Medical Students' Mental Health Scores and the Relationship to Their Health Satisfaction

Lily Hartmann

Department of Computer Science, Denver University

COMP 4448: Data Science Tools II

Daniel Parada

March 18th, 2024

Table of Contents

Introduction

Purpose

This paper aims to predict student satisfaction of health (on a scale of 1 to 5) based on several factors using their mental health, empathy, and burnout scores. This paper will be comparing the models: Decision Tree Classifier, Logistic Regression, K Neighbors Classifier, Random Forest Classifier, Ridge Classifier, and Support Vector Classifier.

Significance

If it is possible to predict students' satisfaction with their mental health, then it would be possible to preemptively provide care to students who may need it.

Research Question(s)

What model best predicts medical students' satisfaction with their health based on their empathy, mental health, and burnout scores?

Description of Dataset

This dataset contains the medical students' scores for several types of mental health testing. The main focus will be on psychotherapy within the last year, STAI (State and Trait Anxiety Inventory) scores, CES-D (Center for Epidemiological Studies-Depression Scale) scores, MBI (Maslach Burnout Inventory) Emotional, Cynicism, and Academic Efficacy scores and how they relate to medical students' rating of satisfaction with their health.

Psychotherapy visits are measured in a boolean format, 1 if the student consulted a psychotherapist or a psychiatrist within the past year, and 0 otherwise. Students rated their satisfaction with their health on a scale from one to five, where one was ‘very dissatisfied’ and five was ‘very satisfied.’

Data Preprocessing

Data Preparation

The first step of data preparation was to create a boolean version of students’ satisfaction with their health for the Logistic Regression model. Since the original scale was from one to five, scores one and two became a ‘0’ (for dissatisfied with their health) whereas scores four and five became ‘1’ (for satisfied with their health). Then, the score of three was replaced as ‘0’ and ‘1’ with equal proportions of each. It was then verified that there was an equal balance of positive and negative ratings students’ satisfaction with their health.

The next step of preparation was to standardize the numeric data. All score values, as well as age and hours studied were standardized with sklearn’s StandardScaler(). Next, dummy variables were created for the remaining variables, excluding health: year (year of medical school), sex, part (whether the student had a partner), job (whether the student had a part time job), and psyt (whether the student had received psychotherapy within the last year).

Finally, a correlation matrix with the health column was created. All variables with a correlation greater than .1 were included in the model. This included: cesd, stai_t, mbi_ex, mbi_ea, mbi_cy, psyt_0, and psyt_1.

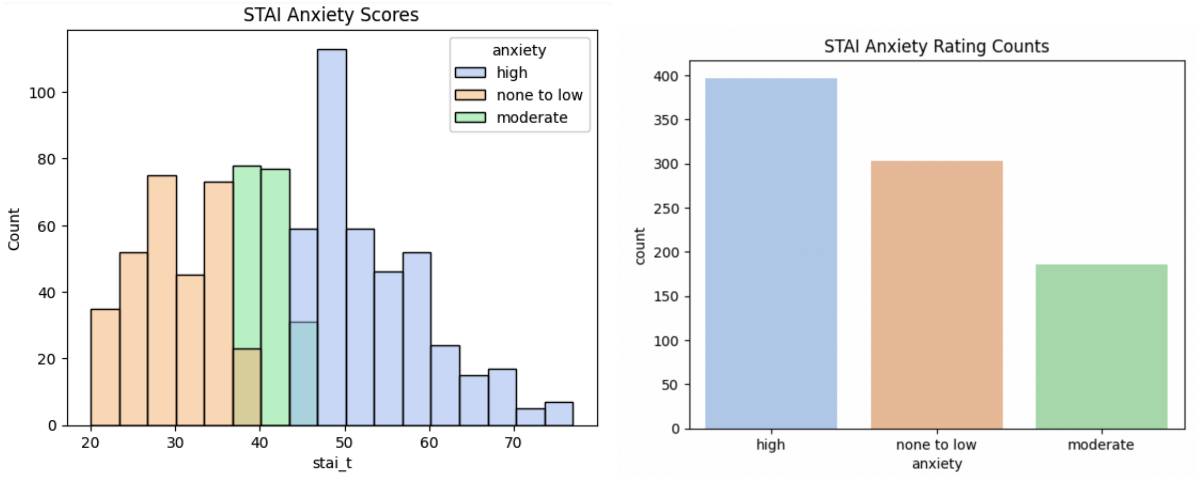
Exploratory Data Analysis and Visualization

State and Trait Anxiety Inventory

STAI is measured on a scale from 0 to 80. Scores from 20 to 37 are recorded as ‘no to low anxiety,’ scores from 38 to 44 are ‘moderate anxiety,’ and from 45 to 80 are ‘high anxiety.’

