SAINT LOUIS UNIVERSITY

INSTRUCTORS

Divya S. Subramaniam, PhD, MPH

Assistant Professor and Program Director

Department of Health and Clinical Outcomes Research

School of Medicine

Dipti P. Subramaniam, PhD, MPH

Associate Professor and Director of Curriculum Development and Strategic Outreach

School of Medicine

Department of Health and Clinical Outcomes Research

Capstone Project By: HDS-5960-01 - Group 02

Team Members

Raju Goud Burra

Hadassah Oli Chinnam

Joshitha Dandlamudi

Pravanith Reddy Kankanala

Deborah Shilpi Raghani

Predicting and Analyzing Mental Health Severity from Digital and Lifestyle Habits

Abstract

Problem Description:

With the increasing dependency on digital devices, screen time has become a key concern for mental well-being. Numerous studies link extended screen usage to sleep problems, anxiety, depression, and overall mental distress. Digital behaviors such as prolonged social media use, binge-watching, or multitasking across screens can affect cognitive and emotional regulation. Variables such as daily screen time, social media hours, gaming, and TV usage are known contributors to mental health changes. Demographic factors like age and gender, along with lifestyle behaviors like physical activity and mindfulness, may influence these associations.

Research Question:

Do screen time and lifestyle habits significantly influence the severity of mental health conditions?

Background:

With the increasing dependency on digital devices, screen time has become a key concern for mental well-being (Devi & Singh, 2023c). Excessive screen time has been strongly linked to depression, anxiety, and emotional distress among young people (Devi & Singh, 2023c; Lin et al., 2016). Adolescents are found to be more vulnerable with screen usage leading to loneliness and reduced well-being (Keles et al. (2020). Studies show that moderate to high screen time are associated with low psychological well-being (Nagata et al. (2024), Twenge and Campbell (2018). Demographic and lifestyle factors such as age, gender, physical activity have a significant impact on the outcomes (Nagata et al., 2024). Hence, understanding the interplay of screen time with the lifestyle habits is essential in developing targeted mental health interventions. This study aims to explore the combined effect of screen time and lifestyle habits on mental health severity and to build predictive models that identify individuals at risk.

Introduction

This capstone project explores the application of machine learning (ML) techniques to predict and analyze mental health severity based on an individual's digital and lifestyle habits. By leveraging supervised learning methods, this study identifies key predictive factors and evaluates the effectiveness of different algorithms in classifying mental health outcomes. Predictive models offer valuable tools for mental health screening and intervention design, with the goal of improving early detection of at-risk individuals. Mental health outcomes are the result of a complex interplay of demographic, behavioral, and psychological variables (Nagata et al., 2024). Understanding these patterns is crucial for healthcare providers and public health stakeholders aiming to design data-informed policies. The core objective of this project is to determine whether machine learning methods can effectively classify individuals into "normal" or "severe" mental health categories based on their daily habits.

This project uses a public dataset combining self-reported digital usage, lifestyle choices, and mental health scores. Unlike traditional statistical analyses that often focus on linear relationships, ML can detect more complex and non-linear patterns within the data. This approach has shown growing utility in health analytics, and this project applies it to the critical area of mental health screening.

Methodology:

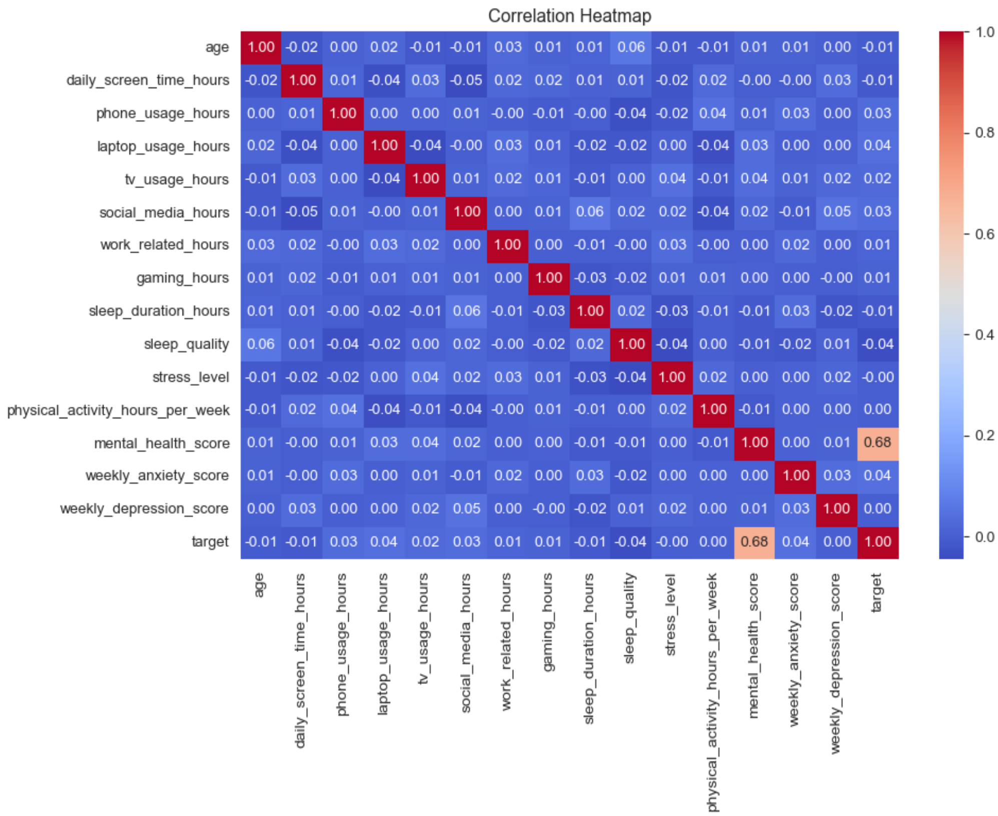
Data Source:
This study uses digital_diet_mental_health.csv dataset a publicly available dataset from Kaggle. The dataset includes about 2,000 participants aged 13 to 64, with complete responses on mental health and screen time habits. Key variables include daily screen time, social media use, gaming, TV, device usage, physical activity, age, gender, and mental health scores. Additional measures like weekly anxiety and depression scores and mindfulness minutes were also included.

Data Cleaning and Preparation:
To prepare the dataset for modeling, initial data cleaning was performed by removing irrelevant variables such as mood_rating, caffeine_intake_mg_per_day, and mindfulness_minutes_per_day, as they were not directly related to the primary research objectives. The continuous variable mental_health_score was then transformed into a binary target variable, mh_level, categorizing scores from 20 to 69 as "normal" and scores from 70 to 80 as "severe" to simplify classification. Categorical variables, including gender and location_type, were encoded using one-hot encoding to convert them into a numerical format suitable for machine learning algorithms. The data was split into training and testing sets in a 70:30 ratio using stratified sampling to maintain the proportion of classes across both subsets. To address class imbalance in the training data, the Synthetic Minority Over-sampling Technique (SMOTE) was applied, ensuring an equal distribution of the target classes. Finally, continuous features were standardized using StandardScaler to ensure that models are sensitive to feature scales.

