TITLE: Mental Health Monitoring via Social Media

Introduction

In recent years, mental health issues such as depression, anxiety, and stress have become increasingly prevalent in modern society, affecting individuals across diverse age groups and social backgrounds. Despite growing awareness, many people continue to suffer in silence due to social stigma, lack of understanding, or limited access to professional care. At the same time, social media platforms have evolved into spaces where individuals frequently share their emotions, experiences, and personal struggles more openly than in face-to-face interactions.

These digital expressions provide a unique window into the psychological state of users, offering valuable insights into their mental well-being. By analyzing patterns in social media activity, researchers can better understand how online behavior reflects emotional and psychological conditions. This connection between social media use and mental health has opened new pathways for studying human behavior, emotional expression, and the societal impact of technology on mental well-being.

Step - 1: Setting the Research Goal

The primary objective of this research is to develop a data-driven system capable of identifying early warning signs of mental health issues such as depression, anxiety, and stress through the analysis of social media content. By leveraging user's online behavior, language patterns, and engagement trends, the system aims to detect subtle indicators of emotional distress that often go unnoticed in traditional health assessments.

This research focuses on integrating machine learning techniques and statistical analysis to establish correlations between social media activity and mental health conditions. The goal is to create a predictive framework that can analyze user data, classify mental health states, and support early interventions.

Step - 2: Retrieving Data

The data for this research is obtained from publicly available social media–related mental health datasets that capture user behavior, emotional expression, and engagement patterns. Platforms like Kaggle provide structured datasets containing user demographics, social media usage habits, self-reported mental health indicators, and survey responses.

Features:

- age, gender, relationship_status, occupation_status, affiliation
- social_media_usage, social_media_platforms, daily_social_media_time
- purposeless_use, distraction_by_social_media, restless_without_social_media, distraction_score, worry_score, concentration_difficulty, social_comparison, feeling_about_comparisons, seeking_validation
- activity_interest, sleep_issues
- **Target Variable:** depression_score

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import pearsonr
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix, roc_auc_score, accuracy_score, ConfusionMatrixDisplay
```

```
df = pd.read_csv('Social_media.csv')
df.head()
```

	1. What is your age?	2. Gender	3. Relationship Status	4. Occupation Status	5. What type of organizations are you affiliated with?	6. Do you use social media?	7. What social media platforms do you commonly use?	8. What is the average time you spend on social media every day?	9. How often do you find yourself using Social media without a specific purpose?	10. How often do you get distracted by Social media when you are busy doing something?	11. Do you feel restless if you haven't used Social media in a while?	12. On a scale of 1 to 5, how easily distracted are you?	13. On a scale of 1 to 5, how much are you bothered by worries?	14. Do you find it difficult to concentrate on things?
0	21.0	Male	In a relationship	University Student	University	Yes	Facebook, Twitter, Instagram, YouTube, Discord...	Between 2 and 3 hours	5	3	2	5	2	5
1	21.0	Female	Single	University Student	University	Yes	Facebook, Twitter, Instagram, YouTube, Discord...	More than 5 hours	4	3	2	4	5	4
2	21.0	Female	Single	University Student	University	Yes	Facebook, Instagram, YouTube, Pinterest	Between 3 and 4 hours	3	2	1	2	5	4
3	21.0	Female	Single	University Student	University	Yes	Facebook, Instagram	More than 5 hours	4	2	1	3	5	3
4	21.0	Female	Single	University Student	University	Yes	Facebook, Instagram, YouTube	Between 2 and 3 hours	3	5	4	4	5	5

Step - 3: Data Preparation

The goal of data preparation is to ensure that the dataset is clean, consistent, and ready for analysis or model training. Data cleansing involves removing missing, duplicate, or invalid entries to ensure the dataset is accurate and reliable. Data transformation involves converting data into the right format or scale for analysis or machine learning. This includes renaming columns, converting categorical data into numeric form, and standardizing units. If the dataset was split across multiple tables, this step would involve merging or joining datasets into a single structured DataFrame.

The dataset includes information such as age, gender, occupation status, relationship status, social media usage habits, daily screen time, and mental health indicators like worry score, distraction score, and depression score. These attributes help identify relationships between social media behavior and mental health conditions such as stress, anxiety, and depression.

Data Cleansing

Key tasks include:

- Handling missing values
- Removing unnecessary columns
- Removing duplicates
- Ensuring data consistency

This step deals with removing missing, duplicate, or invalid data entries. For example, empty cells in the column *"Affiliation"* are filled using the most frequent value (mode), while unnecessary columns like *"Timestamp"* are dropped to improve dataset quality.

