In [9]:
```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import numpy as np
import pandas as pd

df = pd.read_csv('depression_data.csv')

# Prepare the data
X = df.drop(['Name', 'History of Mental Illness'], axis=1)
y = df['History of Mental Illness']

# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta

# Define preprocessing steps
numeric_features = ['Age', 'Number of Children', 'Income']
categorical_features = ['Marital Status', 'Education Level', 'Smoking Status',
                        'Physical Activity Level', 'Employment Status',
                        'Alcohol Consumption', 'Dietary Habits', 'Sleep Patterns',
                        'History of Substance Abuse', 'Family History of Depressior
                        'Chronic Medical Conditions']

preprocessor = ColumnTransformer(
    transformers=[
        ('num', StandardScaler(), numeric_features),
        ('cat', OneHotEncoder(drop='first', sparse_output=False), categorical_featu
    ])

# Create pipelines
lr_pipeline = Pipeline([
    ('preprocessor', preprocessor),
    ('classifier', LogisticRegression(random_state=42, max_iter=1000))
])

rf_pipeline = Pipeline([
    ('preprocessor', preprocessor),
    ('classifier', RandomForestClassifier(random_state=42))
])

# Fit and evaluate models
models = [lr_pipeline, rf_pipeline]
model_names = ['Logistic Regression', 'Random Forest']

for name, model in zip(model_names, models):
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    print(f"\n{name} Results:")
    print(f"Accuracy: {accuracy_score(y_test, y_pred):.4f}")
    print("\nClassification Report:")
    print(classification_report(y_test, y_pred))
    print("\nConfusion Matrix:")
    print(confusion_matrix(y_test, y_pred))

# Feature importance for Random Forest
try:
    rf_feature_importance = rf_pipeline.named_steps['classifier'].feature_importanc
    feature_names = (numeric_features +
```

```
                     preprocessor.named_transformers_['cat']
                     .get_feature_names_out(categorical_features).tolist())

    importance_df = pd.DataFrame({'feature': feature_names, 'importance': rf_featur
    importance_df = importance_df.sort_values('importance', ascending=False)
    print("\nTop 10 Most Important Features:")
    print(importance_df.head(10))
except Exception as e:
    print(f"An error occurred while getting feature importances: {str(e)}")

# Print class distribution
print("\nClass Distribution:")
print(y.value_counts(normalize=True))
```

```
Logistic Regression Results:
Accuracy: 0.6945

Classification Report:
```

```
C:\Users\ibrah\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:146
9: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to
0.0 in labels with no predicted samples. Use `zero_division` parameter to control
this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\ibrah\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:146
9: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to
0.0 in labels with no predicted samples. Use `zero_division` parameter to control
this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\ibrah\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:146
9: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to
0.0 in labels with no predicted samples. Use `zero_division` parameter to control
this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

```
              precision    recall  f1-score   support

          No       0.69      1.00      0.82     57471
         Yes       0.00      0.00      0.00     25283

    accuracy                           0.69     82754
   macro avg       0.35      0.50      0.41     82754
weighted avg       0.48      0.69      0.57     82754


Confusion Matrix:
[[57471     0]
 [25283     0]]

Random Forest Results:
Accuracy: 0.6577

Classification Report:
              precision    recall  f1-score   support

          No       0.70      0.88      0.78     57471
         Yes       0.36      0.15      0.21     25283

    accuracy                           0.66     82754
   macro avg       0.53      0.52      0.50     82754
weighted avg       0.60      0.66      0.61     82754


Confusion Matrix:
[[50686  6785]
 [21538  3745]]

Top 10 Most Important Features:
                             feature  importance
2                             Income    0.393166
0                                Age    0.273199
1                 Number of Children    0.062713
23    Chronic Medical Conditions_Yes    0.027249
21    History of Substance Abuse_Yes    0.026299
22  Family History of Depression_Yes    0.024691
16      Alcohol Consumption_Moderate    0.019747
15           Alcohol Consumption_Low    0.017990
12  Physical Activity Level_Moderate    0.016192
20               Sleep Patterns_Poor    0.015683

Class Distribution:
No     0.695904
Yes    0.304096
Name: History of Mental Illness, dtype: float64
```