Data Analysis:
Supervised machine learning models were developed to predict mental health risk levels based on screen time and behavioral variables. A total of six models were constructed and evaluated: Logistic Regression, Random Forest, XGBoost, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Naive Bayes. Each model underwent hyperparameter tuning using GridSearchCV with 3-fold cross-validation to identify the optimal parameter configurations. Model performance was assessed using a comprehensive set of evaluation metrics, including accuracy, precision, recall, F1-score, and AUC-ROC. Given the public health relevance of identifying individuals at high risk, particular emphasis was placed on maximizing recall for the "severe" mental health class to ensure that as many true positive cases as possible were correctly identified. The outcomes of the models were compared to determine which algorithm offered the best balance between sensitivity and overall predictive power.

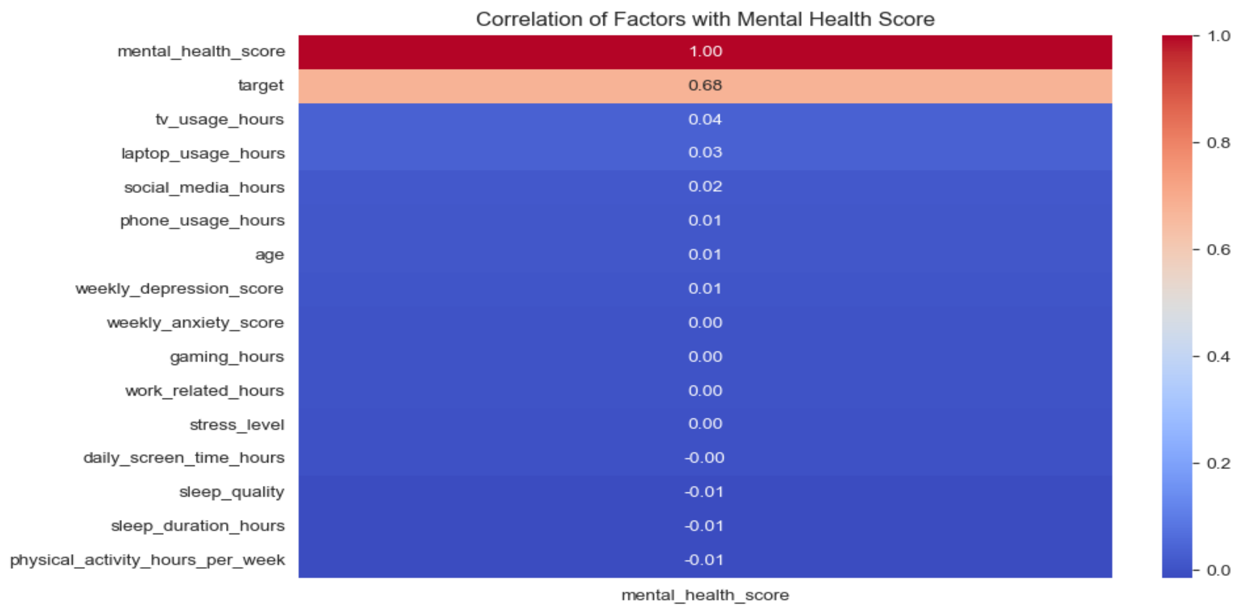
Data Visualizations:

Correlation of Factors with Daily Screen Time:



This analysis examined the relationship between daily screen time and various behavioral, emotional, and demographic factors. The results revealed that individuals with lower sleep duration, reduced physical activity, and poorer mood tend to spend more time on screens. Lower academic performance, diminished self-esteem, and fewer in-person social interactions also correlated negatively with screen time, suggesting possible compensatory or avoidance behaviors. On the other hand, higher screen time was moderately associated with increased age, elevated anxiety levels, and frequent screen use before bed. Caffeine intake also showed a slight positive correlation. These findings suggest that screen time is closely linked to overall lifestyle and mental well-being, highlighting the importance of addressing these interconnected factors in digital health interventions.

Top 10 Factors Affecting Screen Time:



The top factors influencing daily screen time included both negative and positive correlations. Less sleep and physical activity were the strongest negative predictors, followed by lower mood, academic performance, and self-esteem. Individuals with limited face-to-face social interaction also tended to spend more time on screens. On the positive side, age, anxiety, bedtime screen use, and caffeine intake were associated with higher screen time. These patterns suggest that screen overuse is often rooted in behavioral and emotional habits, with digital engagement potentially serving as a coping mechanism or substitute for healthier routines.

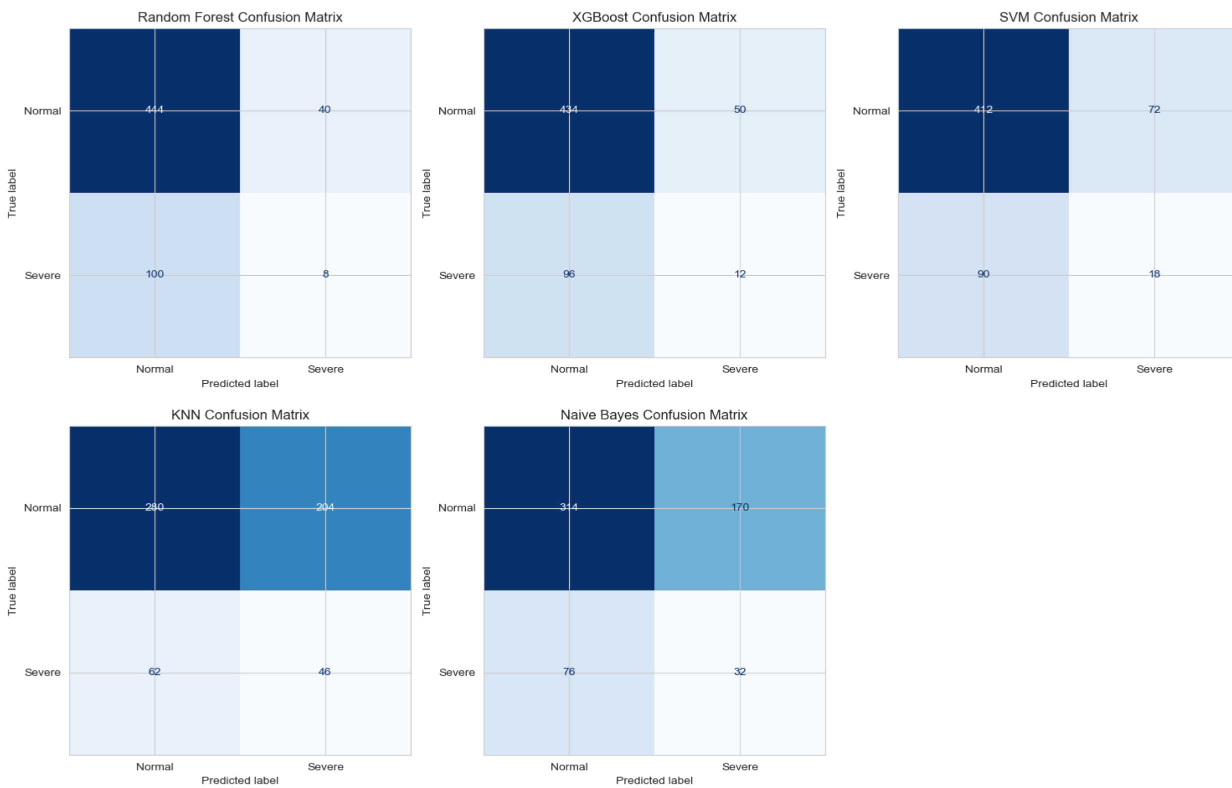
Results:

This study explored how well different machine learning models could predict mental health severity based on lifestyle and behavioral data. We tested five different classifiers: Random Forest, XGBoost, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Naive Bayes. While all models showed strong overall accuracy, the real challenge was detecting individuals with more severe mental health issues — the smaller group in our dataset. Because of this imbalance, we placed more importance on the model's ability to correctly identify severe cases rather than just getting the overall prediction right.