```
# Check for missing data in the entire dataset
df.isnull().sum()
```

```
# Fill missing data using the mode
df['5. What type of organizations are you affiliated with?'] = df['5. What type of organizations are you affiliated with?'].fillna(d
df.isnull().sum()
```

```
# Delete unnecessary column information

df.drop('Timestamp', axis = 1, inplace = True)
df.head()
```

Data Transformation

Data transformation involves converting data into the right format or scale for analysis or machine learning. This includes renaming columns, converting categorical data into numeric form, and standardizing units.

Key tasks include:

- Renaming columns for readability
- Converting text data (e.g., "Yes"/"No") to numeric values (1/0)
- Converting text-based time data into numeric hours

Since some responses are in text format, they must be converted into numerical values for modeling. For instance, *"Yes"* and *"No"* in *"Do you use social media?"* are encoded as 1 and 0. Similarly, the column *"Daily social media time"* is standardized to numerical hours (e.g., "Less than an hour" = 0.5, "Between 2–3 hours" = 2.5).

```
new_column_names = {
    '1. What is your age?': 'age',
    '2. Gender': 'gender',
    '3. Relationship Status': 'relationship_status',
    '4. Occupation Status': 'occupation_status',
    '5. What type of organizations are you affiliated with?': 'affiliation',
    '6. Do you use social media?': 'social_media_usage',
    '7. What social media platforms do you commonly use?': 'social_media_platforms',
    '8. What is the average time you spend on social media every day?': 'daily_social_media_time',
    '9. How often do you find yourself using Social media without a specific purpose?': 'purposeless_use',
    '10. How often do you get distracted by Social media when you are busy doing something?': 'distraction_by_social_media',
    '11. Do you feel restless if you haven\'t used Social media in a while?': 'restless_without_social_media',
    '12. On a scale of 1 to 5, how easily distracted are you?': 'distraction_score',
    '13. On a scale of 1 to 5, how much are you bothered by worries?': 'worry_score',
    '14. Do you find it difficult to concentrate on things?': 'concentration_difficulty',
    '15. On a scale of 1-5, how often do you compare yourself to other successful people through the use of social media?': 'social_comparison',
    '16. Following the previous question, how do you feel about these comparisons, generally speaking?': 'feeling_about_comparisons',
    '17. How often do you look to seek validation from features of social media?': 'seeking_validation',
    '18. How often do you feel depressed or down?': 'depression_score',
    '19. On a scale of 1 to 5, how frequently does your interest in daily activities fluctuate?': 'activity_interest',
    '20. On a scale of 1 to 5, how often do you face issues regarding sleep?': 'sleep_issues'
}
df.rename(columns = new_column_names, inplace = True)
