large and it involves binary classification, which supports L1 and L2 regularization. We did not need to use a scaler because this issue was addressed in the preprocessing dataset. For our hypertuning parameters, we used GridSearchCV on the penalty (L1 and L2) and regularization strength of c because it helps to optimize the predictive performance without overfitting. We found that our logistic regression model performs best based on the ROC-AUC score, with an L1 penalty and a regularization strength of 0.01, which represents a simpler model.

- **Random Forest:** Next, we tried the Random Forest model method because it is a tree-based model that is robust to feature interaction, which is important for our wide dataset. We again used the filtered data set created in preprocessing and used the same train-test split as we used in the logistic regression model. For our hypertuning parameters, we used GridSearchCV with a parameter grid that included “n_estimators”, “max_depth”, and “min_samples_split”. We tuned the “n_estimators” parameter - the number of trees- to see if an increase in complexity would perform better or if a simpler model would work, similar to logistic regression. Additionally, we optimized the “max_depth” parameter, which determines the depth of each tree, and the “min_samples_split” parameter, which specifies the minimum number of samples required for Node Splitting, to explore different levels of complexity and regularization. Our best results again for the ROC-AUC were 'max_depth': 10, 'min_samples_split': 10, 'n_estimators': 200. These results show that a moderately more complex model performs better, but there is some control for overfitting based on the higher number of trees - 'n_estimators'.
- **Gradient Boosting:** The final method we tried was Gradient Boosting, a boosted tree-based model to see if we could improve our accuracy through an iterative boosting approach. For this approach, we again used the filtered data and the same train-test split of 20/80. For our hypertuning parameters, we used two of the same from the random forest model, “n_estimators” and “max_depth”. We added a parameter, “learning_rate”, to determine if the model performed better with a more stable, slower rate versus a less stable, faster rate that converges quicker. Our optimal results were “n_estimators”= 100, “max_depth” = 3, and “learning_rate” = .1. These parameters reflect that our model was able to quickly converge, with a minimal number of shallow trees. These are promising results that show our model was not prone to overfitting.

Supervised Evaluation

The goal of this project is to predict, based on several features, if a person is diagnosed with Major Depressive Disorder. The metrics we chose to evaluate the model were Accuracy, Precision, Recall, F1-score, and ROC-AUC. We selected accuracy as our

baseline for overall correctness; however, it is limited due to the imbalanced nature of our dataset. Precision and Recall were selected because, given the sensitivity of the data, it is vital that we do not misdiagnose a mental health condition. F1-score was chosen to evaluate the trade-off between precision and recall. Finally, we selected ROC-AUC as our most important metric because of its ability to distinguish between positive and negative cases at various thresholds. It can effectively handle class imbalance and probabilistic classification models.

We ran all three models using cross-validation folds, as discussed above with the hypertune. Below is a chart of the results.

Results: ROC-AUC and Test Accuracy					
Model	**CV ROC-AUC Mean**	**Test ROC-AUC Mean**	**Test ROC-AUC Std**	**Test Accuracy Mean**	**Test Accuracy Std**
Logistic Regression	0.7863	0.7473	**0.0243**	0.7982	0.0148
Random Forest	0.7775	**0.7633**	0.0288	**0.8062**	**0.0127**
Gradient Boosting	**0.7940**	0.7594	0.0347	0.7882	0.0324

Table 2

Results: Precision, Recall and F1-score						
Model	**Precision (Neg)**	**Recall (Neg)**	**F1-Score (Neg)**	**Precision (Pos)**	**Recall (Pos)**	**F1-Score(Pos)**
Logistic Regression	**0.84**	0.95	**0.89**	0.58	0.28	0.38
Random Forest	0.82	**0.98**	**0.89**	0.60	0.13	0.21
Gradient	0.55	0.27	0.36	**0.84**	**0.95**	**0.89**

Results: Precision, Recall and F1-score						
Boosting						

Table 3

Based on these results, Gradient Boosting would be our best model, but it was still not as accurate as we would have liked to put into production. Gradient Boosting excelled in predicting positive cases with a recall-positive score of 0.95. In our case, we mustn't misdiagnose someone with Major Depressive Disorder. Even if our model yields false positives, we can conduct future testing to further evaluate the respondents. Both Logistic Regression and Random Forest did well in predicting negative cases, but in this scenario, this is less concerning. Finally, Gradient Boosting achieved the highest CV ROC-AUC score of 0.7941, indicating that the model performed well in predicting classes during cross-validation, although all the models were closely ranked in this metric.

