Essay / Assignment Title: Improving the medical care: A Data Analytic Approach

Programme title: MSc Data Analytics

Name: Birce SARI

Year: 2023

CONTENTS

Statement of compliance with academic ethics and the avoidance of plagiarism

I honestly declare that this dissertation is entirely my own work and none of its part has been copied from printed or electronic sources, translated from foreign sources and reproduced from essays of other researchers or students. Wherever I have been based on ideas or other people texts I clearly declare it through the good use of references following academic ethics.

(In the case that is proved that part of the essay does not constitute an original work, but a copy of an already published essay or from another source, the student will be expelled permanently from the postgraduate program).

Name and Surname (Capital letters):

..........BIRCE SARI.............

Date: ..30.../.01.../..2024.

INTRODUCTION

The healthcare industry is constantly evolving to meet the needs of patients, with new techniques for cures and treatments emerging every day. As a hub for diverse patient needs, we have implemented a system that allows patients to share valuable insights into their health concerns, which forms the basis for a personalized care model. To optimize our services, we use data analytics to develop an algorithm that can effectively categorize patients. However, integrating data analytics into healthcare also poses challenges, particularly as we strive to provide comprehensive and personalized care at our medical center.

The healthcare industry is constantly evolving to cater to the needs of human care. The techniques of cures and treatments are advancing rapidly. The Internet of Things (IoT) enables personalized data collection, making it easier for us to receive better support for our medical needs. Additionally, we are exploring the possibility of using both real patient data and synthetic data generated through advanced technology to enhance the precision and efficiency of our categorization algorithm. This approach aligns with the broader discussions surrounding personalized healthcare, patient engagement, and the utilization of data analytics to navigate the intricacies of modern medical practices. Therefore, I will try to explain and demonstrate the usage of a synthetic healthcare dataset obtained from code. I chose a synthetic dataset that was denoted as computer-generated datasets engineered to emulate real-world data. Possessing identical mathematical properties to authentic data, synthetic datasets, nonetheless, abstain from replicating specific information. As detailed in studies such as (Arents, J. & Greitans, M. 2022), synthetic datasets recreate situations mirroring real-world complexities. While in controlled environments and less intricate scenarios, the benefits of synthetic data may be constrained by the reality-gap issues, the potential becomes evident when faced with the need for vast datasets encompassing diverse environmental parameters and complex scenes. The paper suggests that a combination of real and synthetic data could strike a balance between the efficiency of the data collection process and the precision of the trained model.

CHAPTER ONE: COLLECTING AND DATA PRE-PROCESSING WITH VISUALIZATION

Aggregated Virtual Patient Model Dataset (VPM), dataset aggregates clinical parameters from older individuals, including scores, device-derived data (e.g., daily heart rate), survey based details, and related events (falls, loss of orientation). It is declared that data collected during clinical evaluations by experts, it reflects the individuals' status across physical, psychological, and cognitive domains. Clinicians use the dataset's medical features to assess overall well-being. The Virtual Patient Model aims to evaluate older individuals' overall condition based on these parameters and identify connections with frailty status (Deltouzos, 2020).

In addition, Patil's 2024 research on healthcare data sets, titled "Healthcare Dataset: Dummy data with Multi-Category Classification Problem," showed us the role of the dataset goes beyond mere educational utility and extends to multi-class classification problems. This enhances its educational value, providing a nuanced understanding of health analytics concepts and fostering innovation in the broader data science landscape.

Demographic Information:

- Gender: Gender of the individuals.
- Age: Age of the individuals.

Health History and Lifestyle Factors:

- Hospitalization One Year: Number of nonscheduled hospitalizations in the last year
- Hospitalization Three Years: Number of nonscheduled hospitalizations in the last three years.
- Orthostatic hypotension: Presence of orthostatic hypotension
- Weight Loss: Unintentional weight loss >4.5 kg in the past year (categorical answer)
- Exhaustion Score: Self-reported exhaustion (categorical answer)
- BMI Score: Body Mass Index (in Kg/m²)
- Health Rate: Self-rated health status (qualitative ordinal evaluation)
- Health Rate Comparison: Self-assessed change since last year (qualitative ordinal evaluation)
- Pain Perception: Self-rated pain (visual analogue scale 0-10)
- Smoking: Smoking (categorical answer)
- Comorbidities Count: Number of comorbidities
- Medication Count: Number of active substances taken on a regular basis

Depression-related Information:

- Depression Total Score: Total depression score, 15-item Geriatric Depression Scale (GDS-15)

Figure 1-1:Dataset details (Deltouzos, 2020)

Data discovery part is constructed with data installment, dropping the missing values, and checking correlations between numerical columns to find relevant information about depression score to obtain correct assumptions. Relevant data changed into more understandable syntax, classified the ‘Depression Total Score’ for better understating, if they need minor or major help or not. For a better understanding, as illustrated in figure 2, general distribution is around 77 years old male patients.