df.drop_duplicates(inplace=True)
df['depression_score'] = df['depression_score'].replace([np.inf, -np.inf], np.nan)
df = df.dropna(subset=['depression_score'])
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 481 entries, 0 to 480
Data columns (total 20 columns):
 #   Column                         Non-Null Count  Dtype  
---  ------                         --------------  -----  
 0   age                            481 non-null    float64
 1   gender                         481 non-null    object 
 2   relationship_status            481 non-null    object 
 3   occupation_status              481 non-null    object 
 4   affiliation                    481 non-null    object 
 5   social_media_usage             481 non-null    object 
 6   social_media_platforms         481 non-null    object 
 7   daily_social_media_time        481 non-null    object 
 8   purposeless_use                481 non-null    int64  
 9   distraction_by_social_media    481 non-null    int64  
 10  restless_without_social_media  481 non-null    int64  
 11  distraction_score              481 non-null    int64  
 12  worry_score                    481 non-null    int64  
 13  concentration_difficulty       481 non-null    int64  
 14  social_comparison              481 non-null    int64  
 15  feeling_about_comparisons      481 non-null    int64  
 16  seeking_validation             481 non-null    int64  
 17  depression_score               481 non-null    int64  
 18  activity_interest              481 non-null    int64  
 19  sleep_issues                   481 non-null    int64  
dtypes: float64(1), int64(12), object(7)
memory usage: 75.3+ KB
```

Combining Data

In this project, there is only one dataset being used, so no merging or integration from other data sources is required. All relevant features like demographic, behavioral, and psychological are already included in the same file. Therefore, this step primarily confirms that the data is complete and well-structured in a single dataset for modeling.

Step - 4: Data Exploration

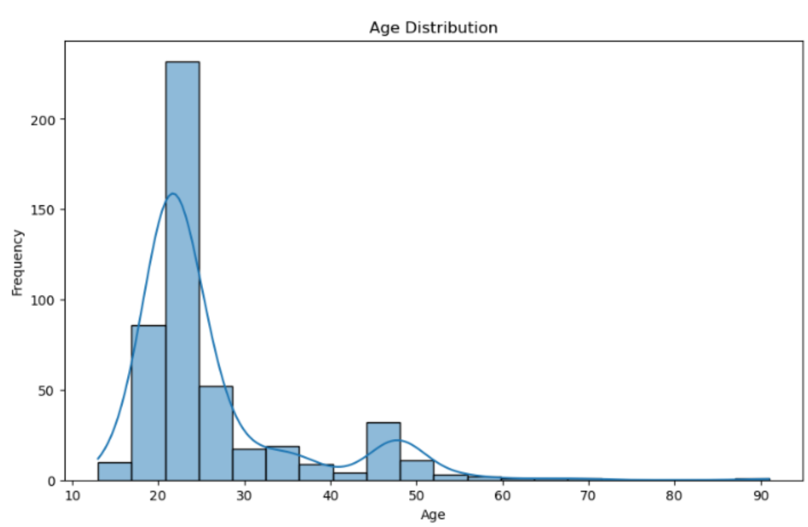
Data exploration is the process of examining, visualizing, and understanding the dataset before applying machine learning models. It helps identify relationships between variables, detect outliers, and uncover trends or correlations that support the research objective. In this case study, the focus is on exploring how social media usage patterns relate to mental health indicators such as depression, worry, and sleep issues.

By performing **Exploratory Data Analysis (EDA)**, we aim to:

- Understand demographic distributions such as age and gender.
- Observe social media usage behavior (time spent, purpose, and engagement).
- Explore the relationship between social media use and depression levels.
- Visualize psychological factors and their correlations.

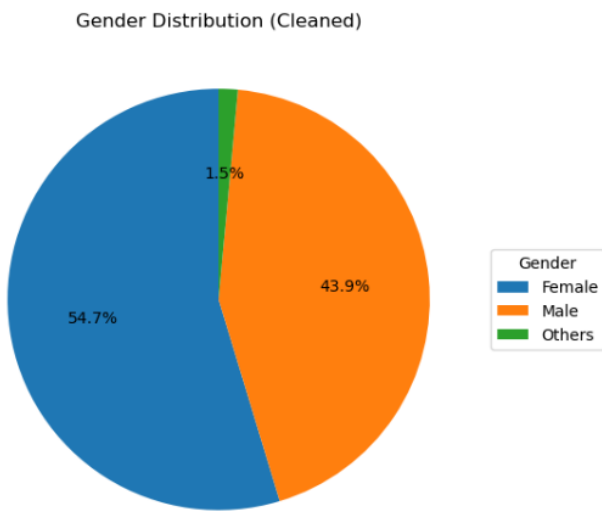
Plot Age distribution

