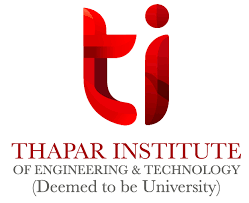
****

**THAPAR INSTITUTE OF ENGINEERING & TECHNOLOGY**

**PATIALA**

**PROJECT REPORT**

**DATA SCIENCE FOUNDATION**

**(PCS110)**

**TOPIC – DEPRESSION**

**SUBMITTED BY -**

**KANV**

**PRABHJIT KAUR (8024320071)**

**PRITAM**

**PRATEEM**

**PRABHPREET KAUR**

**ACKNOWLEDGMENT**

We are particularly indebted to the creators of the **Depression Dataset: A Comprehensive Dataset for Analyzing Health, Lifestyle, and Socio-Economic Factors**, from Kaggle whose diligent efforts in curating this rich dataset made this research possible. Their work has provided an invaluable resource for the study of mental health and its complex determinants.

We wish to acknowledge my colleagues and peers for their valuable input and collaboration during this project. Their critical discussions and suggestions have greatly enhanced the scope and depth of this research.

Lastly, I would like to express my gratitude to my family and friends for their continuous support and encouragement, which have strengthened me throughout this endeavor.

.

**DATA PREPROCESSING**

1. **IMPORT NECESSARY LIBRARIES**

We have imported all the required Libraries which *includes pandas, numpy, matplotlib.pyplot, seaborn, standard scalar, label encoder, Google Colab files.*

1. **UPLOAD DATASET**

**We uploaded our Depression Dataset, which was downloaded from Kaggle, to the google colab.**

1. **LOAD DATASET**

DataFrame *df* will contain the data from the CSV file in a structured tabular format with rows and columns.

1. **DATA EXPLORATION**

EDA involves understanding the dataset's structure, spotting anomalies, identifying patterns, and generating hypotheses that can guide further analysis or modeling efforts.

**The df.describe() method** is useful for understanding the distribution of numerical features. It allows you to see whether variables have outliers (e.g., a max value that’s significantly higher than the 75th percentile), or whether some features might need scaling.

1. **IDENTIFYING NULL VALUES**

It is essential for data cleaning, as missing values can affect data analysis and machine learning models, and often need to be handled appropriately (e.g., by filling or dropping them).

**DATA VISUALISATION**

1. **Visualize missing data** in your dataset with a heatmap, which provides an intuitive way to spot patterns of missing values.

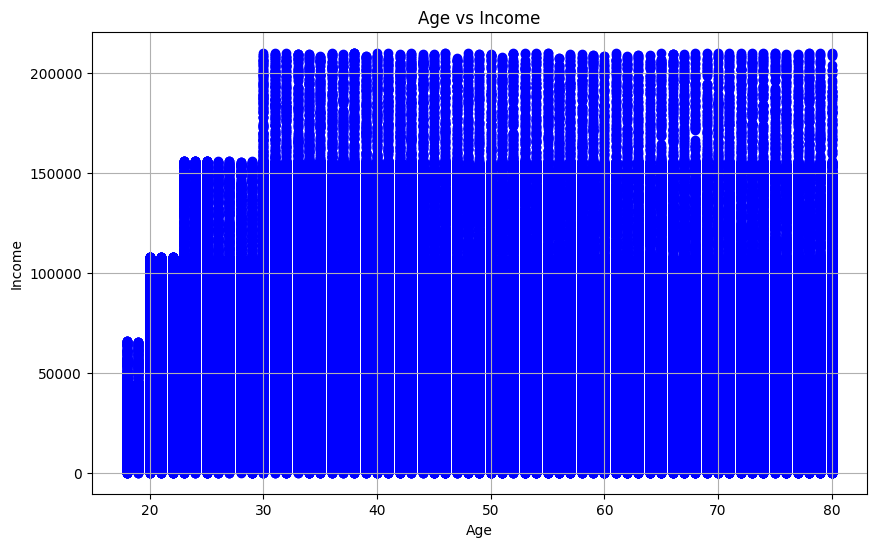
* **Vertical Stripes**: If entire columns have missing values, you will see vertical lines in the heatmap.
* **Horizontal Stripes**: If specific rows have missing values, you will see horizontal lines.
* **Random Patches**: If missing data is scattered, you’ll see isolated cells marked.