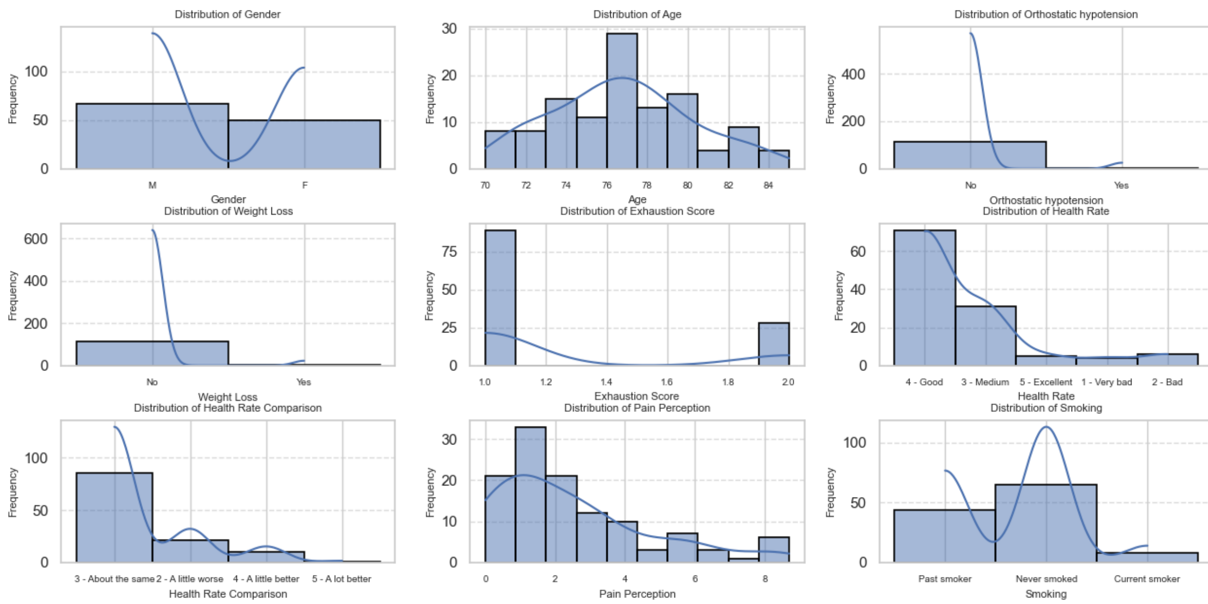


Figure 1-2: Categorical column distributions for better understanding. The line charts provides us the fluctuation of the data while bars show the data frequencies in relevant classifications.

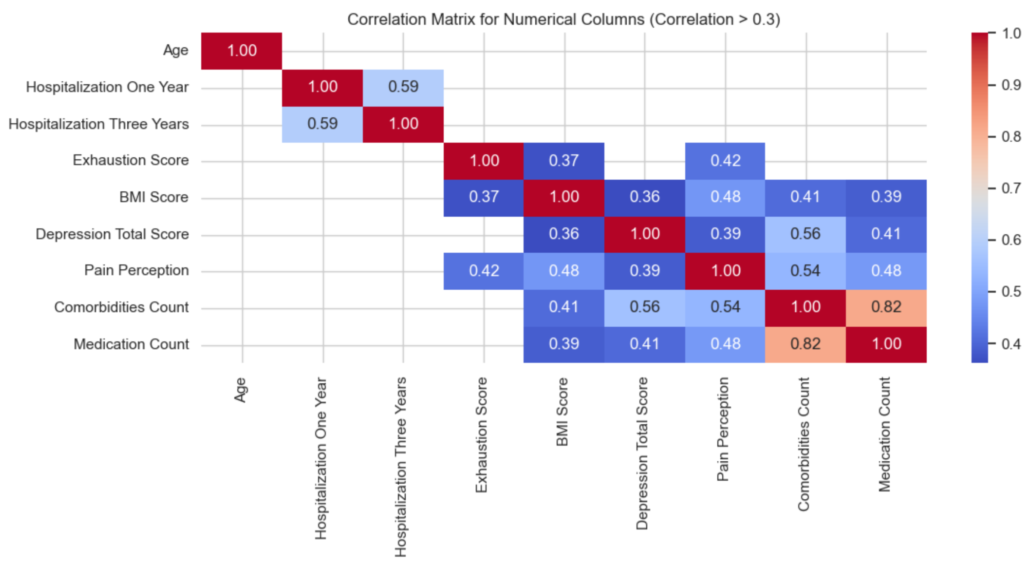


Figure 1-3: Correlation matrix for numerical columns to insight of the frailty of the data.

```
Descriptive Statistics:
              Age  Hospitalization One Year  Hospitalization Three Years  \
count  117.000000                117.000000                   117.000000
mean    76.726496                  0.239316                     0.598291
std      3.478069                  0.582053                     0.831076
min     70.000000                  0.000000                     0.000000
25%     74.000000                  0.000000                     0.000000
50%     77.000000                  0.000000                     0.000000
75%     79.000000                  0.000000                     1.000000
max     85.000000                  3.000000                     3.000000

       Exhaustion Score   BMI Score  Depression Total Score  Pain Perception  \
count        117.000000  117.000000              117.000000       117.000000
mean           1.239316   28.664850                2.256410         2.452137
std            0.428501    5.187214                2.009262         2.228677
min            1.000000   22.479339                0.000000         0.000000
25%            1.000000   24.744350                0.000000         1.000000
50%            1.000000   27.168115                2.000000         2.000000
75%            1.000000   29.788797                4.000000         3.500000
max            2.000000   44.658044                8.000000         8.700000

       Comorbidities Count  Medication Count  Depression Total Score Mapped
count           117.000000        117.000000                     117.000000
mean              4.487179          4.632479                       0.521368
std               3.390187          3.281606                       0.624200
min               0.000000          0.000000                       0.000000
25%               2.000000          2.000000                       0.000000
50%               4.000000          4.000000                       0.000000
75%               6.000000          7.000000                       1.000000
max              15.000000         15.000000                       2.000000
```

Figure 1-4: Descriptive statistics for numerical columns

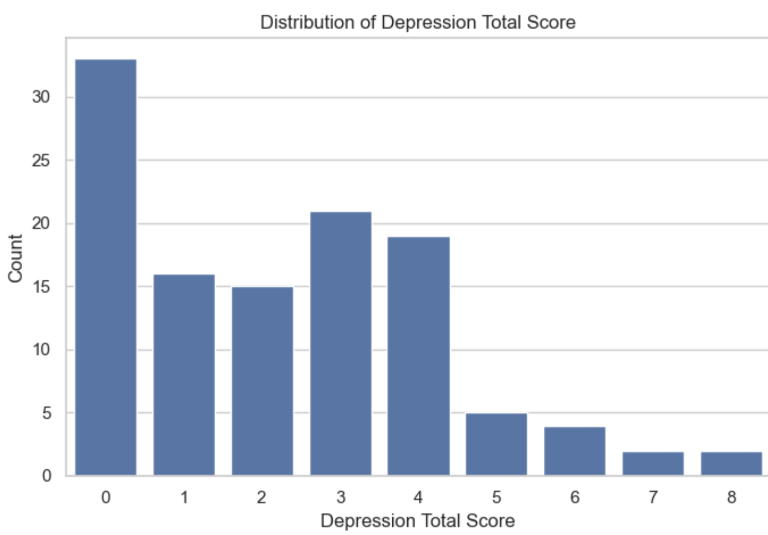


Figure 1-5: Dataset distribution, bar chart of Depression Total Score

CHAPTER TWO: PROBLEM SCOPE AND ALGORITHM SELECTION

We aim to find an approach to diminish the depression score of our patients for a better life quality by providing them further assistance in their healthcare. We believe that this assistance will not only be helping their mental well-beings but also their physical problems, either the reason of their admission to the facility or the underlying contagions.