Figures 1 and 2

STAI Anxiety Scores Histogram and Anxiety Counts



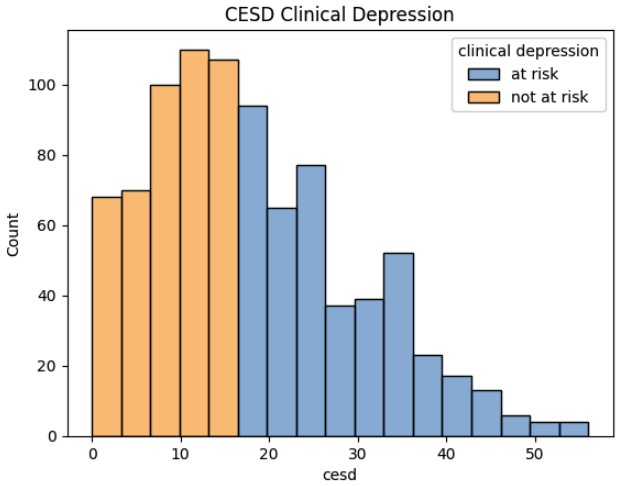
Note. Figure 1 is a histogram that depicts the STAI scores of medical students hued by the resulting anxiety rating. The majority of these medical students exhibit high levels of anxiety. Figure 2 is a countplot that depicts the overall counts per anxiety level as defined by STAI scores. The majority of these medical students exhibit a high level of anxiety.

Center for Epidemiological Studies-Depression Scale

The CES-D scale has a possible range from 0 to 60, where higher scores indicate the presence of more symptomatology of clinical depression. Scores below 16 are considered ‘not at risk’ for clinical depression, whereas scores greater than 16 are ‘at risk’ for clinical depression.

Figure 3

CESD Clinical Depression Histogram



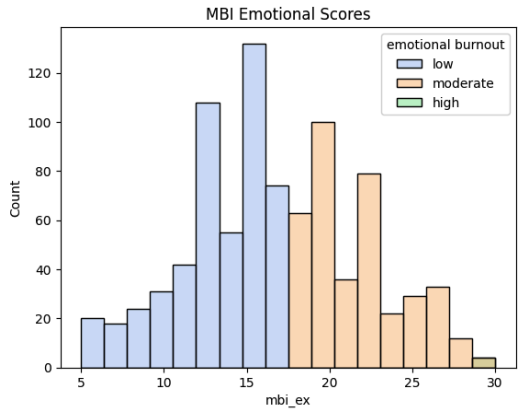
Note. The above histogram shows the distribution of students' scores CESD testing, hued by whether the student is at risk for clinical depression. The histogram is centralized around 15, however, it is skewed right, demonstrating that the majority of students are at risk for clinical depression.

Maslach Burnout Inventory

The MBI is measured differently for each category. The emotional score, is in three sections, low (0 to 17), moderate (18 to 29), and high (≥30). The cynicism or depersonalization score is rated as low (0 to 5), moderate (6 to 11), and high (≥12). Finally, the academic efficacy score is rated as low (≥40), moderate (34 to 39), and high (≤33).

Figure 4

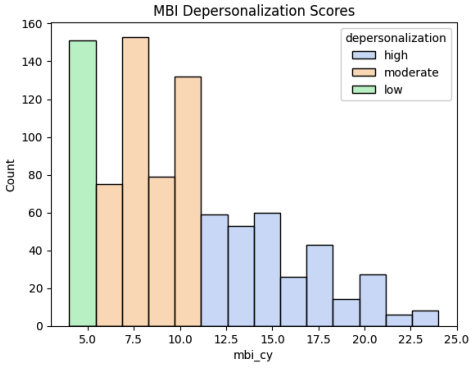
MBI Emotional Burnout Histogram



Note. The above figure shows the distribution of MBI scores for emotional burnout. Most students had either low or moderate burnout, while very few had high burnout scores.