MODEL	Accuracy	AUC-ROC	Recall (Severe)	Precision (Severe)
KNN	0.55	0.50	0.43	0.18
Naive Bayes	0.58	0.46	0.30	0.16
SVM	0.73	0.53	0.17	0.20
XG Boost	0.75	0.45	0.11	0.19
Random Forest	0.76	0.47	0.07	0.17

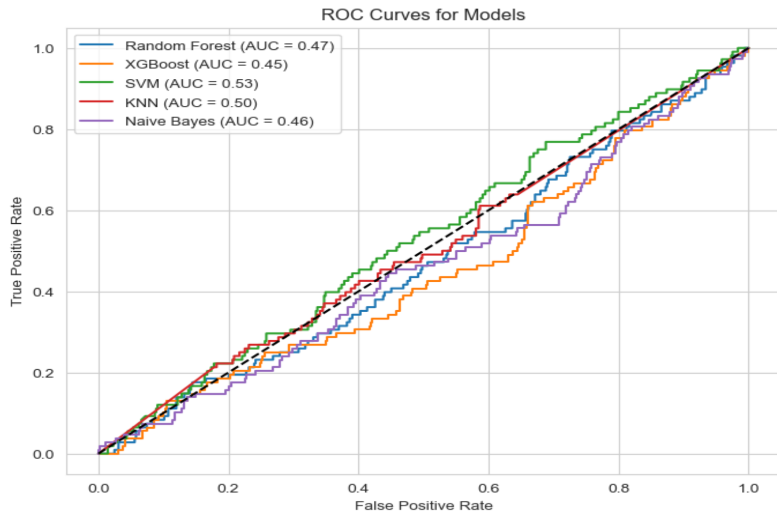
Model Summary:
Among the models, Random Forest achieved the highest overall accuracy at 76.3%, closely followed by XGBoost at 75.3%. However, this high accuracy did not translate into effective detection of the severe class, as both models exhibited very low recall for these cases. SVM, although slightly lower in accuracy (72.6%), demonstrated the best AUC-ROC score at 0.526, suggesting marginally better discrimination between the two classes across different thresholds. Interestingly, the KNN model, which had the lowest accuracy (55%), yielded the highest recall for the severe group at 43%, highlighting its potential utility in contexts where sensitivity is prioritized over precision. Naive Bayes offered more balanced results, with moderate recall and a lower false negative rate than the tree-based models, although it remained suboptimal overall.

Confusion Matrices:



When we looked at the confusion matrices , charts that break down how each prediction compares to the actual results ,we saw that most models performed well at catching the “normal” class but missed the mark on severe cases. For example, Random Forest and XGBoost barely identified any severe cases correctly. SVM did a little better, and KNN had the best performance in this regard, though it flagged a lot of false positives along the way. Naive Bayes was somewhere in the middle not great, but not the worst either. Overall, these matrices helped confirm that the models had trouble balancing accuracy with sensitivity to the more critical, underrepresented class.

ROC curves for Models:



The ROC curves helped visualize how well each model distinguished between normal and severe cases. Most curves closely followed the diagonal—suggesting weak performance—though SVM showed a slight edge, indicating a more balanced classification. We also examined the confidence of the Random Forest model using predicted probability plots. The results showed a clear bias toward predicting "normal" cases, with low probability scores assigned to severe ones. This imbalance highlights the model's limited sensitivity, likely due to the class imbalance in the dataset.

In addition, we explored how daily screen time correlates with various lifestyle and emotional factors. The analysis revealed that individuals who slept less, exercised less, or had lower mood, self-esteem, and academic performance tended to spend more time on screens. Conversely, screen time was positively linked with age, anxiety, bedtime screen use, and caffeine intake—suggesting that emotional or behavioral distress may drive excessive digital engagement.

Random Forest Hyperparameter Tuning:

To improve our Random Forest model, we also ran a randomized hyperparameter tuning process. Instead of testing every single possible combination, we let the algorithm try 20 randomly chosen sets of parameters. The goal was to find the best balance of settings that would improve model performance — especially for catching severe cases. The best setup included 444 trees, a maximum depth of 26, a split requirement of 5 samples, and at least 1 sample per leaf node. While these parameters slightly improved the model's overall performance, they still didn't significantly help with recall for severe cases, suggesting that tuning alone isn't enough when data is imbalanced.

In summary, each model brought its own strengths and limitations. Random Forest and XGBoost were great at overall classification but poor at detecting the most vulnerable individuals. SVM offered a good balance with decent AUC and recall. KNN, despite being less accurate overall, was the only model that truly picked up on severe cases. This trade-off between accuracy and sensitivity is critical especially in healthcare, where missing a high-risk individual can have real-world consequences. Our findings also highlight how lifestyle habits, especially screen time, are deeply intertwined with mental health and may serve as early indicators for potential issues.

Using Generative AI:

To help overcome the issue of class imbalance in our dataset, we used a generative AI technique called CTGAN to create 1,000 new synthetic samples based on key behavioral and emotional features. These new data points were combined with the original dataset to better represent individuals at high risk for mental health concerns. After splitting and scaling the data, we trained both Logistic Regression and Random Forest models on this expanded dataset. The results were promising especially for Logistic Regression, which saw a notable improvement in its ability to identify high-risk individuals. Its recall jumped from around 17% to nearly 49%, showing that the model was much more sensitive after being trained on the more balanced data. While Random Forest maintained its strong overall accuracy, it still struggled to pick up on high-risk cases, likely due to its conservative nature. Still, the experiment demonstrated that synthetic data generation can be a valuable tool for boosting model performance where identifying minority classes like individuals with severe mental health concerns is critical.

Conclusion:

This study set out to understand whether machine learning could help identify individuals at high risk of severe mental health issues based on their digital behavior and lifestyle habits. We tested multiple models including Random Forest, XGBoost, SVM, KNN, and Logistic Regression to see how well they could predict mental health severity. While models like Random Forest and XGBoost performed strongly in terms of overall accuracy, they had difficulty detecting the high-risk group, which was a much smaller portion of the dataset. This imbalance made it clear that accuracy alone is not enough when dealing with real-world health concerns where early identification is critical.

