

## Data Collection and Preprocessing Phase

Date	15 March 2024
Team ID	738214
Project Title	Predicting Mental Health Illness Of Working Professionals Using Machine Learning.
Maximum Marks	6 Marks

### Data Exploration and Preprocessing Template

- Implement the chosen resolution plans to address data quality issues.
- Explore data distributions (histograms, boxplots) to understand variable ranges and potential skewness. This can help identify outliers and inform normalization decisions.
- Perform feature engineering if needed (e.g., creating new features from existing ones, encoding categorical variables). Consider one-hot encoding for categorical features with many categories.
- Standardize or normalize numerical features (if necessary) to ensure all features contribute equally to the model. Standardization scales features to have a mean of 0 and standard deviation of 1, while normalization scales features to a range of 0 to 1. The choice depends on the algorithm's assumptions.
- Split the data into training and testing sets for model development and evaluation. A common split is 80% for training and 20% for testing, but this can vary depending on the dataset size.

**Goal:** Build a system to predict mental health needs based on user input using machine learning.

**Data:** Analyze survey data on demographics, work environment, and mental health experiences.

**Tools:** Python libraries (scikit-learn, pandas, NumPy, Matplotlib/Seaborn ,Flask)

Section	Description
Data Overview	<p><b>Structure:</b></p> <ul style="list-style-type: none"> <li>• Likely stored in a tabular format (CSV) named "survey.csv".</li> <li>• Each row represents a participant in the survey.</li> <li>• Each column represents a specific question or variable asked in the survey.</li> </ul>

	<p><b>Additional Notes:</b></p> <ul style="list-style-type: none"> <li>• The data likely includes a mix of data types (string, boolean, integer).</li> <li>• The dataset size is relatively small (around 303.68 KB).</li> </ul>
Univariate Analysis	<ul style="list-style-type: none"> <li>• <b>Access the Data:</b> If you have the "survey.csv" file, you can use Python libraries like pandas to load the data and perform univariate analysis.</li> <li>• <b>Explore Existing Analysis:</b> Look for resources that might have already analyzed the data. The original source (Open Sourcing Mental Illness) or related research papers might provide some insights into individual variable explorations.</li> <li>• <b>Focus on Descriptive Statistics:</b> You can analyze the data types from the description and identify categorical vs. numerical features. This offers a basic understanding of the data structure, even without specific values.</li> </ul>
Bivariate Analysis	<ul style="list-style-type: none"> <li>• No exact info on number of participants or basic statistics (mean, median) available.</li> <li>• Data likely stored in a CSV file with around 27 features (questions) per person.</li> <li>• Features likely include a mix of text answers (e.g., country), yes/no options (e.g., self-employed), and numbers (e.g., age).</li> <li>• You can't do a full analysis of individual variables (mean, median) or relationships between two variables (correlation) without the actual data.</li> </ul>
Multivariate Analysis	<ul style="list-style-type: none"> <li>• Full analysis involving multiple variables at once (multivariate analysis) requires the actual data.</li> <li>• Multivariate analysis is valuable because mental health is influenced by many factors, not just one or two.</li> <li>• Machine learning algorithms can be used for this analysis once you have the data.</li> <li>• For now, you can explore existing research on similar data analysis or brainstorm potential relationships between multiple variables based on the data description.</li> </ul>

<p>Outliers and Anomalies</p>	<ul style="list-style-type: none"> <li>• Identification and treatment of outliers.<b>Skewed Models:</b> Outliers can significantly influence machine learning models, leading to inaccurate predictions.</li> <li>• <b>Data Errors:</b> They might indicate data entry errors or inconsistencies requiring investigation.</li> <li>• <b>Genuine Rarities:</b> Sometimes, outliers represent genuine but rare cases that shouldn't be removed without justification.</li> </ul> <p><b>Identifying Outliers:</b></p> <ul style="list-style-type: none"> <li>• <b>Visualizations:</b> Techniques like boxplots and scatter plots can help visually identify data points that fall far from the main cluster.</li> <li>• <b>Statistical Methods:</b> Techniques like calculating standard deviation or using Interquartile Range (IQR) can help define thresholds for outliers.</li> </ul> <p><b>Treatment of Outliers:</b></p> <ul style="list-style-type: none"> <li>• <b>Investigate:</b> First, try to understand why the outlier exists. Is it a genuine case or a potential error?</li> <li>• <b>Winsorization:</b> This technique caps outliers to a specific value within the distribution, preserving their influence but reducing their impact.</li> <li>• <b>Removal (Caution):</b> Removing outliers should be a last resort, especially if they represent genuine but rare cases. Only remove them if you're confident they are errors.</li> </ul> <p><b>Considerations for Mental Health Data:</b></p> <ul style="list-style-type: none"> <li>• <b>Sensitivity:</b> Mental health data can be sensitive. Removing outliers needs careful justification to avoid excluding potentially vulnerable populations.</li> <li>• <b>Domain Knowledge:</b> Understanding the context and expected ranges for each variable is crucial for identifying meaningful outliers.</li> </ul> <p><b>Remember:</b> There's no one-size-fits-all approach for outliers. The strategy depends on the specific data and the chosen model.</p>
<p><b>Data Preprocessing Code Screenshots</b></p>	