1. **Label Encoding** to transform categorical variables into numerical form. This is necessary because most machine learning algorithms work only with numerical data.

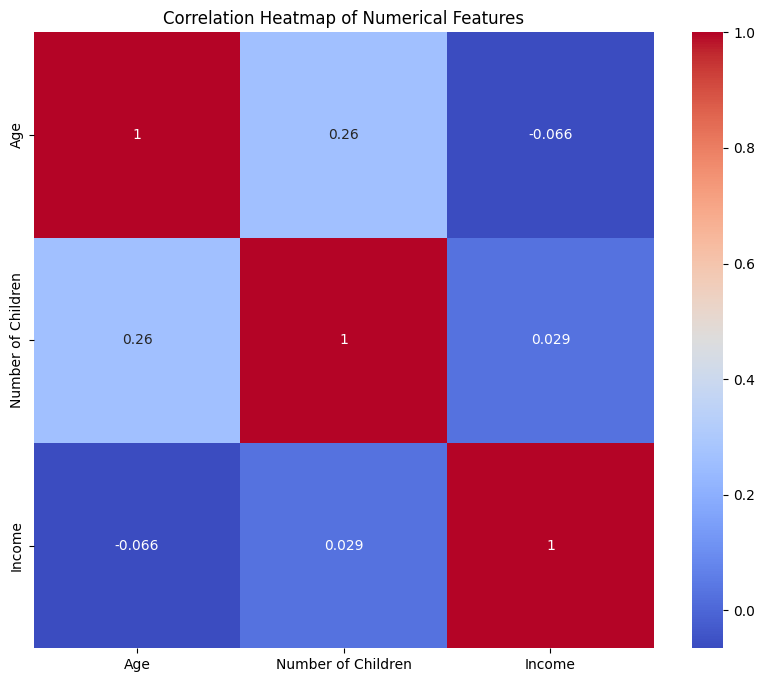
For each column, the categories (which are non-numeric, like "Yes" or "No" or other categorical strings) are transformed into integers. For example:

* "Married" → 0
* "Single" → 1
* "Divorced" → 2

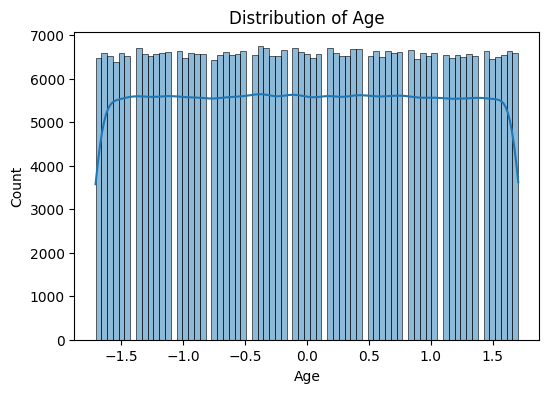
1. **Plotting scatter plot** to visualize the relationship between two variables in the dataset: **Age** and **Income**. A scatter plot is useful for identifying any patterns, correlations, or trends between the two variables.

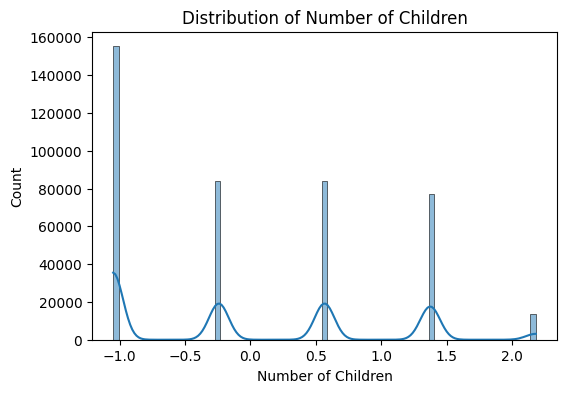
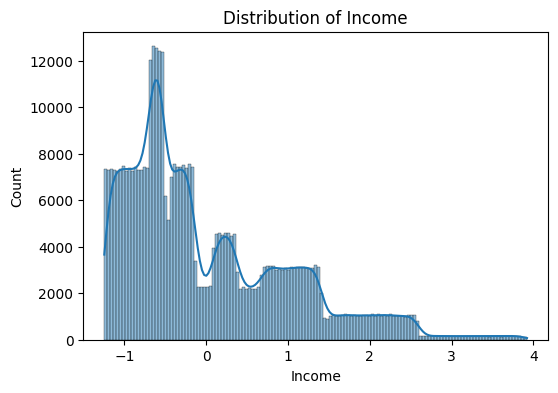
****