Among the models, KNN stood out for its ability to detect more severe cases, though it came with more false positives. SVM offered a more balanced performance, and Logistic Regression improved significantly once we addressed the data imbalance. Using CTGAN, a generative AI technique, we created synthetic samples that helped improve recall and fairness. Notably, Random Forest's accuracy increased to over 82%, and Logistic Regression became far more sensitive to severe cases — a major step forward. These results show that combining machine learning with thoughtful data balancing can significantly improve how we identify individuals who may be struggling, especially in underrepresented or overlooked groups.

Recommendations:

Looking ahead, we recommend shifting the focus from just model accuracy to a deeper emphasis on recall and sensitivity — especially when predicting high-risk individuals. In mental health applications, it's far more important to catch someone who may need support, even at the cost of a few false alarms. One clear takeaway from our study is that synthetic data generation, such as CTGAN, can help rebalance datasets and drastically improve a model's ability to identify at-risk individuals. These tools should be part of any workflow dealing with imbalanced health data.

It’s also worth exploring ensemble or hybrid modeling approaches, since no single model in our study was perfect across all metrics. Combining strengths from different algorithms may lead to more stable and robust predictions. Additionally, bringing in richer behavioral data — like screen usage patterns, mobile phone activity, sleep tracking, and even social interaction frequency — could give models more context and improve their ability to pick up subtle warning signs.

Finally, models should be tested and validated in real-world environments, such as universities, clinics, or digital mental health platforms. This helps ensure they’re not just effective on paper but also practical, ethical, and inclusive when deployed at scale. By focusing on fairness, early detection, and contextual understanding, we can build smarter systems that truly support mental well-being.

References

1.Devi, K. A., & Singh, S. K. (2023c). The hazards of excessive screen time: Impacts on physical health, mental health, and overall well-being. *Journal of Education and Health Promotion*, *12*(1). https://doi.org/10.4103/jehp.jehp_447_23

2.Keles, B., McCrae, N., & Grealish, A. (2019). A systematic review: the influence of social media on depression, anxiety and psychological distress in adolescents. *International Journal of Adolescence and Youth*, *25*(1), 79–93. https://doi.org/10.1080/02673843.2019.1590851

3.Lin, L. Y., Sidani, J. E., Shensa, A., Radovic, A., Miller, E., Colditz, J. B., Hoffman, B. L., Giles, L. M., & Primack, B. A. (2016). ASSOCIATION BETWEEN SOCIAL MEDIA USE AND DEPRESSION AMONG U.S. YOUNG ADULTS. *Depression and Anxiety*, *33*(4), 323–331. https://doi.org/10.1002/da.22466

4.Nagata, J. M., Al-Shoaibi, A. A., Leong, A. W., Zamora, G., Testa, A., Ganson, K. T., & Baker, F. C. (2024). Screen time and mental health: a prospective analysis of the Adolescent Brain Cognitive Development (ABCD) Study. *BMC Public Health*, *24*(1). https://doi.org/10.1186/s12889-024-20102-x

5.Orben, A., & Przybylski, A. K. (2019). The association between adolescent well-being and digital technology use. *Nature Human Behaviour*, *3*(2), 173–182. https://doi.org/10.1038/s41562-018-0506-1

6.Twenge, J. M., & Campbell, W. K. (2018). Associations between screen time and lower psychological well-being among children and adolescents: Evidence from a population-based study. *Preventive Medicine Reports*, *12*, 271–283. https://doi.org/10.1016/j.pmedr.2018.10.003

Appendix A: Jupyter Notebook

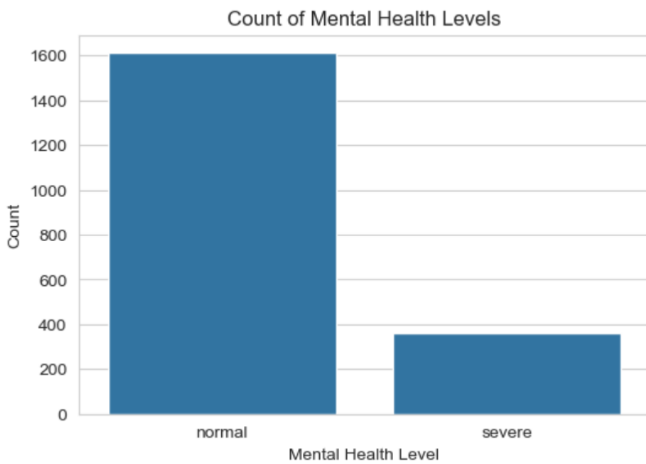
The full source code and outputs are provided in the accompanying Jupyter Notebook file.

Git hub link for the code https://github.com/Pravanith/Projects-

Libraries used: pandas, numpy, matplotlib, seaborn, scikit-learn, statsmodels, and xgboost.

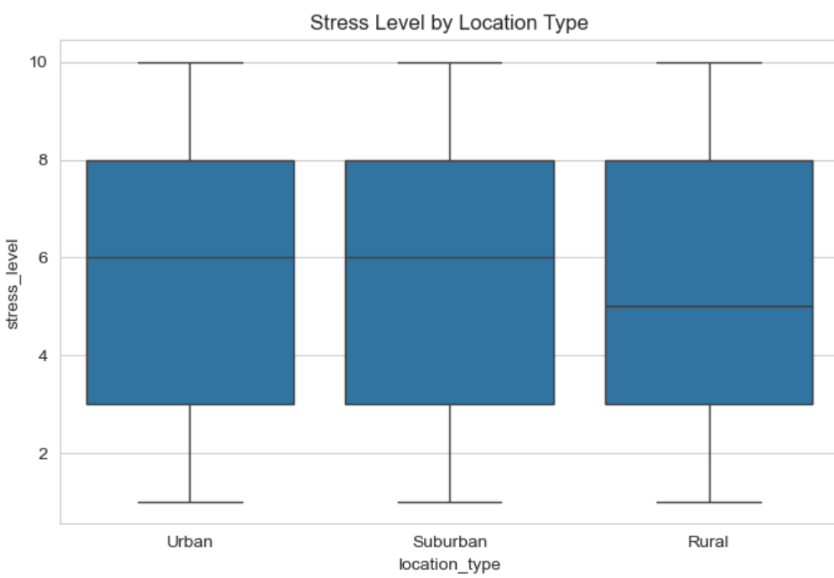
Appendix B:

```
plt.figure(figsize=(6,4))
sns.countplot(data=data, x="mh_level", order=["normal", "severe"])
plt.title("Count of Mental Health Levels")
plt.xlabel("Mental Health Level")
plt.ylabel("Count")
plt.show()
```