During our studies, we learned many algorithms and many more from the side researches. The most suitable algorithms would be Decision Trees, Random Forests, K-Nearest neighbor, K-Means clustering, and Logistic regression. We will neglect the data size consideration for choosing the most suitable one. Due to generally suggested algorithms applications (figure 2-1) Logistic regression is first chosen. Sigmoid function will give us the most likelihood possibility for binary classification either to give additional assistance or not.

Algorithm	Suggested Data Size	Typical Output Types	Learning Type	Notes
Linear Regression	100s to 1000s	Continuous values	Supervised	Assumes a linear relationship in the data.
Multiple Linear Regression	100s to 1000s	Continuous values	Supervised	Extension of Linear Regression for multiple features.
Decision Trees	100s to 1000s	Classification or Regression	Supervised	Robust to outliers and non-linear relationships.
Random Forests	1000s to 10,000s	Classification or Regression	Supervised	Ensemble of decision trees.
Support Vector Machines	1000s to 10,000s	Classification or Regression	Supervised	Effective in high-dimensional spaces.
k-Nearest Neighbors	Small to Medium-sized (e.g., 1000s)	Classification or Regression	Supervised	Sensitive to irrelevant and redundant features.
Logistic Regression	100s to 10,000s	Binary Classification (0 or 1)	Supervised	Suitable for binary classification tasks.
Neural Networks	1000s to Millions	Classification or Regression	Supervised	Deep learning models may require large datasets.
K-Means Clustering	Any size	Cluster Assignments	Unsupervised	Divides data into clusters based on similarity.
Hierarchical Clustering	Any size	Dendrogram, Cluster Assignments	Unsupervised	Arranges data into a hierarchy of clusters.
Similarity and Dissimilarity Measures for Nominal Attributes	Any size	Distance or Similarity Values	Unsupervised	Measure dissimilarity/similarity between nominal attributes.
Similarity and Dissimilarity Measures - Levenshtein Distance	Any size	Distance Values	Unsupervised	Measures the edit distance between two strings.
Similarity and Dissimilarity Measures for Binary Attributes	Any size	Distance or Similarity Values	Unsupervised	Measure dissimilarity/similarity between binary attributes.
Similarity and Dissimilarity Measures for Numerical Attributes	Any size	Distance or Similarity Values	Unsupervised	Measure dissimilarity/similarity between numerical attributes.
Similarity and Dissimilarity Measures - Cosine Similarity	Any size	Similarity Values	Unsupervised	Measures the cosine of the angle between two vectors.
Dimensionality Reduction	Any size	Reduced Dimensionality Data	Unsupervised	Reduces the number of features while preserving information.

Figure 2-1: Best practices for algorithms

As Wade et al., 2015 had a research over a guided regularized random forests algorithm, the research assesses the accuracy of brain metrics to classify participants as pre- or post-ECT and distinguish between MDD and control participants in a matched cohort.

CHAPTER THREE: MODEL EVALUATION

Using data from 117 individuals, a predictive model is developed employing decision trees, logistic regression, and random forest. The random forest model achieves the highest accuracy at 70.83%. Smoking, BMI score, Age, and comorbidities count are found to be more influential than demographic factors in predicting depression. This study highlights the importance of a psychological support approach and suggests that a random forest is beneficial for establishing a comprehensive prediction model for depression in seniors' dataset.

Logistic regression gives us the results (figure 3-1) that low accuracy 0.58, so we need further investigation of the algorithms. So for considering the dataset size, we looked for Decision Trees to find relevant choices or the given features. We accomplished the results of 0.58 which is also the same as logistic regression version. After lack of reliable results, we searched for more complicated version, Random Forests algorithm, to see if the feature selection changes the results of the accuracy level. With multiple featured trees, this algorithm shows us the 0.71 accuracies for further assistance on the patients for decreasing their depression scores.

