|  |
| --- |
| HelloBetter |
| Case Study Challenge |
| Machine Learning Engineer / Data Scientist |

|  |
| --- |
| Ouael Chabir  11-15-2023 |

**Tasks 1**

Starting a machine learning project, especially in the context of an adaptive trans-diagnostic intervention program, involves several structured steps. Here's a general outline:

1. **Understanding the domain and Defining the Problem**: The first step is to have a clear understanding of what the problem is and what you are trying to solve. This involves discussions with stakeholders (in this case, management and possibly healthcare professionals) to understand the objectives and desired outcomes of the project, and to understand the domain itself (**Adaptive Trans-diagnostic Intervention program**)
2. **Data Collection and Preparation**: Gather all relevant data. This would include the clinical data (psychological assessments) and behavioral platform data (like login frequency) as mentioned in the problem description. Data preparation includes cleaning, normalizing, and structuring the data for analysis. This step is essential to understand which data do we have in our datastore, and figure out if the existing data is enough to reach our target.
3. **Exploratory Data Analysis (EDA)**: Perform an initial investigation on your data to discover patterns, spot anomalies, test hypothesis, and check assumptions with the help of statistical summaries and graphical representations. This step is essential to conduct fruitful discussion with stakeholders and professionals.
4. **Feature Engineering and Selection**: Identify which data features are most relevant to the problem. This may involve creating new features from the existing data and selecting those that will be most useful for our model.
5. **Choosing the Right Model(s)**: Based on the problem at hand, decide on the appropriate machine learning algorithms.
6. **Training the Model**: Use the prepared dataset to train your model. This involves splitting the data into training, validation and test sets, and possibly using cross-validation techniques to validate the performance of the model.
7. **Model Evaluation and Tuning**: Evaluate the performance of your model using appropriate metrics. Based on the evaluation, we might need to go back and tweak the model parameters, features, or even the model itself.
8. **Deployment**: Once the model is ready and performing well, it's time to deploy it into our production environment where it can start providing insights or predictions. This could be integrating the model into your web/mobile application for real-time interventions.
9. **Monitoring and Maintenance**: After deployment, continuous monitoring is necessary to ensure the model performs as expected over time. The model may require updates or retraining as new data comes in or as the underlying data patterns change.
10. **Feedback Loop**: Implement a mechanism to gather feedback on the model's performance and its impact on the intervention program. Use this feedback for continuous improvement of the model.

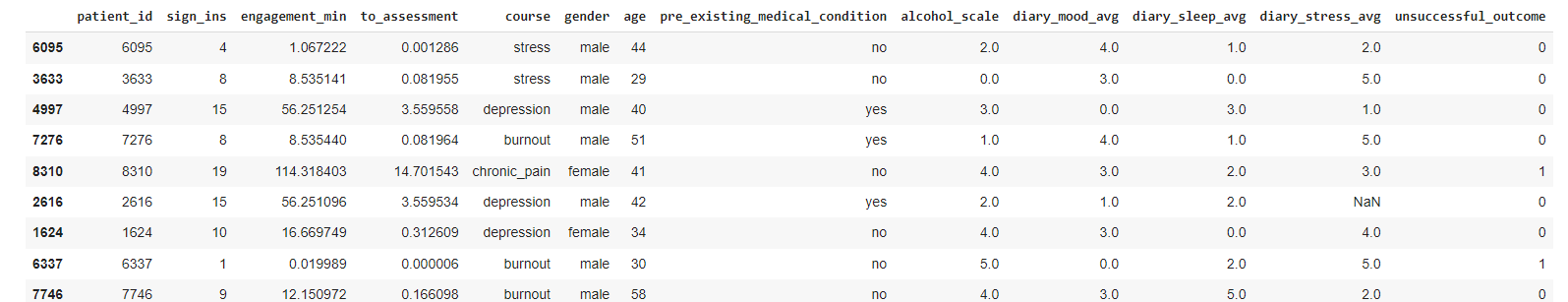
The success of a machine learning project heavily depends on clearly defined objectives, quality of data, and continuous iteration and improvement based on feedback and changing requirements.