1. Applying **standardization** to the numerical features: **Age**, **Income**, and **Number of Children**, using the StandardScaler from sklearn.preprocessing.
2. **Correlation heatmap** to visualize the relationships between numerical features in the dataset. A correlation heatmap helps in understanding the strength and direction of linear relationships between variables.

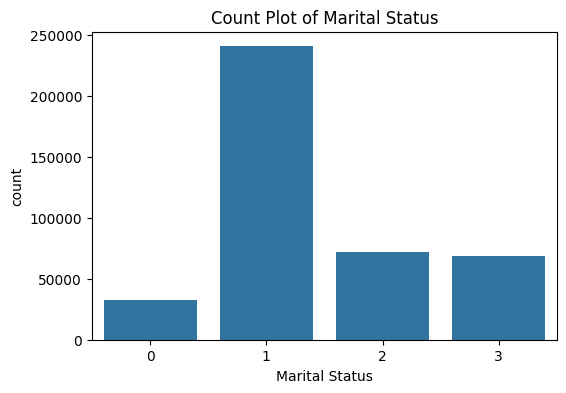
****

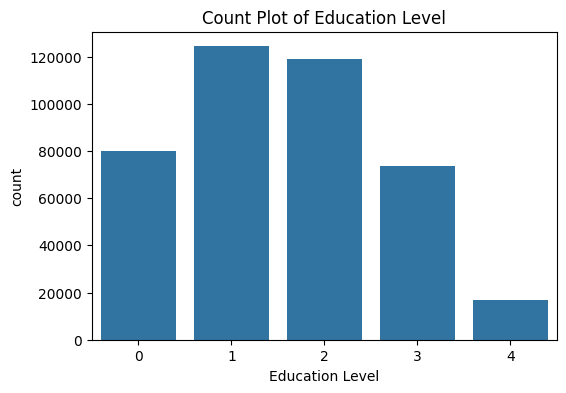
1. **Plotting Histogram** providing a visualization of the **distribution** of each numerical variable. It helps in understanding how the data is spread across different values (e.g., normal, skewed, etc.).