The supervised models helped us to find an approach for considering other features with different importance. What if we consider unsupervised models for further investigation? This will lead us to analyze the K-nearest neighbor and the K-Means algorithm's performance check. That investigation ended up with -0.09 and 0.24 relevantly. With these results, a supervised method will be more accurate for selecting accurate support for our patients.

```
C:\Users\x\PycharmProjects\pythonProject1\.venv\Scripts\python.exe C:\
Users\x\PycharmProjects\pythonProject1\Comparison_RealData.py
   Gender  Age  ...  Medication Count  Depression Total Score Mapped
0       1   78  ...                 5                              0
1       1   79  ...                 6                              0
2       1   79  ...                 6                              0
3       1   80  ...                 7                              1
4       0   72  ...                10                              0

[5 rows x 16 columns]
Accuracy (Logistic Regression (Balanced)): 0.5833

Confusion Matrix (Logistic Regression (Balanced)):
[[4 1 2 0 1 0 0 0]
 [1 2 0 0 0 0 0 0]
 [0 1 2 0 1 0 0 0]
 [0 0 0 2 0 1 0 0]
 [0 0 0 0 3 0 0 1]
 [0 0 0 0 1 0 0 0]
 [0 0 0 0 0 0 1 0]
 [0 0 0 0 0 0 0 0]]

Classification Report (Logistic Regression (Balanced)):
              precision    recall  f1-score   support

           0       0.80      0.50      0.62         8
           1       0.50      0.67      0.57         3
           2       0.50      0.50      0.50         4
           3       1.00      0.67      0.80         3
           4       0.50      0.75      0.60         4
           5       0.00      0.00      0.00         1
           6       1.00      1.00      1.00         1
           8       0.00       nan      0.00         0

    accuracy                           0.58        24
   macro avg       0.54      0.58      0.51        24
weighted avg       0.66      0.58      0.60        24

Accuracy (Decision Tree): 0.5833

Confusion Matrix (Decision Tree):
[[5 2 1 0 0 0 0]
 [0 3 0 0 0 0 0]
 [1 2 1 0 0 0 0]
 [0 0 0 2 0 1 0]
 [0 0 0 1 2 1 0]
 [0 0 0 0 1 0 0]
 [0 0 0 0 0 0 1]]

Classification Report (Decision Tree):
              precision    recall  f1-score   support

           0       0.83      0.62      0.71         8
Page 1 of 3
           1       0.43      1.00      0.60         3
           2       0.50      0.25      0.33         4
           3       0.67      0.67      0.67         3
           4       0.67      0.50      0.57         4
           5       0.00      0.00      0.00         1
           6       1.00      1.00      1.00         1

    accuracy                           0.58        24
   macro avg       0.59      0.58      0.56        24
weighted avg       0.65      0.58      0.59        24

Accuracy (Random Forest): 0.7083

Confusion Matrix (Random Forest):
[[7 1 0 0 0 0 0 0]
 [1 2 0 0 0 0 0 0]
 [0 1 2 1 0 0 0 0]
 [0 0 0 3 0 0 0 0]
 [0 0 0 1 2 0 0 1]
 [0 0 0 0 1 0 0 0]
 [0 0 0 0 0 0 1 0]
 [0 0 0 0 0 0 0 0]]

Classification Report (Random Forest):
              precision    recall  f1-score   support

           0       0.88      0.88      0.88         8
           1       0.50      0.67      0.57         3
           2       1.00      0.50      0.67         4
           3       0.60      1.00      0.75         3
           4       0.67      0.50      0.57         4
           5        nan      0.00      0.00         1
           6       1.00      1.00      1.00         1
           8       0.00       nan      0.00         0

    accuracy                           0.71        24
   macro avg       0.66      0.65      0.55        24
weighted avg       0.78      0.71      0.70        24

Silhouette Score (KNN): -0.0862
Silhouette Score (KMeans): 0.2384
Feature importances (Gini Index):
                          Feature  Gini Index
14  Depression Total Score Mapped    0.260294
11                        Smoking    0.148885
7                       BMI Score    0.147141
1                             Age    0.094447
12           Comorbidities Count    0.094005
10               Pain Perception    0.086960
9          Health Rate Comparison    0.049413
6               Exhaustion Score    0.046762
2         Hospitalization One Year    0.029831
3      Hospitalization Three Years    0.029258
Page 2 of 3
13               Medication Count    0.013003
0                          Gender    0.000000
4         Orthostatic hypotension    0.000000
5                     Weight Loss    0.000000
8                     Health Rate    0.000000
Accuracy: 0.7083333333333334

Confusion Matrix:
[[8 0 0 0 0 0 0 0]
 [1 2 0 0 0 0 0 0]
 [1 1 1 1 0 0 0 0]
 [0 0 0 3 0 0 0 0]
 [0 0 0 1 2 0 0 1]
 [0 0 0 0 1 0 0 0]
 [0 0 0 0 0 0 1 0]
 [0 0 0 0 0 0 0 0]]

Classification Report:
              precision    recall  f1-score   support

           0       0.80      1.00      0.89         8
           1       0.67      0.67      0.67         3
           2       1.00      0.25      0.40         4
           3       0.60      1.00      0.75         3
           4       0.67      0.50      0.57         4
           5        nan      0.00      0.00         1
           6       1.00      1.00      1.00         1
           8       0.00       nan      0.00         0

    accuracy                           0.71        24
   macro avg       0.68      0.63      0.53        24
weighted avg       0.78      0.71      0.68        24

Best Hyperparameters: {'max_depth': None, 'min_samples_leaf': 1, '
min_samples_split': 10, 'n_estimators': 50}

Process finished with exit code 0
```

Figure 3-1: Comparison of trained and evaluated models output

What if we had more data points with relevant mean values of the real dataset? Then we found out that predictions are not concluded as much as real data. The data distributions are normalized and not correlated to each other that our algorithms failed to show accurate predictions. Therefore we left this approach and continued with our patients analytics.