```
# Age Distribution
plt.figure(figsize=(10, 6))
sns.histplot(df['age'], bins=20, kde=True)
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```



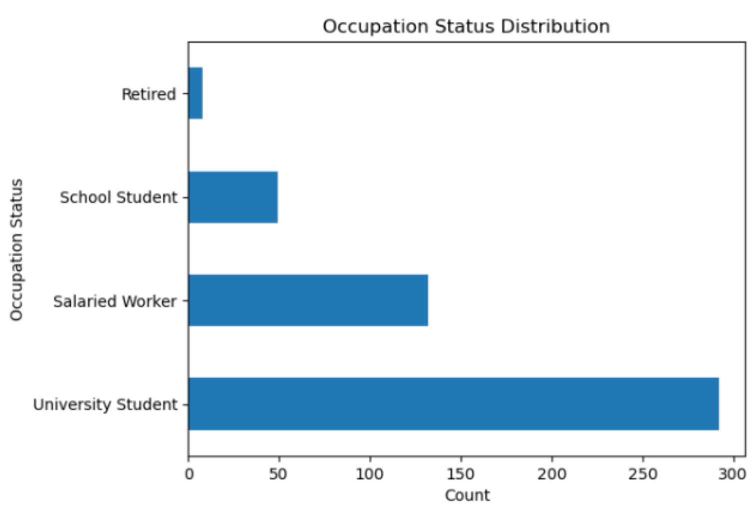
Clean Gender Values and plot pie(Male, Female, & Others)

```
#Gender Distribution
# Keep only Male, Female, and group others
df['gender'] = df['gender'].apply(lambda x: x if x in ['Male', 'Female'] else 'Others')
# Count values
gender_counts = df['gender'].value_counts()
# Plot pie chart
plt.figure(figsize=(8,6))
plt.pie(gender_counts, labels=None, autopct='%1.1f%%', startangle=90)
plt.title('Gender Distribution (Cleaned)')

# Add legend on the right
plt.legend(gender_counts.index, title='Gender', bbox_to_anchor=(1, 0.5), loc='center left')
plt.show()
```



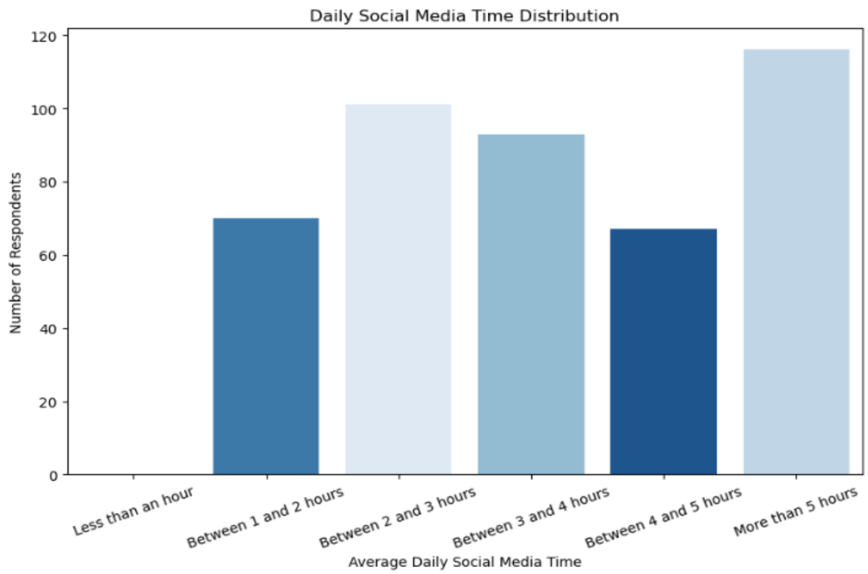
Occupation Status

```
# Occupation Status
occupation_counts = df['occupation_status'].value_counts()
occupation_counts.plot(kind='barh')
plt.title('Occupation Status Distribution')
plt.ylabel('Occupation Status')
plt.xlabel('Count')
plt.show()
```