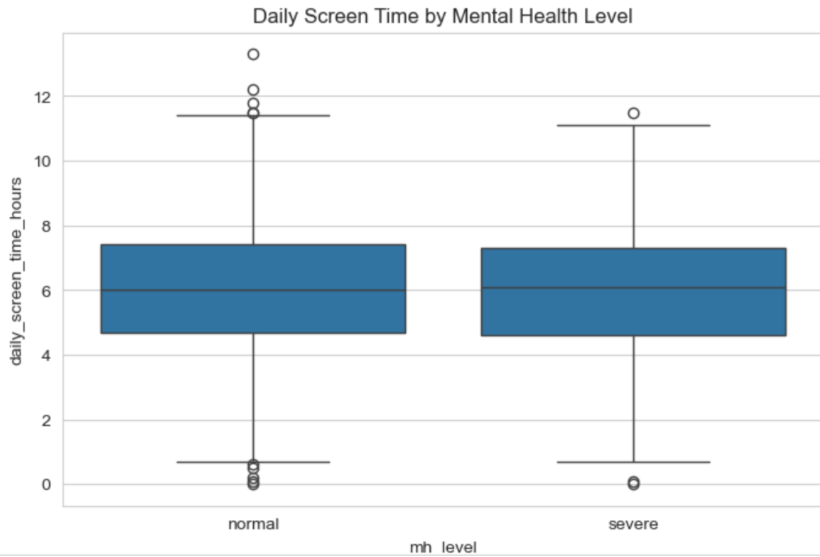


In [33]:

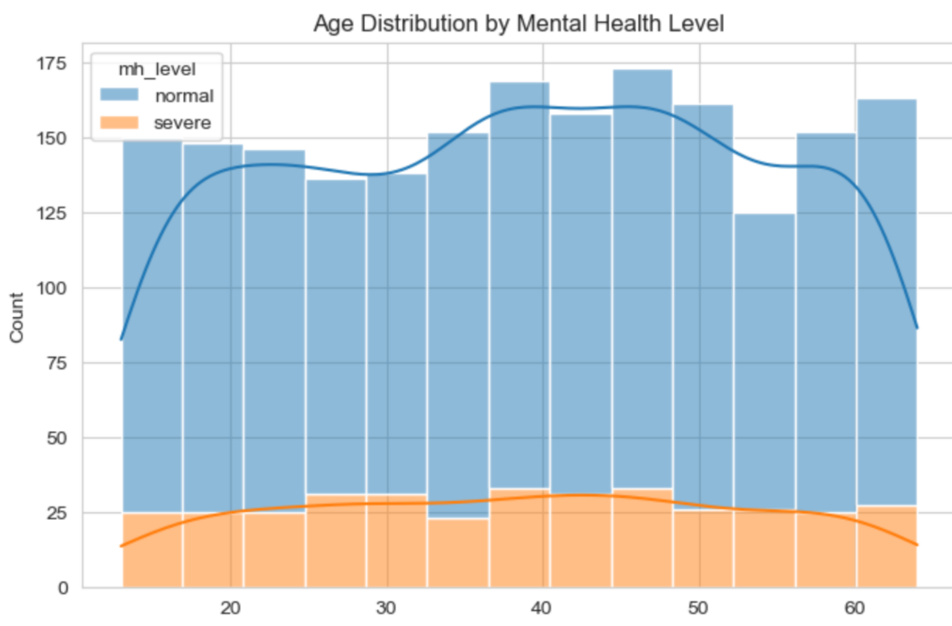
```
plt.figure(figsize=(8,5))
sns.boxplot(data=data, x="location_type", y="stress_level")
plt.title("Stress Level by Location Type")
plt.show()
```



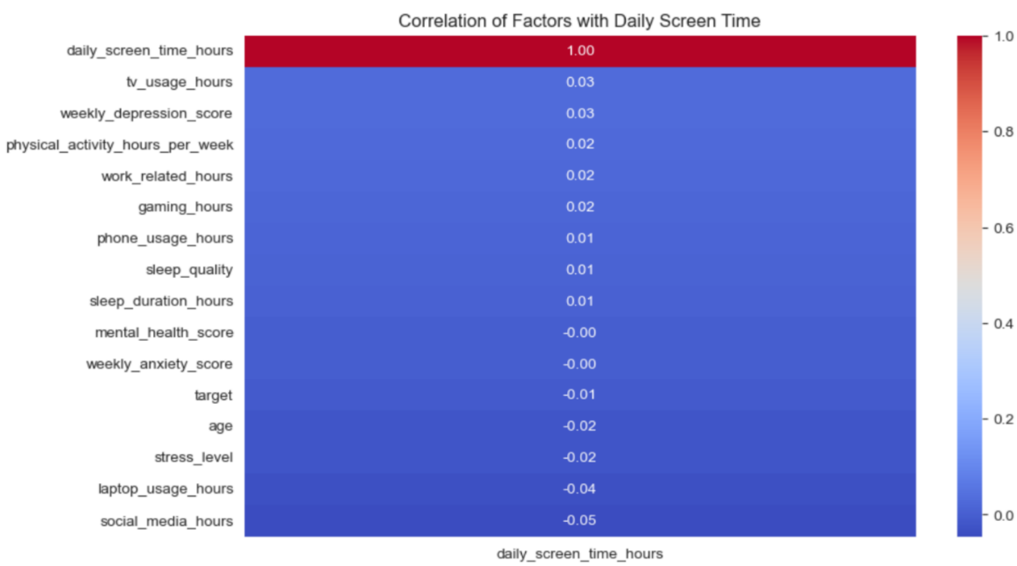
In [36]:
```
plt.figure(figsize=(8,5))
sns.boxplot(data=data, x='mh_level', y='daily_screen_time_hours')
plt.title("Daily Screen Time by Mental Health Level")
plt.show()
```



In [37]:
```
plt.figure(figsize=(8,5))
sns.histplot(data, x="age", hue="mh_level", kde=True, multiple="stack")
plt.title("Age Distribution by Mental Health Level")
plt.show()
```