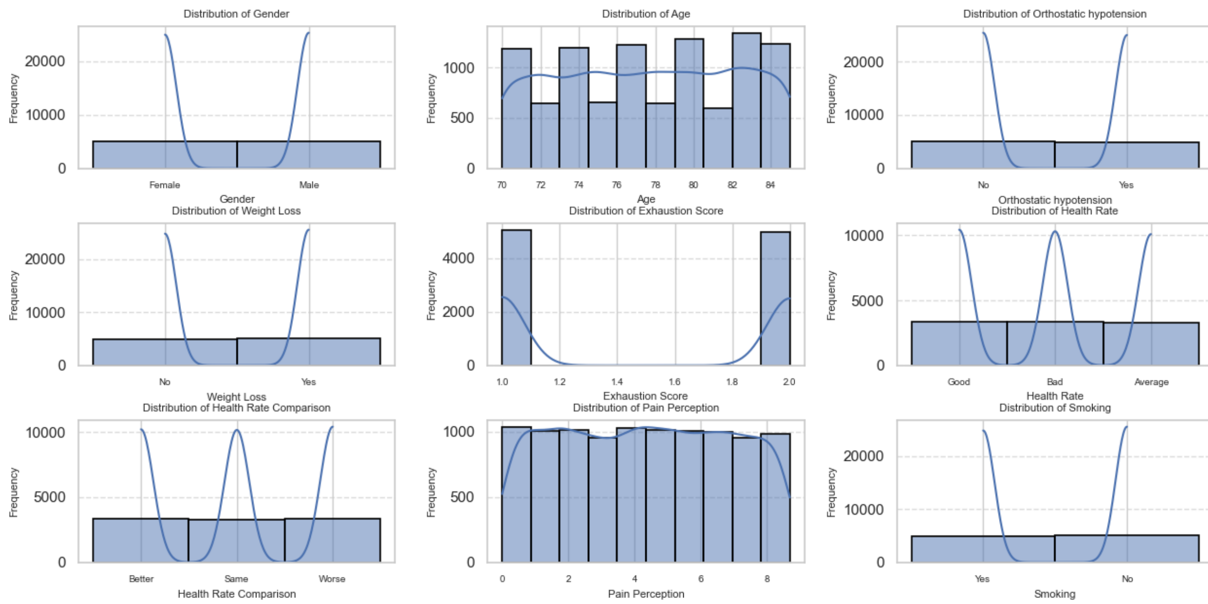


Figure 3-2: Faker Dataset analysis

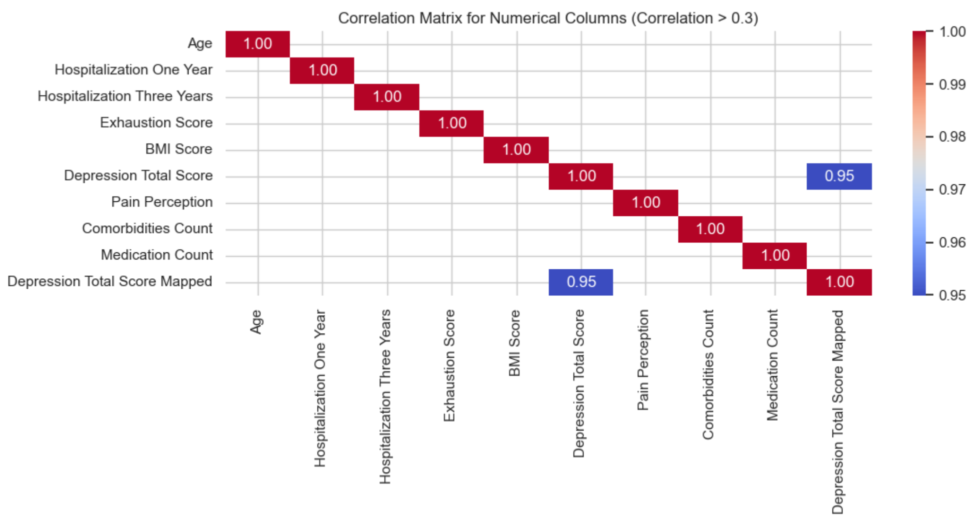


Figure 3-3: Correlation matrix for numerical columns to insight of the frailty of the Faker data.

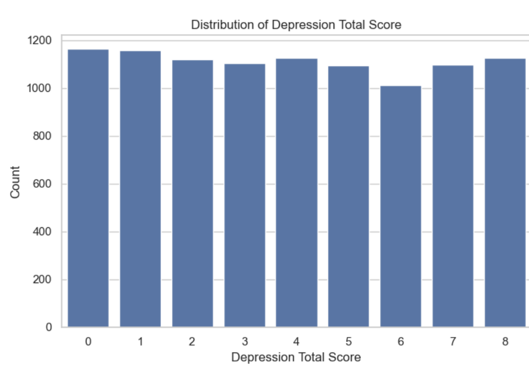


Figure 3-4: Dataset distribution, bar chart of Depression Total Score

Accuracy (Logistic Regression (Balanced)): 0.1120
Accuracy (Decision Tree): 0.1075
Accuracy (Random Forest): 0.1130
Silhouette Score (KNN): -0.0213
Silhouette Score (KMeans): 0.1799

Feature importances (Gini Index):

	Feature	Gini Index
7	BMI Score	0.176289
10	Pain Perception	0.164252
12	Comorbidities Count	0.126964
13	Medication Count	0.106193
1	Age	0.070334
2	Hospitalization One Year	0.060174
3	Hospitalization Three Years	0.058181
9	Health Rate Comparison	0.052485
8	Health Rate	0.037071
5	Weight Loss	0.033402
6	Exhaustion Score	0.030400
4	Orthostatic hypotension	0.029602
0	Gender	0.028751
11	Smoking	0.025902

Accuracy: 0.1085
Best Hyperparameters: {'max_depth': 20, 'min_samples_leaf': 1, 'min_samples_split': 10, 'n_estimators': 50}

Figure 3-5. Faker dataset results

CHAPTER FOUR: RESULTS ANALYSIS AND RECOMMENDATIONS

Results of our research indicates that the importance of data volume for a cohort analysis and altering parameters for better accuracy. High-dimensional shape features with a greater dataset and real-time observations of IoT's can be another method to use (Wade et al., 2015). When increasing well-being is considered, further investigation will be helpful for patients whereas providing us medical data and test results for more comprehensive understanding of their health and tailored treatment plans.

We may have a greater analysis with more data points since we had 117 patients set sampling. We may have investigated the Support Vector Machines algorithm for this classification problem. We tried to create a fake dataset from Python's faker function, but the real data was more accurate in the mines of non-random values. We changed the parameters such as the estimator and sample selection for deep understanding. In our pursuit of a deeper understanding, we adjusted parameters, including the estimator and sample selection. Despite these efforts, the outcomes indicated reduced accuracy, attributed to non-divisible breakdowns with no values under specific classifications. This candid acknowledgment of challenges in synthetic data creation and algorithmic selection contributes to a nuanced understanding of the complexities involved in obtaining accurate and meaningful results in healthcare analytics.

```
Accuracy (Random Forest): 0.5000

Confusion Matrix (Random Forest):
[[5 1 0 2 0 0 0 0]
 [1 2 0 0 0 0 0 0]
 [0 0 1 3 0 0 0 0]
 [2 0 0 1 0 0 0 0]
 [1 0 0 0 2 0 0 1]
 [0 0 0 0 1 0 0 0]
 [0 0 0 0 0 0 1 0]
 [0 0 0 0 0 0 0 0]]

Classification Report (Random Forest):
              precision    recall  f1-score   support

           0       0.56      0.62      0.59         8
           1       0.67      0.67      0.67         3
           2       1.00      0.25      0.40         4
           3       0.17      0.33      0.22         3
           4       0.67      0.50      0.57         4
           5        nan      0.00      0.00         1
           6       1.00      1.00      1.00         1
           8       0.00       nan      0.00         0

    accuracy                           0.50        24
   macro avg       0.58      0.48      0.43        24
weighted avg       0.64      0.50      0.51        24

Classifier Accuracy: 0.50
Accuracy: 0.5

Confusion Matrix:
[[6 1 0 1 0 0 0 0]
 [1 2 0 0 0 0 0 0]
 [0 0 1 3 0 0 0 0]
 [2 0 1 0 0 0 0 0]
 [1 0 0 0 2 0 0 1]
 [0 0 0 0 1 0 0 0]
 [0 0 0 0 0 0 1 0]
 [0 0 0 0 0 0 0 0]]

Classification Report:
              precision    recall  f1-score   support

           0       0.60      0.75      0.67         8
           1       0.67      0.67      0.67         3
           2       0.50      0.25      0.33         4
           3       0.00      0.00      0.00         3
           4       0.67      0.50      0.57         4
           5        nan      0.00      0.00         1
           6       1.00      1.00      1.00         1
           8       0.00       nan      0.00         0

    accuracy                           0.50        24
   macro avg       0.49      0.45      0.40        24
weighted avg       0.54      0.50      0.50        24

Best Hyperparameters: {'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}

Process finished with exit code 0
```

Figure 4-1:Random Forest with different sample split and estimators

CONCLUDING REMARKS

Machine learning techniques play a crucial role in disease detection, providing predictive capabilities for early diagnosis and informed decision-making in healthcare. In conclusion, this study emphasizes the crucial role of data volume and parameter differentiation in accurately predicting depression. Our aim is to decrease the levels of depression among physically healthier patients. Thus, we opted to use real-world data instead of synthetic data due to its richness and authenticity. Furthermore, we also trained our algorithms using synthetic data to enhance their performance. However, we found that this scenario was not effective under random variables due to inadequate values.