Feature importance

Top 10 Features		
Feature Code	**Description**	**Importance**
PH4	Medication taken in the past 12 months	0.097139
PEC52	Often feel empty inside	0.068884
SA15	Severity of distress created by separation concerns	0.040146
M5	Have changes in behavior at time of irritability / grouchy	0.035889
NSD1E	Past 30 day often felt: blue	0.035313
M9	Severity of episodes interfere w/ work / social life / relationships	0.027730
PD9	Can remember exact age 1st time had an attack	0.026710
IR1INTRO2	Problems occur during irritable episode	0.022511

Top 10 Features		
PD1B	Problems during attack: Short of breath	0.021600
PD1G	Problems during attack: Dry mouth	0.020214

Table 4

The feature importance was calculated using the Gradient Boosting Model with the best model from the GridSearchCV module, and then we utilized feature_importances_ to identify the top 10 features. The most important feature was Medication taken in the past 12 months. This feature could be misleading if we were trying to predict whether someone would be diagnosed, as they may have been diagnosed in the past and are now receiving treatment. Still, it does show a correlation between taking medication and depression. Three features were related to panic attacks/disorder, and two features pertained to social interactions. The other features were related to a respondent's reporting of feeling empty or blue. There were 294 features with an importance of 0.00.

Sensitivity analysis

For our sensitivity analysis portion, we wanted to see how our Gradient Boosting model would perform if we eliminated our most important feature, "PH4" - Medication taken in the past 12 months, because there might have been some data leakage related to that feature. The models' F1-Score - positive went from 0.89 to 0.90, and for F1-Score - negative, it increased from 0.36 to 0.44. The test accuracy also increases from 0.7882 to 0.7972, accompanied by a lower standard deviation. However, the CV ROC-AUC Score decreased from 0.7941 to 0.781. Overall, removing the feature had a positive impact on many essential metrics but lowered the CV ROC-AUC score.

Tradeoffs

- **Positive vs. Negative Class:** The gradient boosting model achieved the highest recall and precision for the positive class but performed poorly for the negative class. Whereas Random Forest and Logistic Regression performed well with the negative class, they had many false negatives. In the class of our mental health analysis, it is more important to have accuracy in the positive class.
- **ROC-AUC vs. Model Complexity/Stability:** The gradient boosting technique had the highest ROC-AUC score but was the most complex model, had a high standard deviation, showing less stability, and required the most computational resources. Logistic regression has a slightly lower ROC-AUC score, but it is the least complex model, more stable than Gradient Boosting, and is the quickest

model to train with the least computational requirements. Finally, the random forest model falls between the other models in terms of ROC-AUC, complexity, and computational cost. Given the requirements, the tradeoff between accuracy and the complexity of Gradient Boosting models justifies the increased complexity, reduced stability, and additional computational cost compared to other models.

Failure analysis

There were two false negative observations (records 210 and 1869) and one false positive (180) that were reviewed in relation to our top features. Record 210 was notably different than the mode positive cases for *SA15 - Severity of distress* created by separation concerns (1 vs -7) and lower than the mode of *NSD1E - Past 30 day often felt: blue* (4 vs. 2). For record 1869 we show similar conflicting signals with a different between the mode for positive case in *M9- Severity of episodes interfere w/ work / social life/relationships* (3 vs -7) and *M5 - Have changes in behavior at time of irritability / grouchy* (1 vs -7) The false negative record 180 showed was close to the mode for the negative cases for the feature of *M9- Severity of episodes interfere w/ work / social life/relationships*.. Overall, it appeared M9 created confusion for the model, and further analysis could be done to see how the model performed with that value. Also, enhanced feature engineering with composite features might yield better results. In particular, clustering or interaction terms may help clarify unclear patterns and reduce misclassification by uncovering hidden relationships between features.