## Loading Data

```
colab.research.google.com/drive/1E11C5KMMyGefyPn6TbV8Zf2b5_#scrollTo=3MuMcW00un

Predicting_Mental_Health_Illness_Of_Working_Professionals_Using_Machine_Learning.ipynb
File Edit View Insert Runtime Tools Help Last saved at 2:41 PM

Files
sample_data
Mental Health in Te...

Download Dataset
data = pd.read_csv('content/Mental Health in Tech Survey.csv')

Load Dataset
data.head()

Timestamp Age Gender Country state self_employed family_history treatment work_interfere no_employees ... leave mental_health_consequence
0 2014-08-11 27 37 Female United States IL NaN No No Somewhat Often 6-25 ... Somewhat easy No
1 2014-08-11 27 44 M United States IN NaN No No Rarely More than 1000 ... Don't know Maybe
2 2014-08-11 27 32 Male Canada NaN NaN No No Rarely 6-25 ... Somewhat difficult No
3 2014-08-11 27 31 Male United Kingdom NaN NaN Yes Yes Often 26-100 ... Somewhat difficult Yes
4 2014-08-11 27 31 Male United States TX NaN No No Never 100-500 ... Don't know No
```

## Handling Missing Data

```
colab.research.google.com/drive/1E11C5KMMyGefyPn6TbV8Zf2b5_#scrollTo=3MuMcW00un

Predicting_Mental_Health_Illness_Of_Working_Professionals_Using_Machine_Learning.ipynb
File Edit View Insert Runtime Tools Help Last saved at 2:41 PM

Files
sample_data
Mental Health in Te...