Overall, the study provides valuable insights into algorithm selection and model evaluation for addressing mental health issues in the healthcare domain. Based on our findings, we recommend further research with larger datasets and exploration of additional algorithms for a more comprehensive analysis. By doing so, we can gain a deeper understanding of the problem and develop more effective solutions to help individuals struggling with depression. These initiatives are essential to improving our understanding of patients' health and, eventually, to enable the creation of customized treatment regimens that support the overarching objective of improving overall health. It gives doctors more time to treat the disease. But even though doctors now have new tools to predict diseases more accurately, it can still take a long time to make a diagnosis. And sometimes, it's hard to know how accurate these predictions really are. So in the future, doctors and scientists need to keep improving the tools they use to predict diseases, so that they can diagnose them more quickly and accurately. By decreasing the depression scores of patients we will diminish the physical stress of the body to be.

BIBLIOGRAPHY

1. Arents, J. & Greitans, M. 2022, 'Smart Industrial Robot Control Trends, Challenges and Opportunities within Manufacturing', *Applied Sciences*, vol. 12, no. 2, pp. 937.
2. Bertrand, F. (2024) *Fbdesignpro/sweetviz: Visualize and compare datasets, target values and associations, with one line of code.*, *GitHub*. Available at: https://github.com/fbdesignpro/sweetviz (Accessed: 29 January 2024).
3. *Choosing the right estimator* (no date) *scikit*. Available at: https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html (Accessed: 30 January 2024).
4. Deltouzos, K. (2020) *Aggregated Virtual Patient Model Dataset*, *Zenodo*. Available at: https://zenodo.org/records/2670048#.Y9Y8fNJBwUE (Accessed: 29 January 2024).
5. Pedregosa, F. *et al.* (1970) *Scikit-Learn: Machine learning in Python*, *Journal of Machine Learning Research*. Available at: https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html (Accessed: 30 January 2024).
6. Wade, B.S.C. *et al.* (2015) *Random Forest classification of depression status based on subcortical brain morphometry following electroconvulsive therapy*, *Proceedings. IEEE International Symposium on Biomedical Imaging*. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4578162/ (Accessed: 30 January 2024).
7. Willett, K. *et al.* (2006) ‘Comparison of bioelectrical impedance and BMI in predicting obesity-related medical conditions’, *Obesity*, 14(3), pp. 480–490. doi:10.1038/oby.2006.63.
8. Üstün, T.B. *et al.* (2004) ‘Global burden of depressive disorders in the year 2000’, *British Journal of Psychiatry*, 184(5), pp. 386–392. doi:10.1192/bjp.184.5.386.

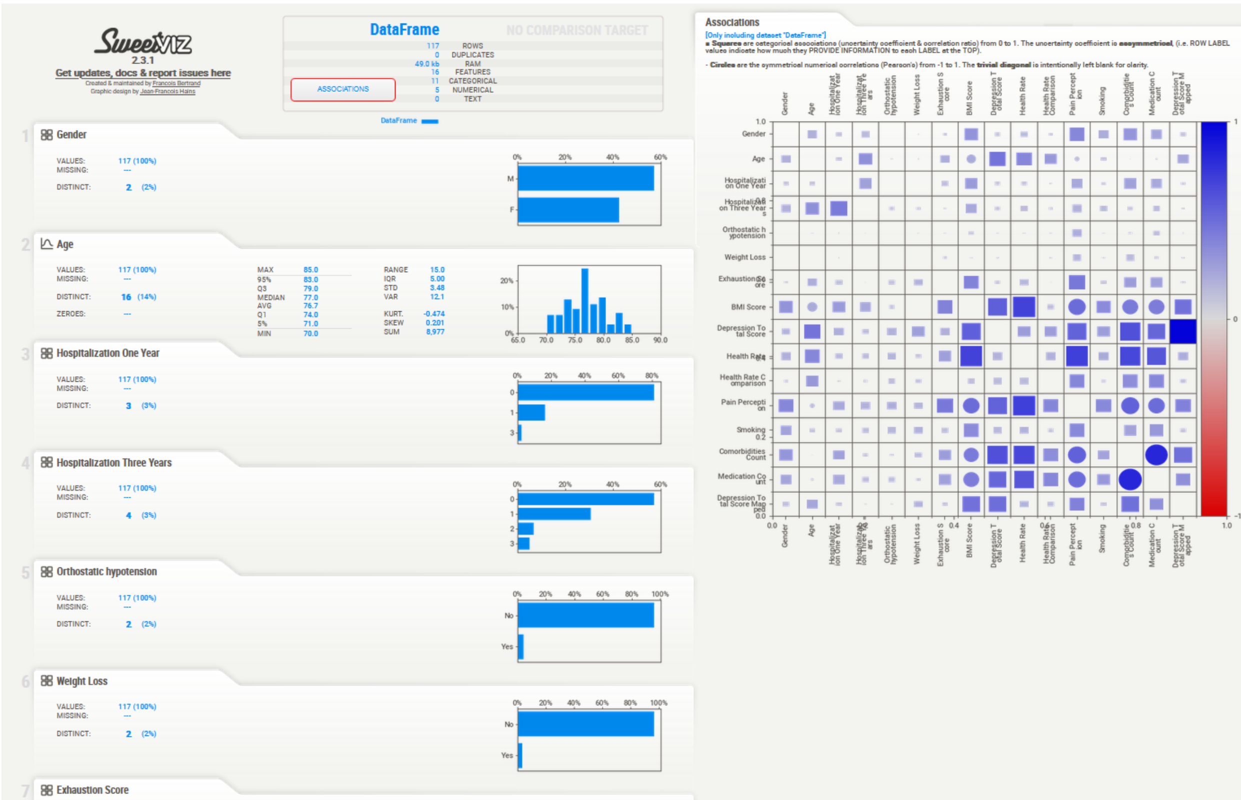
APPENDIX (if necessary)