Unsupervised Learning

Methods description

- **Hierarchical Clustering**: Clustering method was a great first step to visualize both clusters and hierarchies within the data. Since it does not require a predefined number of clusters it is easy to implement.
- **K Means**: is computationally efficient and scales well to large datasets like the NCS-2 data.The results of the algorithm are straightforward to interpret; each data point is assigned to a single cluster. Additionally, the algorithm can be a good baseline model. The algorithm does however require defining a predetermined set of clusters.
- **DBSCAN**: is capable of detecting clusters with nonlinear and non-spherical formations. This flexibility is important because mental health data rarely conforms to simple linear or globular structures. It does not require predefining the number of clusters, which is ideal for the NCS-2 dataset where the true

number of mental health groupings or comorbidity patterns is unknown. Lastly, the algorithm naturally identifies noise and outliers in the data.

Evaluation

In order to get the correct number of clusters for our algorithms we tried two methods, a dendrogram for the hierarchical clustering and the elbow method for our other methods.

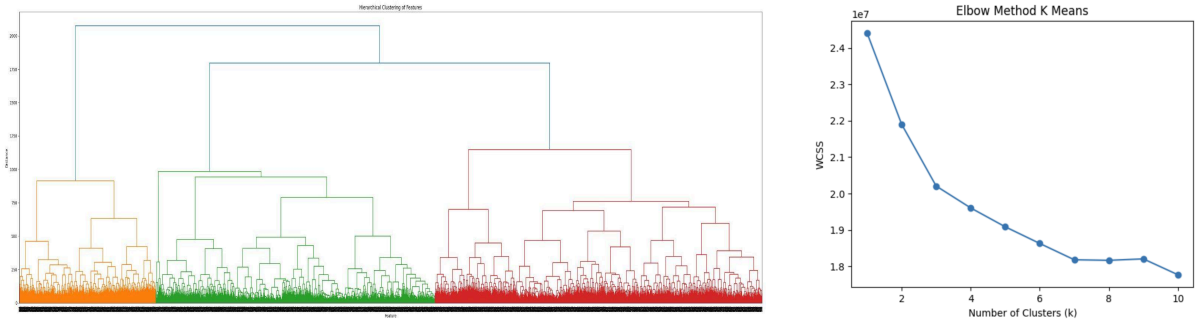

Figure 1

Reviewing the dendrogram (Figure 1) with ward linkage, it suggested that three clusters were prevalent in the data. Looking at the elbow curve in Figure 1, the K Means algorithm suggested that the correct number of clusters was also three, as seen by the largest drop in the graph. We used the Within Cluster Sum of Squares (WCSS) as the means of measurement for the K Means algorithm. Using WCSS gave a smoother graph than using silhouette score.

When first applying the KMeans algorithm based on the three clusters, to visualize the clusters we used PCA. Upon examining the graph it was clear that the data was dense and did not have clear linear relationships. We then chose to use the UMAP algorithm for better visualization and dimensionality reduction since the algorithm is more suited for large dimensional data like the NCS-2 dataset. Additionally, we tried to visualize the DBSCAN algorithm using PCA and got a similar result as the K Means graph. Switching to the UMAP algorithm also proved fruitful with the DBSCAN algorithm.

Now that we changed our visualization technique to use UMAP, we were seeing more clear clusters in our high dimensional dataset. Originally we used our dataframe without any dimensionality reduction and that resulted in the K Means model performing with a silhouette score of .107 and Davis-Boulden score of 2.61. We chose silhouette score because not only is it widely used but also works well for arbitrary cluster shapes. Silhouette score provides insight into the optimal number of clusters for K-means clustering and can handle noise and outliers for DBSCAN clustering. In addition to silhouette score, we chose to use the DBI metric because it also is widely used and can measure compactness and separation of clusters. It also works well with centroid-based

methods like K-means and helps evaluate DBSCAN algorithm eps and minimum number of points to form a cluster.

After reviewing the results of silhouette score and DBI on the data frame without dimensionality reduction, we decided to evaluate the models with the reduced dimensions. Table 5 shows the comparative results.