Figure 5

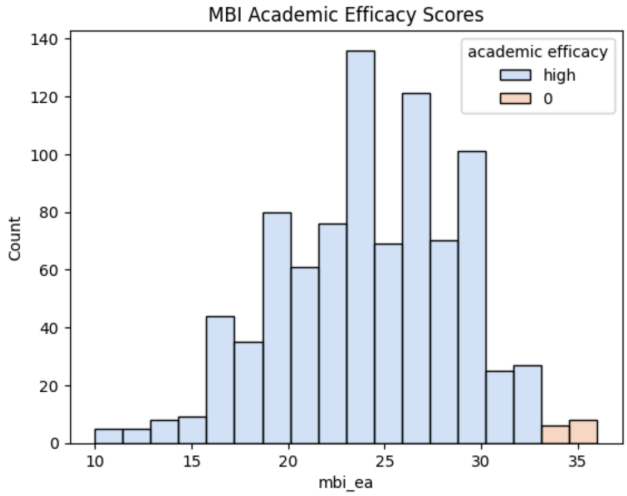
MBI Depersonalization and Cynicism Histogram



Note. The above figure is a histogram of depersonalization scores. The histogram is heavily skewed right, demonstrating that the majority of students have a high level of depersonalization, allowing them to distance themselves from their patients.

Figure 6

MBI Academic Efficacy Histogram



Note. The above histogram is above students' academic efficacy scores. Most students show a high level of academic efficacy.

Data Splitting

To split the data, train_test_split from sklearn's model selection was used with a test size of 30%. The splitting was done on a dataframe with the boolean health column and on a dataframe with the standard health column.

Figure 7

Data Splitting

```
from sklearn.model_selection import train_test_split

# using only top variables
feat_scores = data[["cesd", "stai_t",
                    "mbi_ex", "mbi_ea",
                    "mbi_cy", "psyt_1",
                    "psyt_0"]]



# SPLIT WITH ONLY SCORES
X_train, X_test, y_train, y_test = train_test_split(feat_scores, data["health"],
                                                    test_size = .3,
                                                    random_state = 13)

X_train_bool, X_test_bool, y_train_bool, y_test_bool = train_test_split(feat_scores, data["health_bool"],
                                                                        test_size = .3,
                                                                        random_state = 13)

print(X_train.shape, y_train.shape, X_test.shape, y_test.shape)
print(X_train_bool.shape, y_train_bool.shape, X_test_bool.shape, y_test_bool.shape)
```

```
(616, 7) (616,) (265, 7) (265,)
(616, 7) (616,) (265, 7) (265,)
```

Model Building and Evaluation

Model Building

Logistic Regression

Logistic regression is a classification model that describes the relationship between a categorical outcome variable and one or more input variables. In this case, a positive (1) is when the student is satisfied with their health and a negative (0) is when the student is dissatisfied.

Figure 8

Logistic Regression Model Building

```
lr = LogisticRegression()
lr.fit(X_train_bool, y_train_bool)

cv_lr = cross_validate(lr, X_train_bool, y_train_bool, cv = 3,
                       scoring = ("r2", "neg_mean_squared_error"))

acc_train = accuracy_score(y_train, lr.predict(X_train))

acc_test = accuracy_score(y_test, lr.predict(X_test))

to_add = ["Logistic Regression", np.mean(cv_lr["test_r2"]),
          np.mean(-cv_lr["test_neg_mean_squared_error"]),
          np.mean(np.sqrt(-cv_lr["test_neg_mean_squared_error"])),
          acc_train,
          acc_test]

scores_df.loc[len(scores_df)] = to_add
```

Note. First, the model is created and then scored with sklearns built in overall accuracy scorer, then R2, Mean Squared Error, and finally Root Mean Squared Error.

Decision Tree Classifier, K Neighbors Classifier, Random Forest Classifier, Ridge Classifier, and Support Vector Classifier

A Decision Tree Classifier is a supervised learning method that creates nodes for each input value and creates leaves labeled with the probability of that event occurring. A Random Forest Classifier is an ensemble method that uses a large selection of decision trees.