Data Pre-Processing and Visualization code: EDA_RealData_DA.py

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import sweetviz as sv

# Load the dataset
m_df = pd.read_csv('Virtual Patient Models_Dataset.csv')

# Formatting
pd.set_option('display.max_columns', None)
# Set Seaborn style and color palette
sns.set(style="whitegrid")
sns.color_palette("flare", as_cmap=True)

# Drop columns with any missing values
m_df = m_df.dropna(axis=1)
#dropping unuseful numeric columns
selected_columns =
['gender','age','hospitalization_one_year','hospitalization_three_years','ort
ho_hypotension',
                     'weight_loss','exhaustion_score','bmi_score',
'depression_total_score',

'health_rate','health_rate_comparison','pain_perception','smoking',
'comorbidities_count',
                    'medication_count']
# Create a new DataFrame with only the selected columns
df = m_df[selected_columns]

# Renaming the existing DataFrame
df.rename(columns={'gender': 'Gender', 'age': 'Age',
'hospitalization_one_year': 'Hospitalization One Year',
                            'hospitalization_three_years': 'Hospitalization
Three Years',
                            'ortho_hypotension': 'Orthostatic
hypotension','weight_loss': 'Weight Loss',
                            'exhaustion_score': 'Exhaustion Score',
'bmi_score': 'BMI Score',
                            'depression_total_score': 'Depression Total
Score', 'health_rate': 'Health Rate',
                            'health_rate_comparison': 'Health Rate
Comparison', 'pain_perception': 'Pain Perception',
                             'smoking': 'Smoking', 'comorbidities_count':
'Comorbidities Count',
                     'medication_count': 'Medication Count'}, inplace=True)

# Mapping 'Smoking' values
smoking_mapping = {'Never Smoked': 'Never Smoked', 'Past smoker (stopped at
least 6 months)': 'Past smoker',
                   'Current smoker':'Current smoker' }
df['Smoking'] = df['Smoking'].replace(smoking_mapping)

# Create a new column based on the condition
df['Depression Total Score Mapped'] = df['Depression Total
Score'].apply(lambda x: 2 if x > 5 else (1 if x > 2 else 0))

# Display basic information about the dataset
```

sweetviz_report_DA_117.html

Data Visualization by Sweetviz function output (Bertrand, 2024)



RandomForest_DA.py

Random Forest Algorithm code:

```python
import pandas as pd
from sklearn.model_selection import train_test_split,GridSearchCV
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import
accuracy_score,confusion_matrix,classification_report
import numpy as np
import matplotlib.pyplot as plt
from sklearn.tree import plot_tree

# Load and Explore Data:
# Reading data from a CSV file
df =
pd.read_csv('C:/Users/x/PycharmProjects/pythonProject1/DA_preprocessed_VPM_da
taset_117.csv')

# Encode categorical variables
label_encoder = LabelEncoder()
df['Gender'] = label_encoder.fit_transform(df['Gender'])
df['Weight Loss'] = label_encoder.fit_transform(df['Weight Loss'])
df['Orthostatic hypotension'] = label_encoder.fit_transform(df['Orthostatic
hypotension'])
df['Health Rate'] = label_encoder.fit_transform(df['Health Rate'])
df['Health Rate Comparison'] = label_encoder.fit_transform(df['Health Rate
Comparison'])
df['Smoking'] = label_encoder.fit_transform(df['Smoking'])

print(df.head())

df = df.drop('Depression Total Score Mapped', axis=1)

# Separate features and target variable
X = df.drop('Depression Total Score',axis=1)
y = df['Depression Total Score']

# Split the data into training and testing sets
X_train,X_test,y_train,y_test =
train_test_split(X,y,test_size=0.2,random_state=42)

def train_evaluate_model(model,X_train,y_train,X_test,y_test,algorithm_name):
    model.fit(X_train,y_train)
    y_pred = model.predict(X_test)

    # Evaluate the model
    accuracy = accuracy_score(y_test,y_pred)
    print(f"Accuracy ({algorithm_name}): {accuracy:.4f}")

    # Print confusion matrix
    print(f"\nConfusion Matrix ({algorithm_name}):")
    print(confusion_matrix(y_test,y_pred))

    # Print classification report
    print(f"\nClassification Report ({algorithm_name}):")
    print(classification_report(y_test,y_pred, zero_division=np.nan))

# Train and evaluate random forest
random_forest = RandomForestClassifier(random_state=42,n_estimators=50)
```

Comparison of possible algorithms and their performance:

Comparison_RealData.py

```
import pandas as pd
from sklearn.model_selection import train_test_split,GridSearchCV
from sklearn.preprocessing import StandardScaler,LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.cluster import KMeans
from sklearn.metrics import
accuracy_score,silhouette_score,confusion_matrix,classification_report
import numpy as np
import matplotlib.pyplot as plt
from sklearn.tree import plot_tree

# Load and Explore Data:
# Reading data from a CSV file
df =
pd.read_csv('C:/Users/x/PycharmProjects/pythonProject1/DA_preprocessed_VPM_da
taset_117.csv')

'''
 0   Gender                         117 non-null    object
 1   Age                            117 non-null    int64
 2   Hospitalization One Year       117 non-null    int64
 3   Hospitalization Three Years    117 non-null    int64
 4   Orthostatic hypotension        117 non-null    object
 5   Weight Loss                    117 non-null    object
 6   Exhaustion Score               117 non-null    int64
 7   BMI Score                      117 non-null    float64
 8   Depression Total Score         117 non-null    int64
 9   Health Rate                    117 non-null    object
 10  Health Rate Comparison         117 non-null    object
 11  Pain Perception                117 non-null    float64
 12  Smoking                        117 non-null    object
 13  Comorbidities Count            117 non-null    int64
 14  Medication Count               117 non-null    int64
 15  Depression Total Score Mapped  117 non-null    int64
'''