	K Means	DBSCAN
DBI score without UMAP	2.6	Undetermined. Not more than one cluster
DBI score with UMAP	0.748	0.445
Silhouette score without UMAP	0.109	Undetermined. Not more than one cluster
Silhouette score with UMAP	0.489	0.65

Table 5

The results in the table demonstrated that DBSCAN was the better model based on metric scores. Visually, the algorithms resulted in similar graphical representations, as seen in Figure 2, with the exception of the number of clusters identified via parameter tuning.

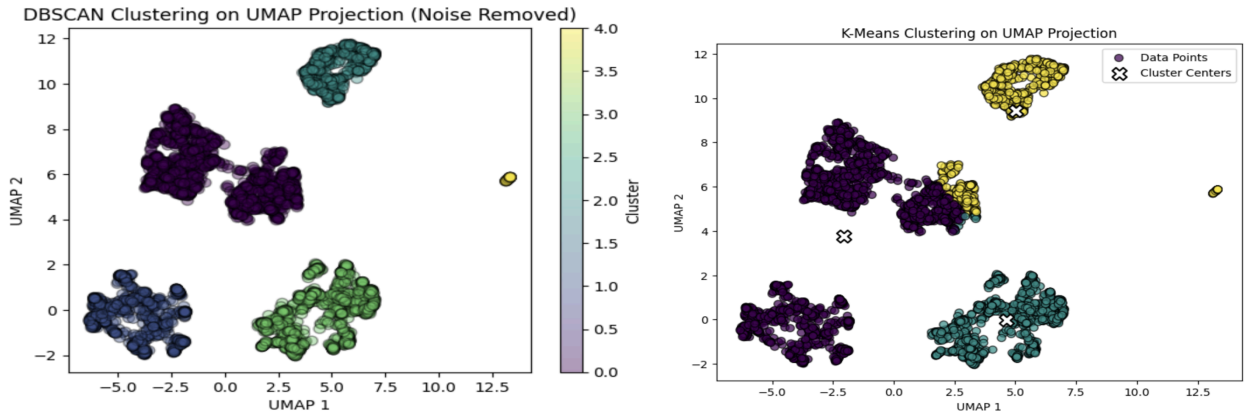

Figure 2

Performing a sensitivity analysis revealed that the DBSCAN algorithm locates the optimal eps between 0.7 and 1.4. Silhouette score stays consistently high (~ 0.65) in this range. Davies-Bouldin score remains low and stable (~ 0.4), indicating compact and

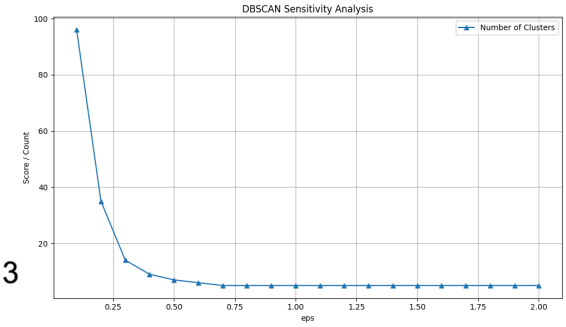

Figure 3

well-separated clusters. This suggests that DBSCAN is robust to small changes in eps within this interval. This range provides high silhouette scores and low Davies-Bouldin scores, suggesting good intra-cluster cohesion, clear inter-cluster separation, and stable results across small eps variations. The number of clusters, when eps is very small, DBSCAN detects 90+ clusters, likely indicating how the data points are close together. The sharp drop in clusters reduces the number of clusters from ~40 to ~15. The plateau region ~0.6-1.6 suggests strong clustering structure in the range and increasing eps has no effect.