A K Neighbors Classifier model is used for classification using the idea that similar data points will have similar labels/classifications. Ridge Classifier models are a type of linear regression where regularization is added. Support Vector Classifiers are used for classification by returning a best fit hyperplane that divides the data.

Figure 9

Model Building Code

```
dtc = DecisionTreeClassifier()
rfc = RandomForestClassifier()
ridge = RidgeClassifier()
knc = KNeighborsClassifier()
svc = SVC()

models = [dtc, rfc, ridge, knc, svc]
names = ["Decision Tree", "Random Forest", "Ridge",
        "K Neighbors", "Support Vector"]

scores_df = pd.DataFrame(columns = ["Model", "R2", "MSE", "RMSE",
                                    "Accuracy (train)", "Accuracy (test)"])

for i in range(len(models)):

    model = models[i].fit(X_train, y_train)

    cv = cross_validate(model, X_train, y_train, cv = 3,
                        scoring = ("r2", "neg_mean_squared_error"))

    acc_train = accuracy_score(y_train, model.predict(X_train))

    acc_test = accuracy_score(y_test, model.predict(X_test))

    to_add = [names[i], np.mean(cv["test_r2"]),
              np.mean(-cv["test_neg_mean_squared_error"]),
              np.mean(np.sqrt(-cv["test_neg_mean_squared_error"])),
              acc_train,
              acc_test]

    scores_df.loc[len(scores_df)] = to_add
```

Note. All models other than logistic regression were created within one for loop using the same testing as the previous model.

Model Optimization

Before tuning the models, this table shows the accuracy values of each model for training and testing data.

Table 1

Model Accuracy Prior to Tuning

	Model	R2	MSE	RMSE	Accuracy (train)	Accuracy (test)
0	Decision Tree	-0.842834	2.089747	1.443681	1.000000	0.347170
1	Random Forest	-0.169646	1.327982	1.152294	1.000000	0.426415
2	Ridge	-0.029812	1.170345	1.081439	0.478896	0.479245
3	K Neighbors	-0.330878	1.511382	1.229272	0.534091	0.445283
4	Support Vector	-0.056039	1.199637	1.094884	0.504870	0.486792
5	Logistic Regression	-1.059346	0.511429	0.714851	0.035714	0.041509

As shown above, almost every model for this data was overfit, where the training data accuracy was higher than the testing data accuracy. In order to fix this, every model was tuned.

For each model, the same procedure was followed. First, a dictionary of parameters was created, and then a Grid Search Cross Validation was performed. The best parameters were output and then a new model was created using those parameters.

Figure 10

Logistic Regression Model Tuning

```
# grid search
params = {"tol": [0.1, 0.001, 0.0001, 0.00001, 0.000001],
          "fit_intercept": [True, False],
          "solver": ["lbfgs", "liblinear", "newton-cg", "newton-cholesky"]}

lr = LogisticRegression(max_iter = 100000, random_state = 13)

lr_cv = GridSearchCV(lr, params)
search = lr_cv.fit(X_train_bool, y_train_bool)
search.best_params_
```

```
{'fit_intercept': True, 'solver': 'newton-cholesky', 'tol': 0.1}
```

Note. Each model followed the same procedure as shown above. However, each model had its own separate group of parameters to search through.

Model Comparison and Selection

The best model, as shown in the table below, is the Logistic Regression model, with a testing accuracy of 0.49. However, since this model is overfit, as demonstrated by the higher training score, the best model, as shown in **Table 3**, without overfitting is the Decision Tree Classifier.

Table 2

Final Model Comparison

	Model	R2	MSE	RMSE	Accuracy (train)	Accuracy (test)
5	Logistic Regression	-1.065896	0.513055	0.715958	0.542208	0.490566
2	Ridge Classifier	-0.054425	1.197987	1.094432	0.478896	0.486792
4	Support Vector Classifier	-0.055265	1.199526	1.094787	0.483766	0.486792
0	Decision Tree Classifier	-0.168917	1.326458	1.149237	0.443182	0.483019
1	Random Forest Classifier	-0.070462	1.215889	1.102594	0.443182	0.483019
3	K Neighbors	-0.326655	1.507941	1.227264	1.000000	0.388679

Note. Table is ordered by Accuracy (test).