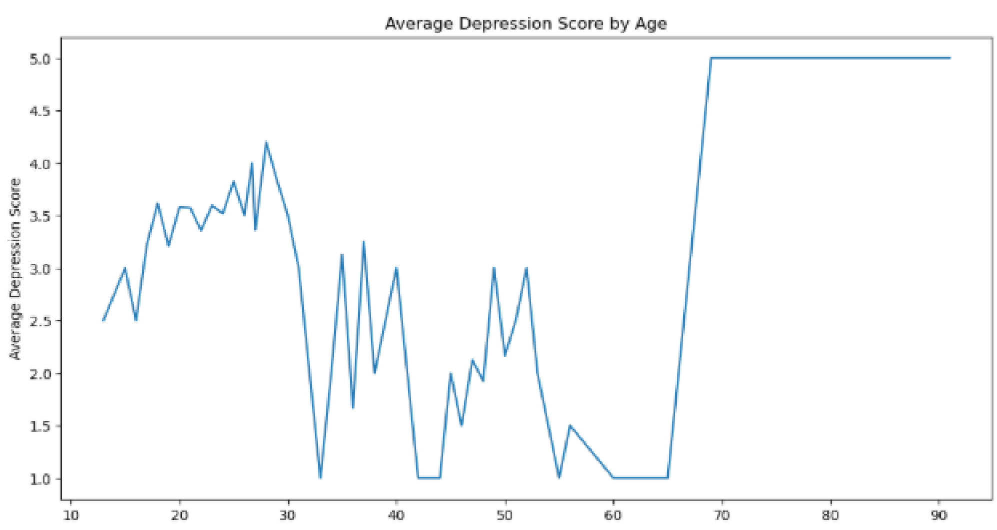
Daily social media time histogram

```
order = [
    'Less than an hour',
    'Between 1 and 2 hours',
    'Between 2 and 3 hours',
    'Between 3 and 4 hours',
    'Between 4 and 5 hours',
    'More than 5 hours'
]

plt.figure(figsize=(10, 6))
sns.countplot(x='daily_social_media_time', data=df, order=order, hue='daily_social_media_time', legend=False, palette="Blues")
plt.title('Daily Social Media Time Distribution')
plt.xlabel('Average Daily Social Media Time')
plt.ylabel('Number of Respondents')
plt.xticks(rotation=20)
plt.show()
```