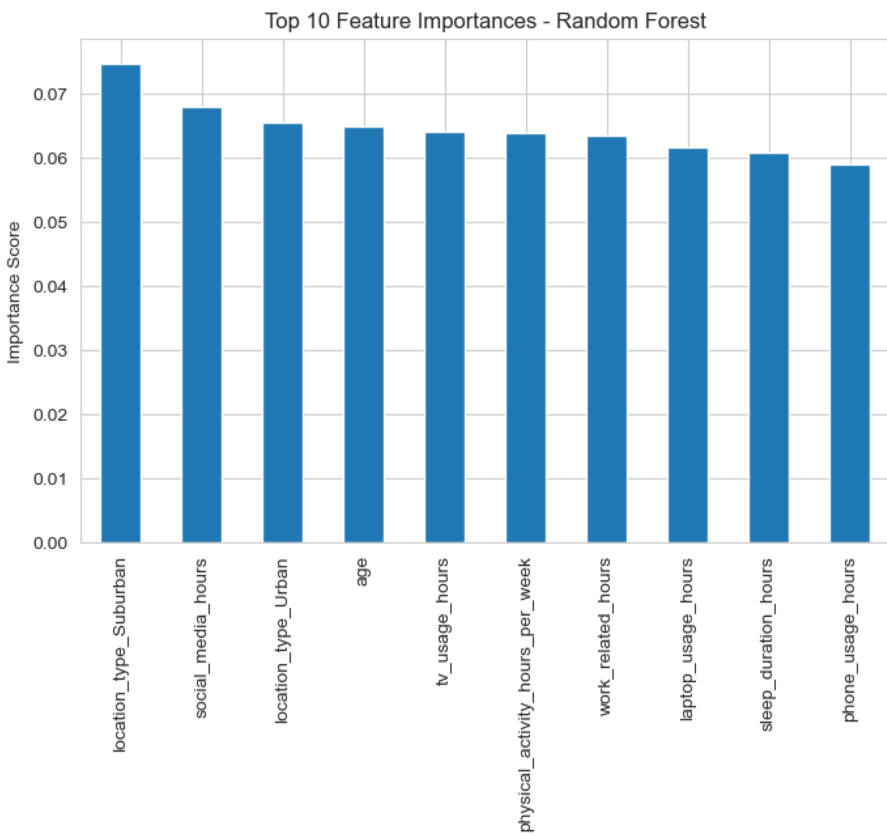


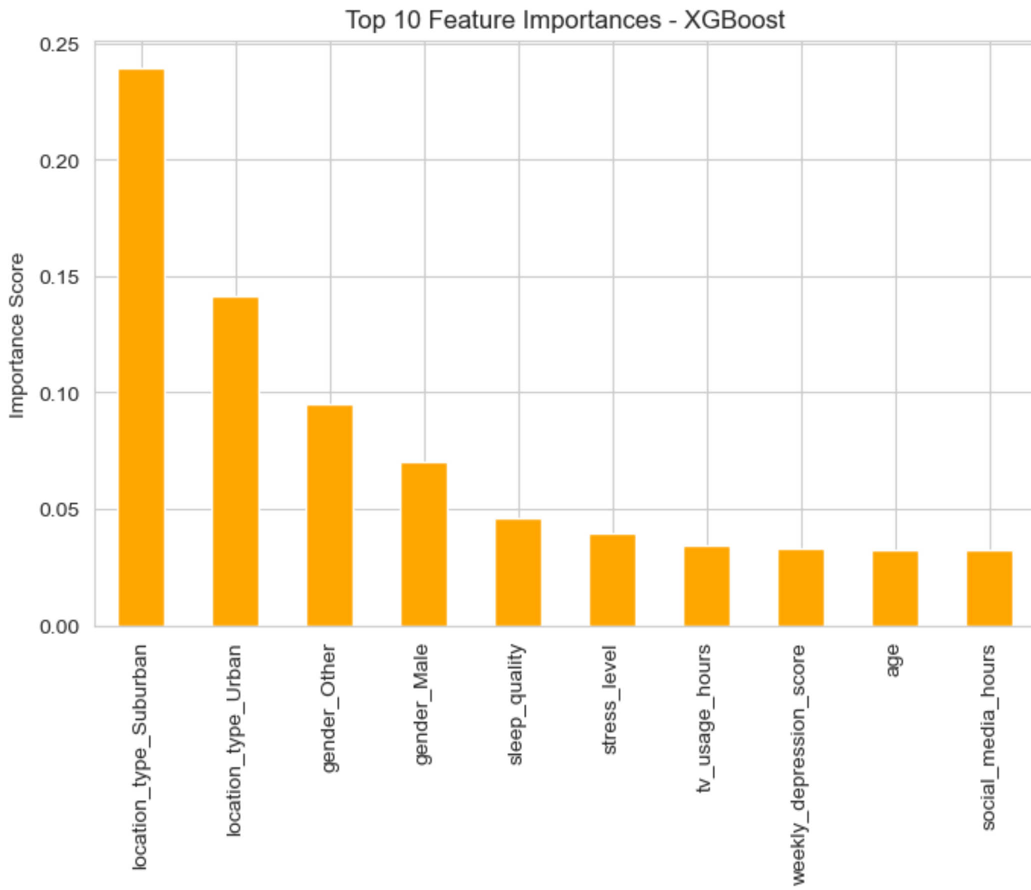
In [59]:
```python
# Correlation with screen time
plt.figure(figsize=(10,6))
sns.heatmap(
    data.corr(numeric_only=True)[['daily_screen_time_hours']].sort_values(by='daily_screen_time_hours', ascending=False),
    annot=True, cmap='coolwarm', fmt=".2f"
)
plt.title("Correlation of Factors with Daily Screen Time")
plt.show()

# Random Forest Regressor for feature importance
from sklearn.ensemble import RandomForestRegressor

features = data.drop(columns=[
    'user_id','mh_level','target','mental_health_score','daily_screen_time_hours'
])
features = pd.get_dummies(features, columns=['gender','location_type'], drop_first=True)
target_screen = data['daily_screen_time_hours']

rf_reg = RandomForestRegressor(random_state=42)
rf_reg.fit(features, target_screen)

importances = pd.Series(rf_reg.feature_importances_, index=features.columns).sort_values(ascending=False)

print("\nTop Factors Affecting Screen Time:\n", importances.head(10))

plt.figure(figsize=(8,5))
importances.head(10).plot(kind='bar')
plt.title("Top 10 Factors Affecting Screen Time (Feature Importance)")
plt.ylabel("Importance")
plt.show()
```



potentially more effective model.

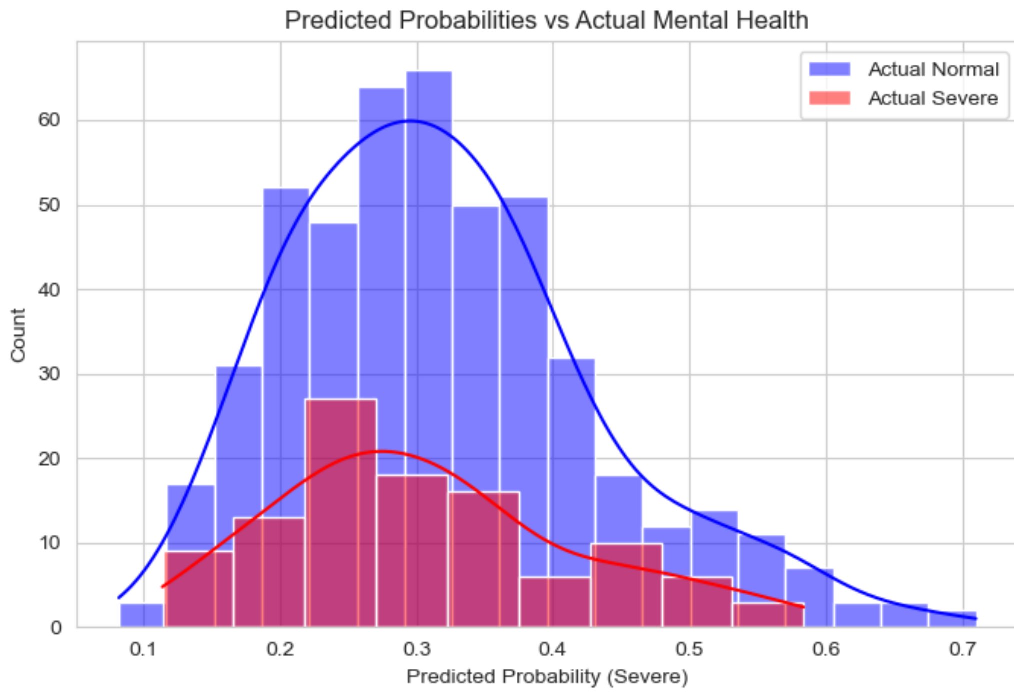
In [64]:
```
# Feature Importance for Random Forest
importances_rf = pd.Series(rf.best_estimator_.feature_importances_, index=X.columns).sort_values(ascending=False)
plt.figure(figsize=(8,5))
importances_rf.head(10).plot(kind='bar')
plt.title("Top 10 Feature Importances - Random Forest")
plt.ylabel("Importance Score")
plt.show()

# Feature Importance for XGBoost
importances_xgb = pd.Series(xgb.best_estimator_.feature_importances_, index=X.columns).sort_values(ascending=False)
plt.figure(figsize=(8,5))
importances_xgb.head(10).plot(kind='bar', color='orange')
plt.title("Top 10 Feature Importances - XGBoost")
plt.ylabel("Importance Score")
plt.show()
```