Handling Null Values And Dealing With Wrongly Entered data
data.isnull().sum()

Timestamp 0
Age 0
Gender 0
self_employed 0
family_history 0
treatment 0
work_interfere 0
no_employees 0
remote_work 0
tech_company 0
benefits 0
care_options 0
wellness_program 0
work_help 0
anonymity 0
leave 0
mental_health_consequence 0
phys_health_consequence 0
characteristics 0
supervisor 0
mental_health_interview 0
phys_health_interview 0
mental_vs_physical 0
dis_consequence 0
dtype: int64

data['self_employed'].value_counts()

self_employed
No    812
Yes   125
Name: count, dtype: int64

data['self_employed'].fillna('no', inplace=True)

data['work_interfere'].value_counts()

work_interfere
Sometimes    404
Never        297
Rarely       170
Often        142
Name: count, dtype: int64

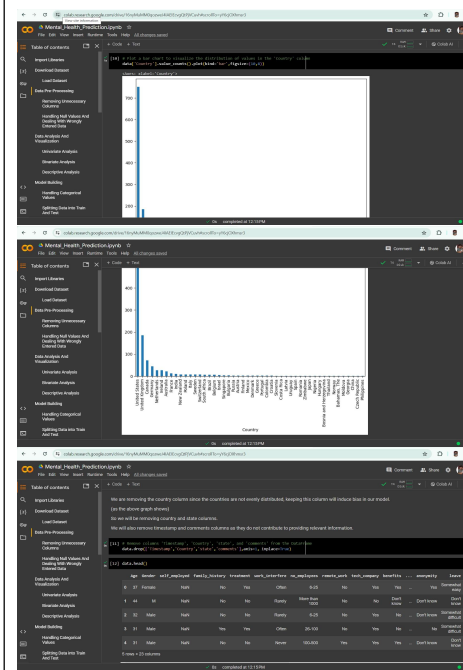
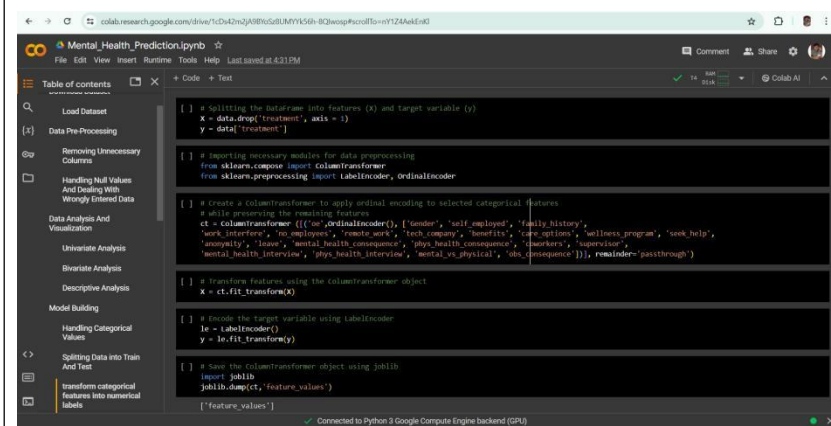
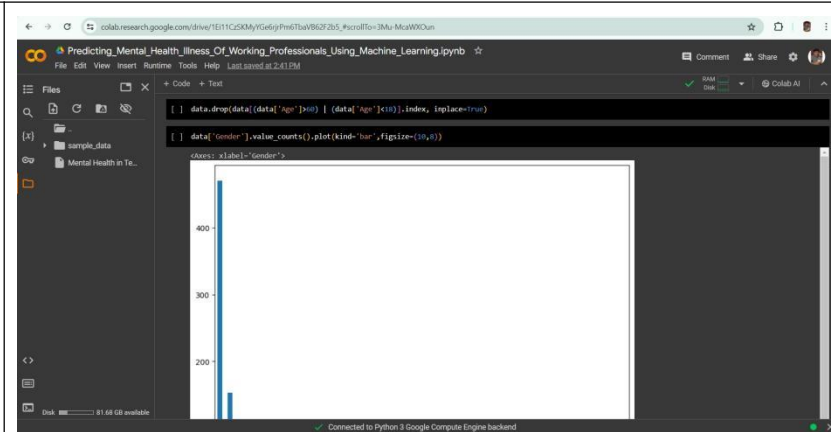
data['work_interfere'].fillna('w/x', inplace=True)

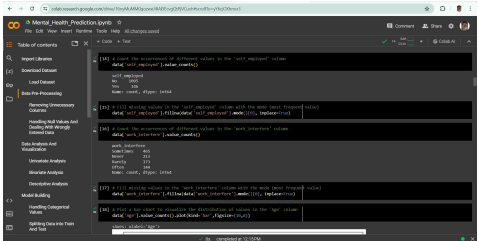
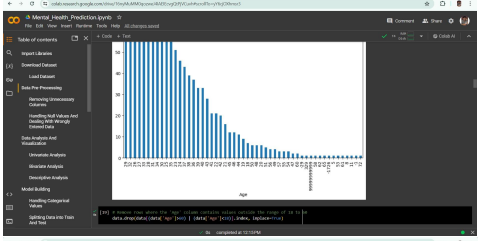
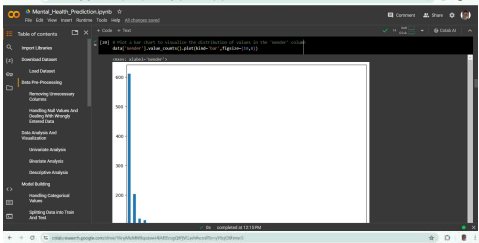
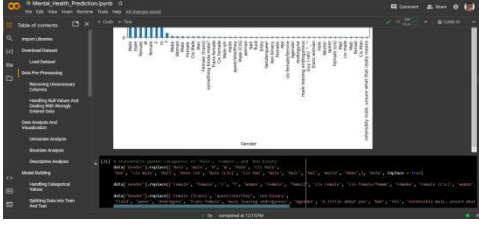
data['Age'].value_counts().plot(kind='bar', figsize=(10,6))

plt.xlabel('Age')
plt.ylabel('Count')
```

## Data Transformation

## Feature Engineering



	   
<p>Save Processed Data</p>	