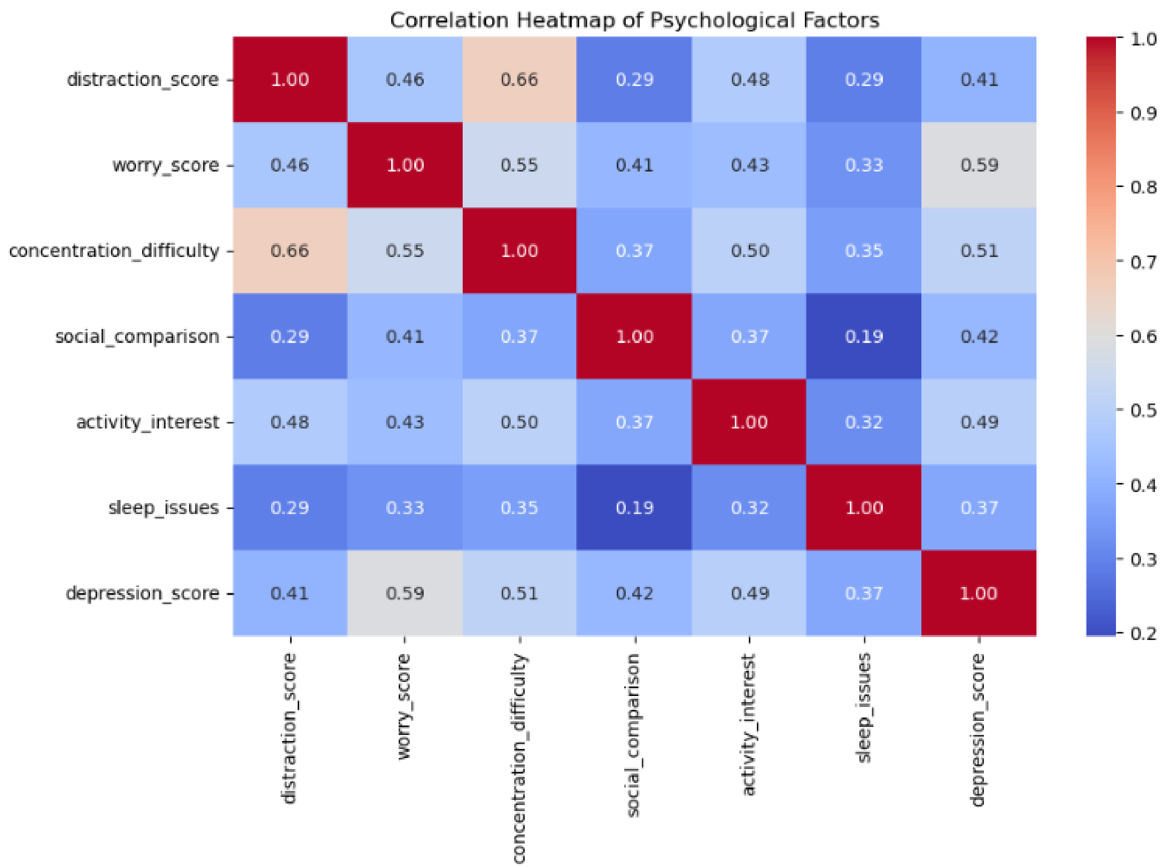
Average depression by age

```
# Relationship between depression and age
age_depression = df.groupby('age')['depression_score'].mean().reset_index()
plt.figure(figsize=(12, 6))
sns.lineplot(data=age_depression, x='age', y='depression_score')
plt.title('Average Depression Score by Age')
plt.xlabel('Age')
plt.ylabel('Average Depression Score')
plt.show()
```



Correlation heatmap of psychological factors

```
factors = ['distraction_score', 'worry_score', 'concentration_difficulty',
           'social_comparison', 'activity_interest', 'sleep_issues', 'depression_score']

plt.figure(figsize=(10,6))
sns.heatmap(df[factors].corr(), annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Correlation Heatmap of Psychological Factors")
plt.show()
```



Step - 5: Data Modeling

Data modeling is a crucial step in the machine learning process, where algorithms are trained to identify patterns and make predictions based on prepared data. In this project, the objective is to predict whether an individual is likely to experience a high level of depression based on social media usage patterns and psychological factors.

Selected model: **Logistic Regression model**

Logistic Regression is one of the most effective and interpretable algorithms for binary classification problems. It is used when the dependent variable has two outcomes (e.g., “Low Depression” or “High Depression”). It is simple, efficient, and provides interpretable results by showing how each feature contributes to the probability of depression.

Reasons for using Logistic Regression:

- It works well for small to medium-sized datasets.
- It gives clear probabilities for predictions.
- It is interpretable and easy to explain.
- It provides a strong baseline for classification.

```
# Create binary target: High depression (>=3) vs Low depression (<3)
df['depression_label'] = (df['depression_score'] >= 3).astype(int)
```

This line creates a new column called "depression_label", which converts the continuous depression_score into a binary classification label.