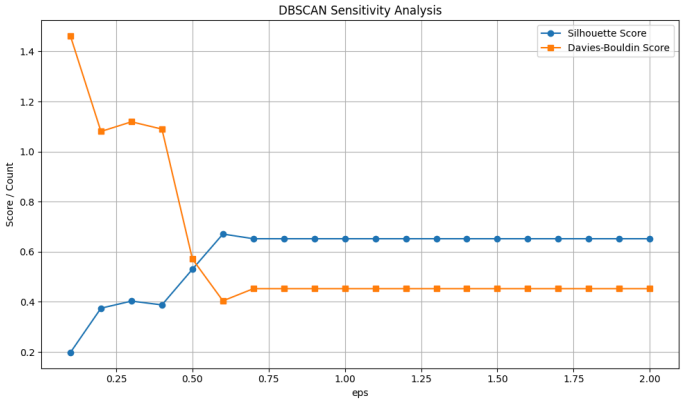

Figure 4

Discussion

- **Part A:** Our Random Forest and Logistic Regression Models performed better on predicting the negative class for the Major Depressive Disorder diagnosis; whereas our Gradient Boosted model was better at predicting the positive class. Predicting the both classes was important, but we prioritized limiting false negatives as much as possible because the consequences for doing so are less severe than predicting false positives (missing a diagnosis is worse than diagnosing someone who should not be). To evaluate these models, we used Precision and Recall metrics. If we had more time, we could use resampling techniques like SMOTE to better handle class imbalance, as well as more advanced feature engineering strategies like domain-related composite variables.
- **Part B**: A surprising result was that clusters 1–3 and 5 appear to be driven by externalizing or comorbid substance-related issues, while Cluster 4 may reflect a more internalizing symptom profile. Cluster 4 is all women respondents. Finding the right number of clusters was challenging because the output of our UMAP projections differed visually from those in our Dendrogram and Elbow Chart. If there was more time we would like to explore the patterns of comorbidity within and across clusters to help understand how co-occurring substance use and internalizing symptoms may interact. We could then understand if these clusters reflect meaningful clinical groupings.

Ethical Considerations

In Part A, we assume that endorsed diagnoses are the ground truth. In our dataset, a diagnosis is considered “endorsed” when survey responses align with DSM-IV criteria.

However, these are based on self-reported answers, not clinical evaluations. Thus, they may not reflect a formal diagnosis. As a result, our models may not reflect formal diagnoses. If our models learn from labels that have uncertainty, they will reinforce or amplify inaccuracies in the source data. To mitigate this we used logistic regression as a baseline and prioritized evaluation metrics like ROC-AUC which captured the trade-offs across classification thresholds.

In Part B, ethical concerns may arise surrounding the risk of inferring psychological groupings without clinical validation. Clustering individuals based on behavioral and diagnostic data may lead to overgeneralization. To address this, we treated clusters as exploratory to observe patterns rather than definitive categorizations. We relied on several internal validation metrics (Silhouette Score, Davies-Bouldin) and dimensionality reduction techniques, like UMAP, to support our interpretation.

Knowing that our dataset includes very personal and sensitive information, we ensured that all data handling aligned with ICPSR's ethical user standards and was conducted solely for analytical purposes, without any attempt to identify or profile individuals.

Statement of Work

Date	**Item**	
Week 1	**Team Formation, Dataset Selection & Decoding**	**All**
Week 2	**Initial Project Proposal, Final Proposal**	**All**
Week 3	**Data manipulation & Cleaning, Feature Selection**	**Nial**
Week 4	**Supervised Learning & Unsupervised Learning**	**Thomas, Sam**
Week 5	**Supervised Learning & Unsupervised Learning**	**Thomas, Sam**
Week 6	**Analysis + Evaluation**	**All**
Week 7	**Final Project Write up + Edits**	**All**
Week 8	**Final Project Review & Final Project Due 6/26**	**All**

Citations

Wijaya, Cornellius Yudha. "Exploring Unsupervised Learning Metrics." *KDnuggets*, 13 Apr. 2023.

Müller, Andreas C., and Sarah Guido. *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O'Reilly Media, 2016.

Pedregosa, Fabian, et al. "Scikit-learn: Machine Learning in Python." scikit-learn.org, 2024, scikit-learn.org/stable/. Accessed 24 May 2025.

Brownlee, Jason. "Machine Learning Mastery." machinelearningmastery.com, 2024, machinelearningmastery.com/. Accessed 16 May 2025.

"Imbalanced-learn Documentation." imbalanced-learn.org, 2024, imbalanced-learn.org/stable/. Accessed 01 June 2025.

Kumar, Rajesh. "A Guide to the DBSCAN Clustering Algorithm." *DataCamp*, 29 Sept. 2024, DataCamp .

Appendix

Features Used