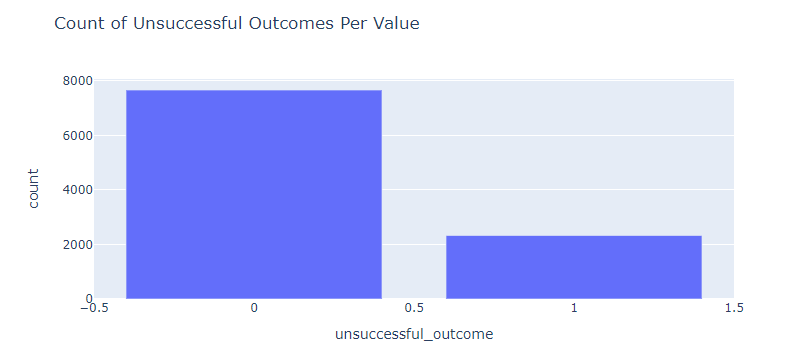
Task 2

1. **Understanding the Problem and Objective**

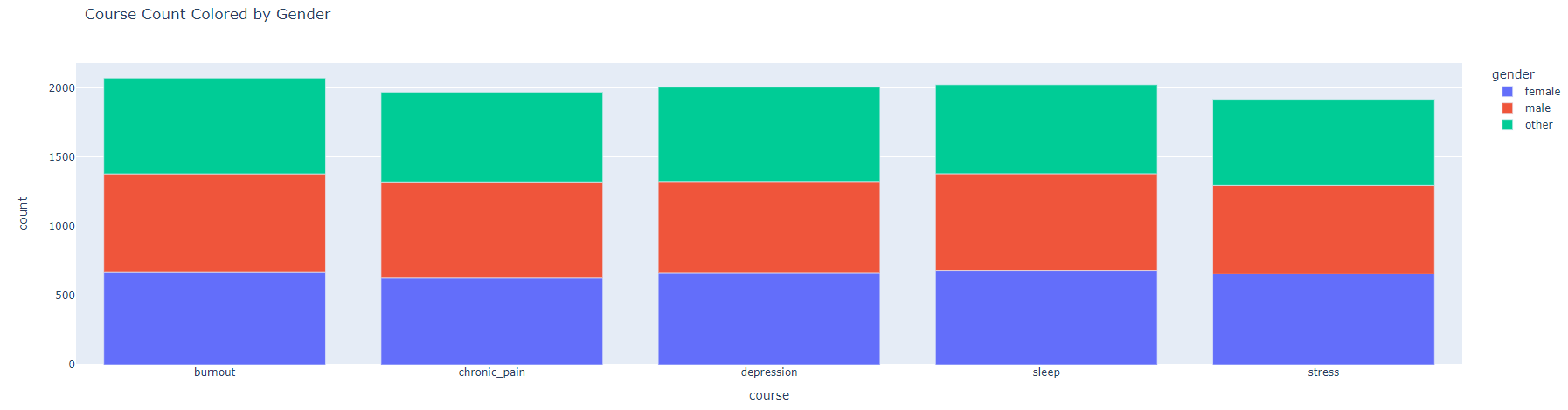
* **Objective**: Predict whether a patient will have a positive or negative outcome after completing a specific part of the intervention program.
* **Importance**: Identifying patients who might have a negative outcome allows for timely intervention, improving overall treatment effectiveness.

1. **Data Exploration and Understanding**

The provided dataset has the following **shape (10000, 13).** Below is a sample screenshot of the dataset:  


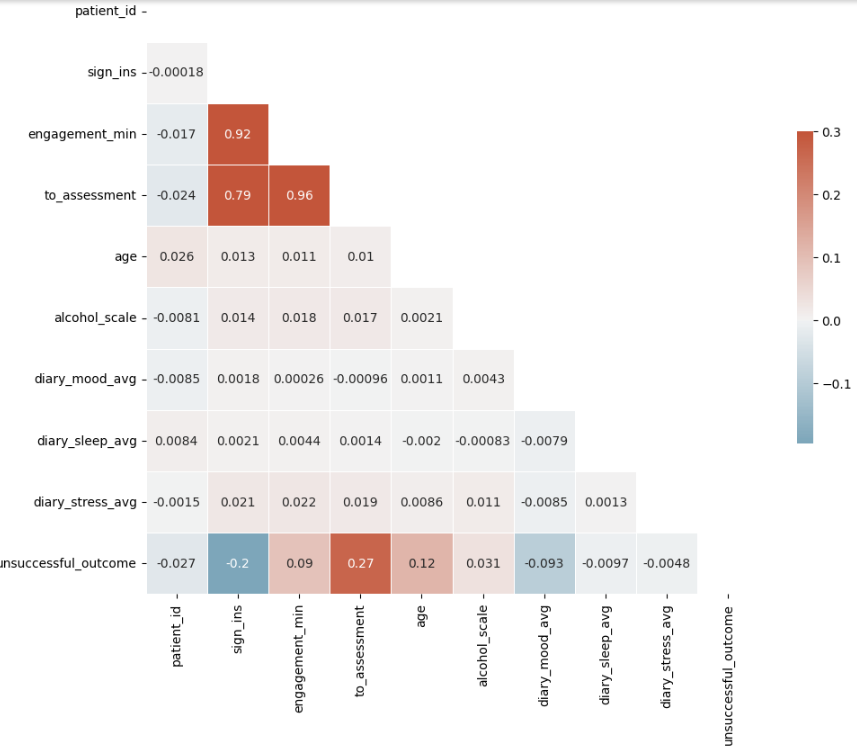
The target column is “**unsuccessful\_outcome**”. When we analyzed the number of records per each target value, we understood that we have an unbalanced dataset (see screenshot below). Having unbalanced dataset may cause the model to not being able predict under-represented class.  