- If a person's depression_score is 3 or higher, they are considered to have High Depression and assigned a value of 1.
- If the score is below 3, they are classified as having Low Depression and assigned a value of 0.
- .astype(int) converts the Boolean values (True or False) into integers (1 or 0).

```
# Select numeric features for now
features = ['age', 'daily_social_media_time', 'distraction_score',
            'worry_score', 'concentration_difficulty',
            'social_comparison', 'sleep_issues']

x = df[features]
y = df['depression_label']
```

```
# Splitting the data into training and test datasets
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
```

```
print(x.shape, x_train.shape, x_test.shape)
```

```
(481, 7) (384, 7) (97, 7)
```

After selecting the important features for prediction, the next step is to split the dataset into two parts — one for training the model and another for testing its performance.

- train_test_split() divides the dataset so that 80% of the data (x_train, y_train) is used for training, and 20% (x_test, y_test) is used for testing.
- The parameter random_state=42 ensures reproducibility, meaning the same split occurs every time the code is run.
- x.shape, x_train.shape, and x_test.shape display the number of rows and columns in the total dataset, training set, and testing set respectively.

Once the data is split, we move to the model training phase, where the algorithm learns from the training data.

```python
model = LogisticRegression(max_iter=1000)
model.fit(x_train, y_train)
```

```
▾ LogisticRegression
LogisticRegression(max_iter=1000)
```

- LogisticRegression() initializes the Logistic Regression model, which is suitable for binary classification (in this case, predicting *High* or *Low Depression*).
- The parameter max_iter=1000 ensures the model has enough iterations to converge and find the best coefficients for prediction.
- model.fit(x_train, y_train) trains the model using the training features (x_train) and their corresponding labels (y_train).

After training the model, it's essential to evaluate its performance to understand how well it has learned from the data.

Accuracy Score

```python
# Accuracy on training data
x_train_pre = model.predict(x_train)
training_data_accuracy = accuracy_score(x_train_pre, y_train)
```

```python
print('Accuracy on Training data: ', training_data_accuracy)
```

```
Accuracy on Training data:  0.7916666666666666
```

```python
# Accuracy on test data
x_test_pre = model.predict(x_test)
test_data_accuracy = accuracy_score(x_test_pre, y_test)
```

```python
print('Accuracy on Test data: ', test_data_accuracy)
```

```
Accuracy on Test data:  0.8350515463917526
```

On Training data:

- The model predicts the depression labels for the training data using model.predict(x_train).
- These predictions are compared to the actual training labels (y_train) using accuracy_score().
- The result (training_data_accuracy) shows how accurately the model fits the data it was trained on.

On Testing data:

- The model then makes predictions on unseen test data (x_test), which was not used during training.
- Comparing these predictions with the true labels (y_test) helps measure the model's generalization ability.
- The test accuracy tells us how well the model performs on new, unseen data — which is crucial for real-world predictions.

Step 6: Presentation and Automation

After training and evaluating the machine learning model, the final step involves presenting the results and automating the prediction process.

1. Presentation of model Results — displaying the model's performance metrics (Accuracy, F1-score, ROC AUC, etc.) and visual results such as confusion matrices and correlation plots.
2. Automation of Prediction — allowing new input data to be tested automatically through the trained model, making the system user-friendly and scalable for real-time use.

By automating the prediction pipeline, the model can assist in identifying individuals at higher risk of depression based on behavioral patterns without requiring manual retraining or complex analysis each time.

After model training, the performance is evaluated using different metrics and visualizations to assess its effectiveness.