In [65]:

```
# Compare predicted probabilities with actual target distribution (e.g., Random Forest)
y_proba_rf = rf.best_estimator_.predict_proba(X_test)[:,1]
plt.figure(figsize=(8,5))
sns.histplot(y_proba_rf[y_test==0], color='blue', label='Actual Normal', kde=True)
sns.histplot(y_proba_rf[y_test==1], color='red', label='Actual Severe', kde=True)
plt.title("Predicted Probabilities vs Actual Mental Health")
plt.xlabel("Predicted Probability (Severe)")
plt.ylabel("Count")
plt.legend()
plt.show()
```



In [78]:

```
# ======================================
# 1. Install Required Libraries
# ======================================
!pip install imbalanced-learn xgboost ctgan

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import (
    accuracy_score, classification_report, roc_auc_score,
    confusion_matrix, ConfusionMatrixDisplay
)
from sklearn.ensemble import RandomForestClassifier, VotingClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from xgboost import XGBClassifier
from imblearn.over_sampling import SMOTE
from ctgan import CTGAN   # New API for ctgan 0.11.0

# ======================================
# 2. Load and Preprocess Original Data
# ======================================
data = pd.read_csv("digital_diet_mental_health.csv")
```

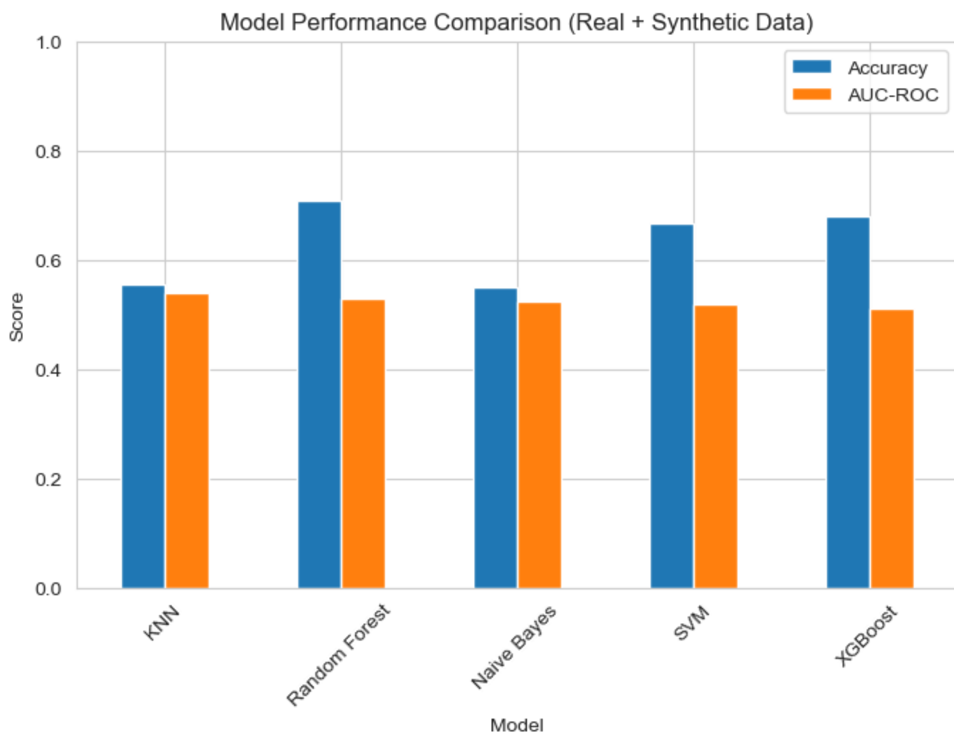
```python
# Add target column
# Clip mental_health_score to valid range
synthetic_data['mental_health_score'] = synthetic_data['mental_health_score'].clip(20, 80)

# Create mh_level safely
synthetic_data['mh_level'] = pd.cut(
    synthetic_data['mental_health_score'],
    bins=bins,
    labels=labels,
    right=True
)

# Map to target and drop NaN rows
synthetic_data['target'] = synthetic_data['mh_level'].map({"normal": 0, "severe": 1}).astype("Int8")
synthetic_data = synthetic_data.dropna(subset=['target'])


# ========================================
# 4. Combine Real + Synthetic Data
# ========================================
augmented_data = pd.concat([data.drop(columns=['user_id']), synthetic_data], ignore_index=True)
augmented_data = augmented_data.dropna(subset=['target'])

# ========================================
# 5. Feature Engineering
# ========================================
X = pd.get_dummies(
    augmented_data.drop(columns=['mental_health_score', 'mh_level', 'target']),
    columns=['gender', 'location_type'],
    drop_first=True
)
```