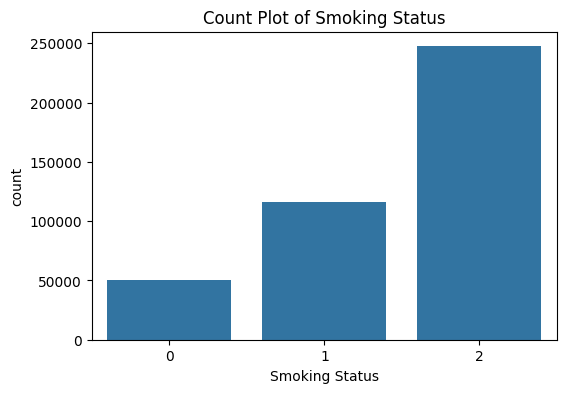
****

****

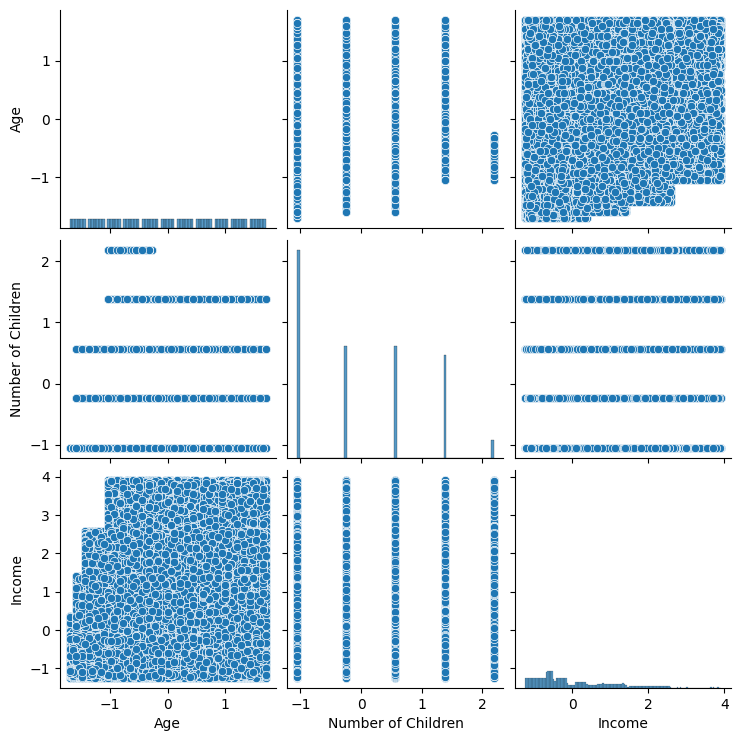
1. **Count plots** for each categorical feature in the dataset. A **count plot** displays the frequency of each unique category in a feature, making it easy to visualize how many observations belong to each category, different count plots plotted in respective to the data set. Example-