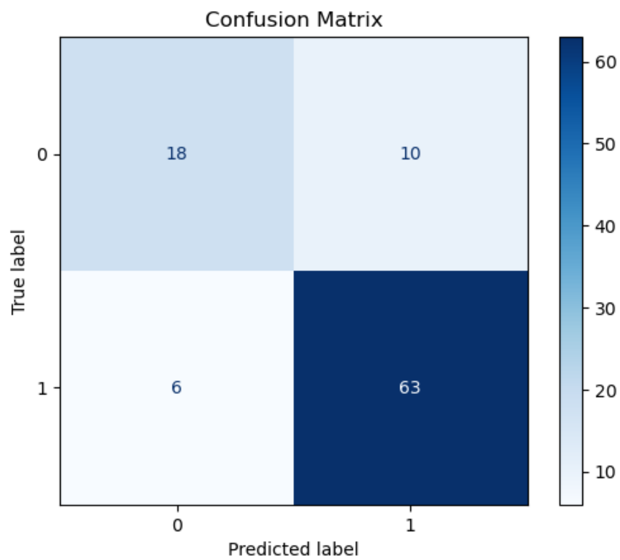
Presentation of model results:

```
# Predictions
y_pred = model.predict(x_test)

print("Classification Report:\n", classification_report(y_test, y_pred))
print("----------------------------------------------------------------------------------------------------")
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("----------------------------------------------------------------------------------------------------")
print("ROC AUC Score:", roc_auc_score(y_test, y_pred))
```

```
Classification Report:
               precision    recall  f1-score   support

           0       0.75      0.64      0.69        28
           1       0.86      0.91      0.89        69

    accuracy                           0.84        97
   macro avg       0.81      0.78      0.79        97
weighted avg       0.83      0.84      0.83        97

----------------------------------------------------------------------------------------------------
Confusion Matrix:
 [[18 10]
 [ 6 63]]
----------------------------------------------------------------------------------------------------
ROC AUC Score: 0.7779503105590062
```

Confusion Matrix

```
cm = confusion_matrix(y_test, x_test_pre)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot(cmap='Blues')
plt.title('Confusion Matrix')
plt.show()
```



Building an Automated Prediction System:

Building a predictive system

```
input_age = 22
input_daily_social_media_time = 3
input_distraction_score = 4
input_worry_score = 3
input_concentration_difficulty = 2
input_social_comparison = 3
input_sleep_issues = 4
input_data = (input_age, input_daily_social_media_time, input_distraction_score, input_worry_score,
              input_concentration_difficulty, input_social_comparison, input_sleep_issues)

# Change the iput data to a numpy array
input_arr = np.asarray(input_data)

# Reshape the numpy array as we are predicting only one instance
input_reshape = input_arr.reshape(1, -1)

prediction = model.predict(input_reshape)

if prediction[0] == 1:
    print("Prediction Result: Person is likely to have HIGH depression (score ≥ 3).")
    print("This indicates that the individual may be experiencing elevated mental stress or emotional strain.")
    print("Recommendation: It is advised to reduce excessive social media usage, maintain a balanced routine, and seek support or counseling if needed.")
else:
    print("Prediction Result: Person is likely to have LOW depression (score < 3).")
    print("The individual appears to have a stable mental state based on the given inputs.")
    print("Recommendation: Continue healthy online habits and maintain a positive balance between social media and real-life activities.")
```

Prediction:

```
Prediction Result: Person is likely to have HIGH depression (score ≥ 3).
This indicates that the individual may be experiencing elevated mental stress or emotional strain.
Recommendation: It is advised to reduce excessive social media usage, maintain a balanced routine, and seek support or counseling if needed.
```

Conclusion

By using **Logistic Regression**, the model achieved strong accuracy and interpretability in identifying individuals who are likely to experience higher levels of depression based on behavioural and psychological indicators.

Through clear visualization of results such as the confusion matrix, accuracy scores, and classification metrics the model's reliability was effectively demonstrated. The automation of the prediction process further enhances its usefulness, allowing real-time mental health risk assessment from user inputs such as age, daily social media time, and self-reported emotional scores.

This predictive system is designed not only to classify users as *low* or *high depression risk* but also to provide personalized recommendations that encourage healthier digital habits and early mental health awareness.