We then tried to check the representation of courses per gender in the dataset, and we got the following plot. We can see that the course/gender ratio is well balanced in the provided dataset:



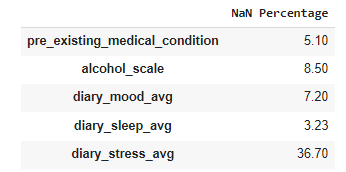
We also noticed a high correlation between the columns “engagement\_min”, “to\_assessment” and “sign\_ins”:

|  |  |  |
| --- | --- | --- |
|  |  |  |

We plotted the correlation matrix between variables:  


1. **Fill NaN values**

The dataset has NaN values in some columns as described in the table below:



To fill them, we trained a KNN model with k=5. The plots below describe the distribution of values before and after filling NaN values:

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

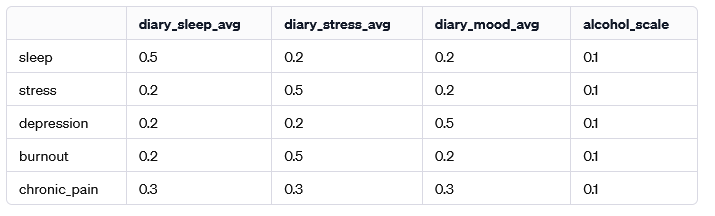
1. **Feature Engineering and Selection**

After analyzing the features provided by the dataset, we decided to select the following features:

* **Course**
* **Gender**
* **To\_assessment :** since the three variables “**engagement\_min”,** “**to\_assessment**” and “**sign\_ins**” are correlated, we will keep only “**to\_assessment**” since it provide the same information

We also engineered two other features:

* **combined\_to\_assessment\_age**:
  + we penalize the progress of a user provided by the variable “to\_assessment” by multiplying it by 0.01\*age. I assumed that age has huge impact on the person’s performance. And thus, even if the person makes progress in the program, age can be an obstacle to transform the progress into success in the program.
* **combined\_age\_medical\_condition**:
  + we assumed that having a medical condition can increase the logical age of the person. We penalize age by multiplying it by 1.5 in case of “pre\_existing\_medical\_condition“.
* **mass\_score**:
  + we introduced a new variable that combine the following variables **'alcohol\_scale**', **'diary\_mood\_avg'**, **'diary\_sleep\_avg'**, **'diary\_stress\_avg'.** MASS is the abbreviation of Mood, Alcohol, Stress, Sleep.
  + my research conducted me to understand how adaptive trans-diagnostic intervention program works. The idea of this methodology is to not target the disease itself, but its enablers instead.
  + The table below will help to understand how we calculate mass\_score. For each treatment program, we assign specific weight to each component. Alcohol is:



We ended-up by the following list of features to be provided to the model for training:

* **Course**
* **Gender**
* **Combined\_to\_assessment\_age**
* **Combined\_age\_medical\_condition**
* **Mass\_score**

1. **Feature transformation**

I applied log transformation on numerical features.

1. **Model training:**

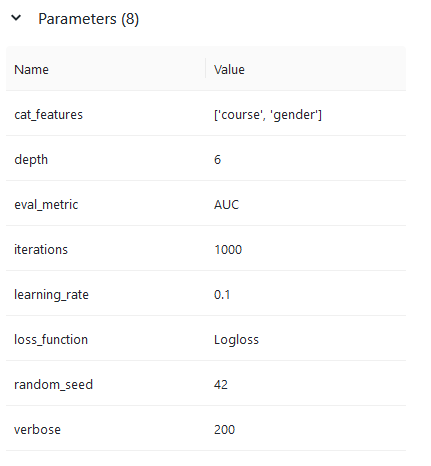
I choose CatBoost model for this purpose. CatBoost handles automatically categorical features (gender and Course) has a good performance.

I splitted the dataset into 3 sub-datasets:

* Training + Validation [cross-validation] (80%)
* Test (20%)

During the training phase, the model never seen the Test dataset.

Below is the list of parameters we set to the model:



After training we end-up with the following performance:

|  |  |
| --- | --- |
| ROC-AUC | 96.4 % |
| Accuracy | 92.1 % |
| Precision | 80.1 % |
| Recall | 88.9 % |
| F1 score | 84 % |

|  |  |
| --- | --- |
|  |  |
|  |  |

1. **Model evaluation**
2. **Deployment**