# Encode categorical variables
label_encoder = LabelEncoder()
df['Gender'] = label_encoder.fit_transform(df['Gender'])
df['Weight Loss'] = label_encoder.fit_transform(df['Weight Loss'])
df['Orthostatic hypotension'] = label_encoder.fit_transform(df['Orthostatic
hypotension'])
df['Health Rate'] = label_encoder.fit_transform(df['Health Rate'])
df['Health Rate Comparison'] = label_encoder.fit_transform(df['Health Rate
Comparison'])
df['Smoking'] = label_encoder.fit_transform(df['Smoking'])

print(df.head())

# Separate features and target variable
X = df.drop('Depression Total Score',axis=1)
y = df['Depression Total Score']

# Split the data into training and testing sets
```

VPM_fakerData.py

Data generation for Faker function:

```
import csv
from faker import Faker
import random

fake = Faker()

with open('VPM_10000_Standardized.csv', 'w',newline='') as file:
    writer = csv.writer(file)
    field = ['Gender', 'Age', 'Hospitalization One Year', 'Hospitalization
Three Years', 'Orthostatic hypotension', 'Weight Loss',
             'Exhaustion Score', 'BMI Score', 'Depression Total Score',
'Health Rate', 'Health Rate Comparison', 'Pain Perception',
             'Smoking', 'Comorbidities Count', 'Medication Count']

    writer.writerow(field)

    for i in range(10000):
        writer.writerow([
            fake.random_element(elements=('Male','Female')),
            random.randint(70,85),
            random.randint(0,3),
            random.randint(0,3),
            fake.random_element(elements=('Yes','No')),
            fake.random_element(elements=('Yes','No')),
            random.randint(1,2),
            round(random.uniform(22.50,44.70), 2),
            random.randint(0,8),
            fake.random_element(elements=('Good','Bad','Average')),
            fake.random_element(elements=('Better','Same','Worse')),
            round(random.uniform(0.00,8.70), 2),
            fake.random_element(elements=('Yes','No')),
            random.randint(0,15),
            random.randint(0,15)
        ])
```

EDA_Faker_DA.py

Data Pre-Processing and Visualization code of Faker dataset:

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import textwrap

# Load the dataset
df = pd.read_csv('VPM_10000_Standardized.csv')

# Formatting
pd.set_option('display.max_columns', None)
# Set Seaborn style and color palette
sns.set(style="whitegrid")
sns.color_palette("flare", as_cmap=True)

# Drop columns with any missing values
m_df = df.dropna(axis=1)


# Create a new column based on the condition
df['Depression Total Score Mapped'] = df['Depression Total
Score'].apply(lambda x: 2 if x > 5 else (1 if x > 2 else 0))

# Display basic information about the dataset
print("Dataset Information:")
print(df.info())

# Distribution of categorical variables
categorical_cols = ['Gender','Age','Orthostatic hypotension','Weight
Loss','Exhaustion Score',
                    'Health Rate','Health Rate Comparison','Pain
Perception','Smoking' ]
    #df.select_dtypes(include='object'))
for col in categorical_cols:
    print(f"\n{col.capitalize()} Distribution:")
    print(df[col].value_counts())

# Set up the figure and axis for subplots with 2 rows
fig, axes = plt.subplots(nrows=3, ncols=len(categorical_cols)//3,
figsize=(12, 8))

# Flatten the axes array to make it easier to iterate
axes = axes.flatten()

# Iterate over columns and create histograms
for i, col in enumerate(categorical_cols):
    sns.histplot(df[col], bins=10, kde=True, ax=axes[i], edgecolor='black',
linewidth=1.2)
    axes[i].set_title(f'Distribution of {col}',fontsize=8)
    axes[i].set_xlabel(col,fontsize=8)
    axes[i].set_ylabel('Frequency',fontsize=8)
    axes[i].grid(axis='y',linestyle='--',alpha=0.7)

# Rotate x-axis labels vertically
    axes[i].tick_params(axis='x', labelsize=7)

plt.tight_layout()
plt.show()
```

Comparison of possible algorithms and their performance with Faker dataset:

Comparison of
FakerData_DA.py

```python
import pandas as pd
from sklearn.model_selection import train_test_split,GridSearchCV
from sklearn.preprocessing import StandardScaler,LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.cluster import KMeans
from sklearn.metrics import
accuracy_score,silhouette_score,confusion_matrix,classification_report
import numpy as np
import matplotlib.pyplot as plt
from sklearn.tree import plot_tree

# Load and Explore Data:
# Reading data from a CSV file
df = pd.read_csv('VPM_10000_Standardized.csv')

'''
 0   Gender                          117 non-null    object
 1   Age                             117 non-null    int64
 2   Hospitalization One Year        117 non-null    int64
 3   Hospitalization Three Years     117 non-null    int64
 4   Orthostatic hypotension         117 non-null    object
 5   Weight Loss                     117 non-null    object
 6   Exhaustion Score                117 non-null    int64
 7   BMI Score                       117 non-null    float64
 8   Depression Total Score          117 non-null    int64
 9   Health Rate                     117 non-null    object
 10  Health Rate Comparison          117 non-null    object
 11  Pain Perception                 117 non-null    float64
 12  Smoking                         117 non-null    object
 13  Comorbidities Count             117 non-null    int64
 14  Medication Count                117 non-null    int64
 15  Depression Total Score Mapped   117 non-null    int64
'''

# Encode categorical variables
label_encoder = LabelEncoder()
df['Gender'] = label_encoder.fit_transform(df['Gender'])
df['Weight Loss'] = label_encoder.fit_transform(df['Weight Loss'])
df['Orthostatic hypotension'] = label_encoder.fit_transform(df['Orthostatic
hypotension'])
df['Health Rate'] = label_encoder.fit_transform(df['Health Rate'])
df['Health Rate Comparison'] = label_encoder.fit_transform(df['Health Rate
Comparison'])
df['Smoking'] = label_encoder.fit_transform(df['Smoking'])

print(df.head())

# Separate features and target variable
X = df.drop('Depression Total Score',axis=1)
y = df['Depression Total Score']

# Split the data into training and testing sets
X_train,X_test,y_train,y_test =
train_test_split(X,y,test_size=0.2,random_state=42)
```