****

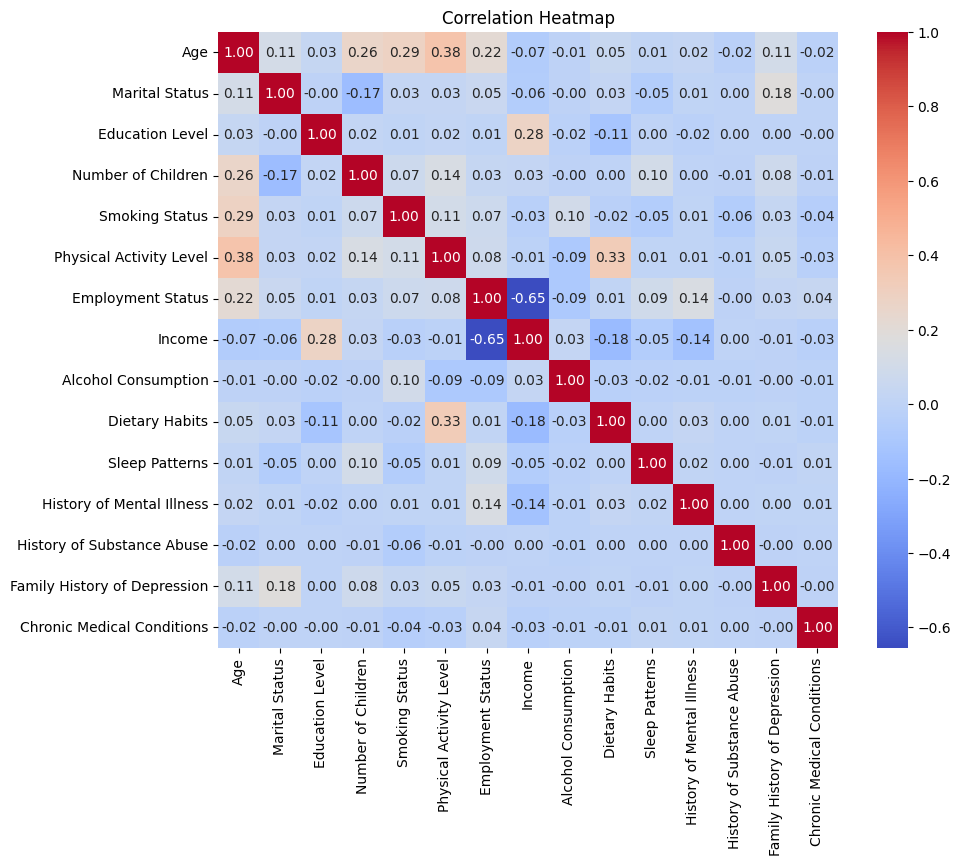
****

****

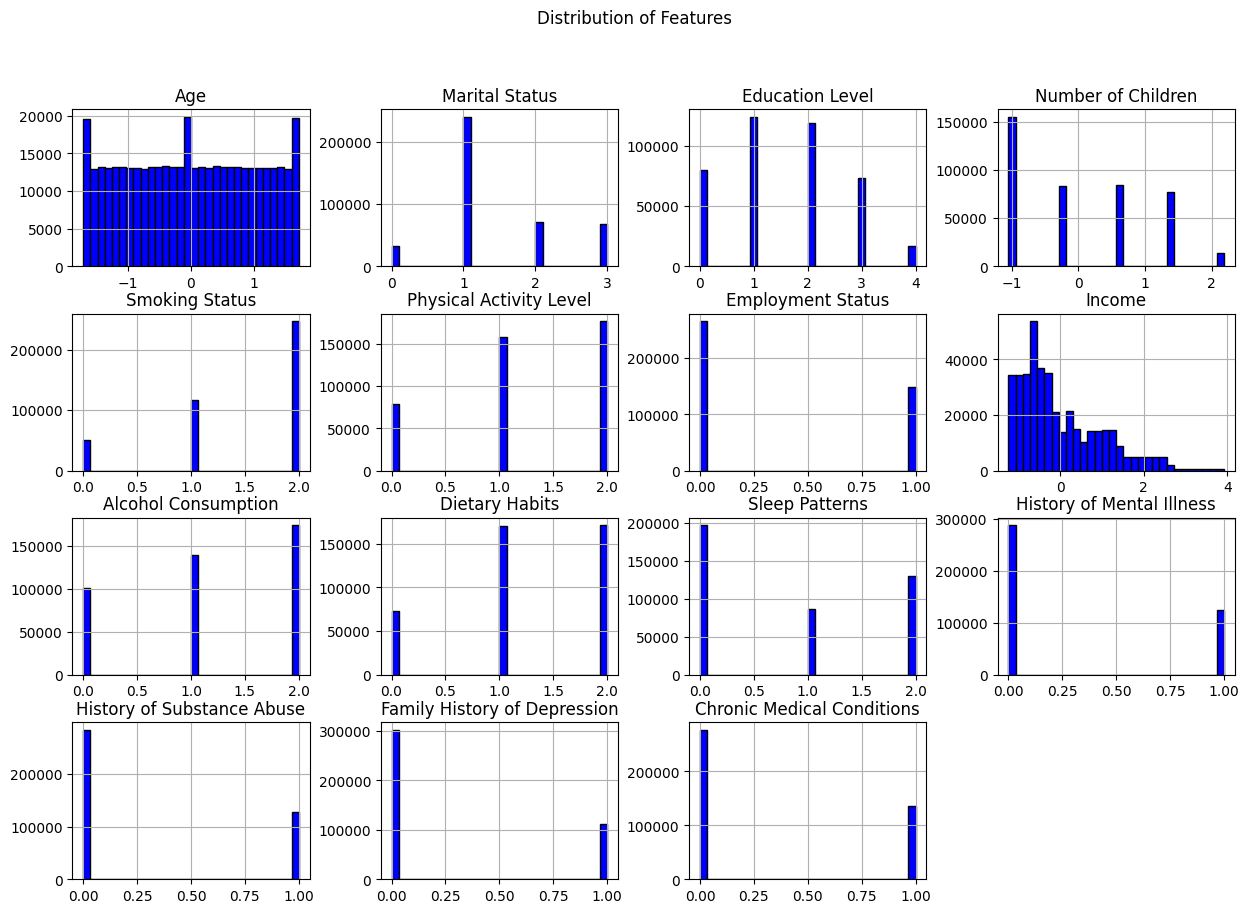
1. **Pairplot** for all numerical features in the dataset using Seaborn. A **pairplot** is a useful visualization tool that provides a grid of scatter plots for each pair of numerical features in the dataset, along with histograms or density plots along the diagonal.

****

1. **Correlation matrix heatmap** for all features in the dataset using Seaborn. A correlation matrix provides a numerical representation of the relationships between variables, showing how they are correlated with each other.

****

1. **Plotting Histograms** for all numerical columns in the dataset using Pandas' built-in plotting capabilities. Histograms provide a visual representation of the distribution of numerical data, allowing you to understand the frequency of data points within specified ranges (bins).

****