Table 3

Final Model Comparison (test accuracy greater than training accuracy)

	Model	R2	MSE	RMSE	Accuracy (train)	Accuracy (test)
0	Decision Tree Classifier	-0.168917	1.326458	1.149237	0.443182	0.483019
1	Random Forest Classifier	-0.070462	1.215889	1.102594	0.443182	0.483019
2	Ridge Classifier	-0.054425	1.197987	1.094432	0.478896	0.486792
4	Support Vector Classifier	-0.055265	1.199526	1.094787	0.483766	0.486792

Note. Same table as **Table 2** where models with higher training accuracy than testing accuracy have been filtered out.

Conclusion

Conclusion

As shown in the model comparison table, there were still several overfit models, but some were no longer overfit. However, none of the models had a high level of accuracy. Despite this, the best model was the Decision Tree Classifier.

I believe the lack of strong accuracy for all the models was due to the disagreement between the number of students satisfied with their health and their other mental health scores. The majority of students, 97.29%, were at risk for clinical depression and a majority of students had high levels of anxiety. However, 86% of medical students rated their health as 'neither satisfied or dissatisfied' or above.

Lessons Learned

Predicting people's satisfaction with their health solely based on mental health testing is not often accurate. The students may have rated their health only using their physical health, or valuing their physical health higher. They also may not feel their mental health scores are accurate or reflective. For example, although a student may be at risk for clinical depression, they may not feel like they are at risk, or feel any symptoms related to depression.

Recommendations

One recommendation would be to verify that the students are rating their health satisfaction with mental health, not physical health. Another recommendation would be to include data about students' physical health.

References

Brady, K. J. S., Ni, P., Sheldrick, R. C., Trockel, M. T., Shanafelt, T. D., Rowe, S. G., Schneider, J. I., & Kazis, L. E. (2020). Describing the emotional exhaustion, depersonalization, and low personal accomplishment symptoms associated with Maslach Burnout Inventory subscale scores in US physicians: an item response theory analysis. *Journal of Patient-Reported Outcomes*, *4*(1). https://doi.org/10.1186/s41687-020-00204-x

Burnout: Depersonalization, Exhaustion, Personal Achievement. (n.d.). Tesidea.com. Retrieved March 17, 2024, from https://tesidea.com/blog/decreasing-burnout-and-increasing-self-care-and-increasing-self-care/#:~:text=Burnout%20(or%20depressive%20anxiety%20syndrome

Carrard, V., Bourquin, C., Berney, S., Schlegel, K., Gaume, J., Bart, P.-A., Preisig, M., Schmid Mast, M., & Berney, A. (2022, July 1). *Dataset for the paper "The relationship between medical students' empathy, mental health, and burnout: A cross-sectional study" published in Medical Teacher (2022)*. Zenodo. https://doi.org/10.5281/zenodo.5702895

Center for Epidemiologic Studies Depression Scale Revised (CESD-R-20). (n.d.). https://www.brandeis.edu/roybal/docs/CESD-R_Website_PDF.pdf

GERT User's Guide 1. Goal, features, and versions of the GERT. (n.d.).

Goodhew, S. C., & Edwards, M. (2022). The relationship between cognitive failures and empathy. *Personality and Individual Differences*, *186*(186), 111384. https://doi.org/10.1016/j.paid.2021.111384

Health, M. (n.d.). *NIDA STTR for Vulnerable Populations Seek, Test, Treat and Retain for Vulnerable Populations: Data Harmonization Measure*. https://nida.nih.gov/sites/default/files/Mental_HealthV.pdf

Kayikcioglu, O., Bilgin, S., Seymenoglu, G., & Deveci, A. (2017). State and Trait Anxiety Scores of Patients Receiving Intravitreal Injections. *Biomedicine Hub*, *2*(2), 1–5. https://doi.org/10.1159/000478993

Kerr-Gaffney, J., Harrison, A., & Tchanturia, K. (2019). Cognitive and Affective Empathy in Eating Disorders: A Systematic Review and Meta-Analysis. *Frontiers in Psychiatry*, *10*(102). https://doi.org/10.3389/fpsyt.2019.00102

Williams, B., & Beovich, B. (2019). Psychometric properties of the Jefferson Scale of Empathy: a COSMIN systematic review protocol. *Systematic Reviews*, *8*(1). https://doi.org/10.1186/s13643-019-1240-0