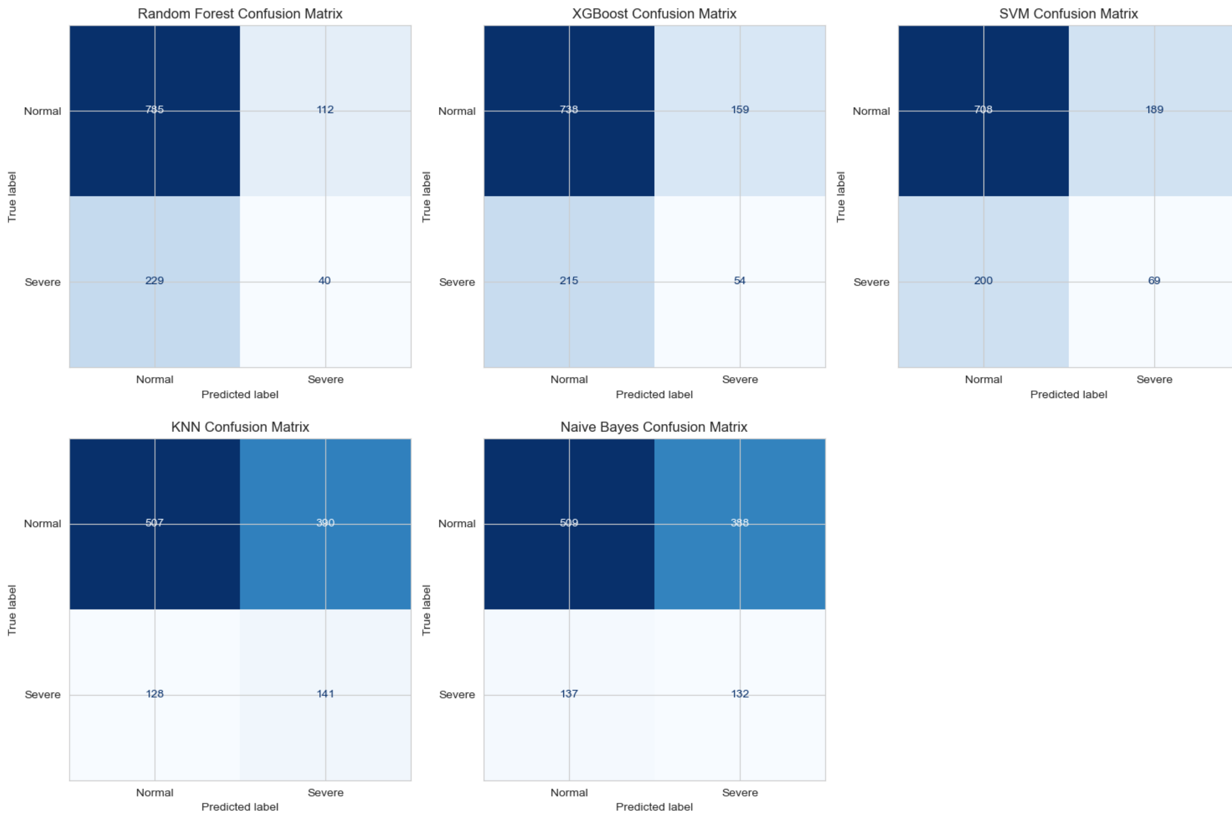
```python
# ========================================
# 6. Train-Test Split & Scaling
# ========================================
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, stratify=y, random_state=42)

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# ========================================
# 7. Handle Class Imbalance with SMOTE
# ========================================
smote = SMOTE(random_state=42)
X_train_res_scaled, y_train_res = smote.fit_resample(X_train_scaled, y_train)
X_train_res_unscaled, _ = smote.fit_resample(X_train, y_train)

# ========================================
# 8. Train Models and Evaluate
# ========================================
model_results = {}

def evaluate_model(model, X_test, y_test, name):
    y_pred = model.predict(X_test)
    y_proba = model.predict_proba(X_test)[:,1] if hasattr(model, "predict_proba") else None
    auc = roc_auc_score(y_test, y_proba) if y_proba is not None else "N/A"

    acc = accuracy_score(y_test, y_pred)
    report = classification_report(y_test, y_pred, target_names=['normal','severe'], output_dict=True)
```

```python
    # Save metrics
    model_results[name] = {
        "Accuracy": acc,
        "AUC-ROC": auc,
        "Precision (normal)": report['normal']['precision'],
        "Recall (normal)": report['normal']['recall'],
        "F1-score (normal)": report['normal']['f1-score'],
        "Precision (severe)": report['severe']['precision'],
        "Recall (severe)": report['severe']['recall'],
        "F1-score (severe)": report['severe']['f1-score']
    }

# Random Forest
rf_params = {'n_estimators':[100,200],'max_depth':[10,20,None],'min_samples_leaf':[2,4]}
rf = GridSearchCV(RandomForestClassifier(random_state=42), rf_params, cv=5, scoring='accuracy', n_jobs=-1)
rf.fit(X_train_res_unscaled, y_train_res)
evaluate_model(rf.best_estimator_, X_test, y_test, "Random Forest")

# XGBoost
xgb_params = {'n_estimators':[100,200],'max_depth':[3,5],'learning_rate':[0.05,0.1]}
xgb = GridSearchCV(XGBClassifier(eval_metric='logloss', random_state=42), xgb_params, cv=5, scoring='accuracy', n_jobs=-1)
xgb.fit(X_train_res_unscaled, y_train_res)
evaluate_model(xgb.best_estimator_, X_test, y_test, "XGBoost")

# SVM
svm_params = {'C':[0.1,1,10],'kernel':['linear','rbf']}
svm = GridSearchCV(SVC(probability=True, random_state=42), svm_params, cv=5, scoring='accuracy', n_jobs=-1)
svm.fit(X_train_res_scaled, y_train_res)
evaluate_model(svm.best_estimator_, X_test_scaled, y_test, "SVM")
```

```python
# KNN
knn_params = {'n_neighbors':[3,5,7],'weights':['uniform','distance']}
knn = GridSearchCV(KNeighborsClassifier(), knn_params, cv=5, scoring='accuracy', n_jobs=-1)
knn.fit(X_train_res_scaled, y_train_res)
evaluate_model(knn.best_estimator_, X_test_scaled, y_test, "KNN")

# Naive Bayes
nb = GaussianNB()
nb.fit(X_train_res_scaled, y_train_res)
evaluate_model(nb, X_test_scaled, y_test, "Naive Bayes")

# ========================================
# 9. Summary and Visualization
# ========================================
results_df = pd.DataFrame(model_results).T.reset_index().rename(columns={"index":"Model"})
results_df = results_df.sort_values(by="AUC-ROC", ascending=False)

print("\n=== Model Performance Summary (sorted by AUC-ROC) ===")
print(results_df)

# Bar Plot
results_df.plot(x='Model', y=['Accuracy','AUC-ROC'], kind='bar', figsize=(8,5))
plt.title("Model Performance Comparison (Real + Synthetic Data)")
plt.ylabel("Score")
plt.xticks(rotation=45)
plt.ylim(0, 1)
plt.show()
```

```python
# Confusion Matrices
models_list = [
    ("Random Forest", rf.best_estimator_, X_test),
    ("XGBoost", xgb.best_estimator_, X_test),
    ("SVM", svm.best_estimator_, X_test_scaled),
    ("KNN", knn.best_estimator_, X_test_scaled),
    ("Naive Bayes", nb, X_test_scaled)
]

plt.figure(figsize=(15,10))
for i, (name, model, X) in enumerate(models_list, 1):
    plt.subplot(2,3,i)
    y_pred = model.predict(X)
    cm = confusion_matrix(y_test, y_pred)
    disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=['Normal','Severe'])
    disp.plot(cmap='Blues', ax=plt.gca(), colorbar=False)
    plt.title(f"{name} Confusion Matrix")
plt.tight_layout()
plt.show()

# Save Summary
results_df.to_csv("model_performance_with_synthetic.csv", index=False)
print("Model performance summary saved as model_performance_with_synthetic.csv")
```




Random Forest Confusion Matrix
True label
Normal
Severe
785
112
229
40
Normal
Severe
Predicted label
XGBoost Confusion Matrix
True label
Normal
Severe
738
159
215
54
Normal
Severe
Predicted label
SVM Confusion Matrix
True label
Normal
Severe
708
189
200
69
Normal
Severe
Predicted label
KNN Confusion Matrix
True label
Normal
Severe
507
390
128
141
Normal
Severe
Predicted label
Naive Bayes Confusion Matrix
True label
Normal
Severe
509
388
137
132
Normal
Severe
Predicted label