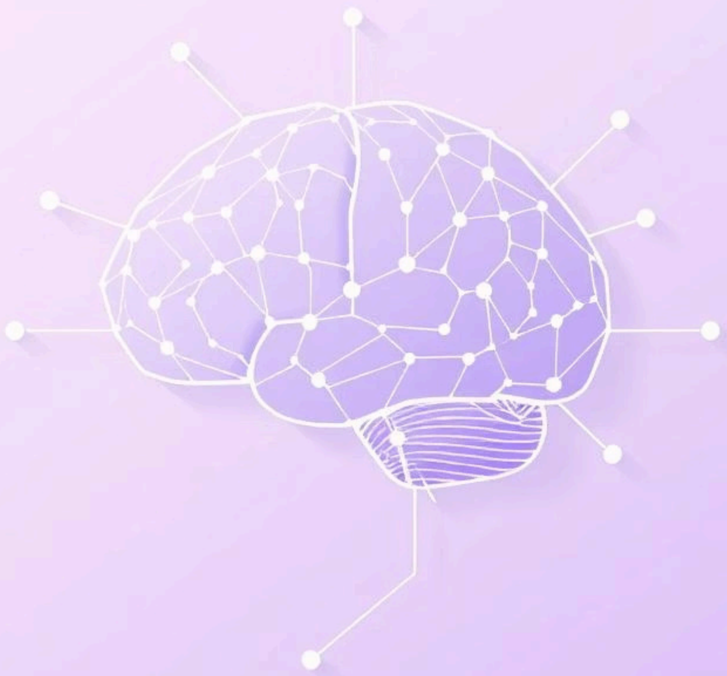
Dopamine Overload Prediction Using Machine Learning

Behavioral, Psychological & Digital Usage Risk Modeling



Presented by:

Zihan Manbit

Batch: CSE-58[A] | ID: 231-115-028

Department of CSE, Metropolitan University

Introduction

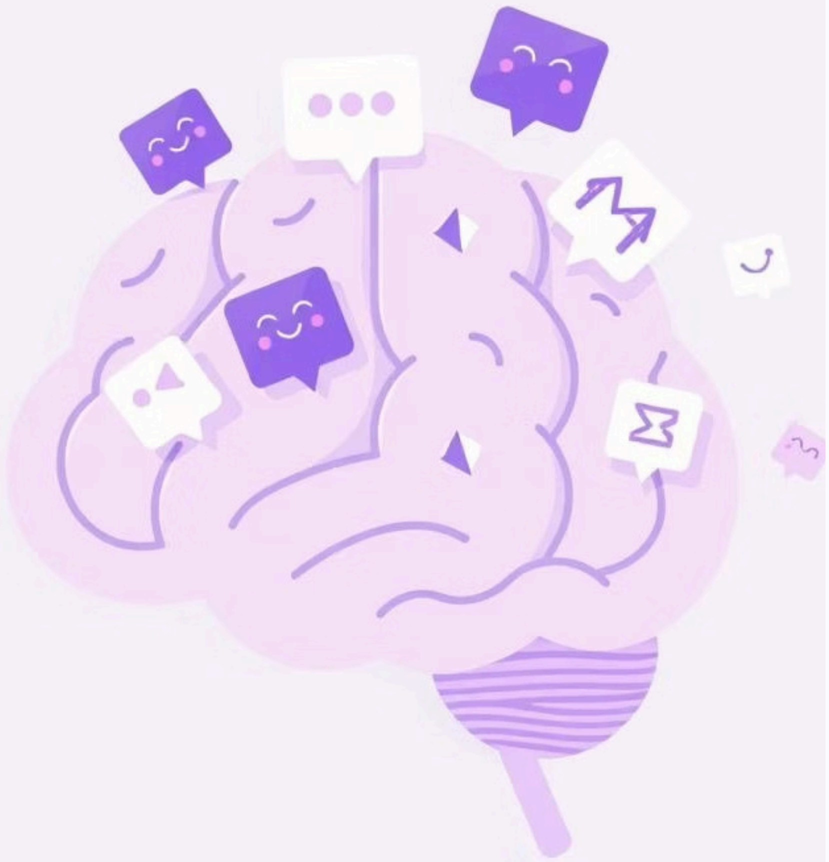
Dopamine is a neurotransmitter that plays a key role in the brain's reward, motivation and mood regulation. Excessive digital stimulation may cause overload, affecting mood, attention and productivity.

Motivation: Early detection of dopamine overload can help improve mental well-being and digital habits.

Importance: Understanding behavioral + psychological patterns enables personalized interventions and healthier digital usage.

Key steps:

- EDA
- Preprocessing & feature engineering
- Model Training
- Evaluation
- Insights & interpretability



Project Overview

Objectives:

- Predict high-risk users.
- Estimate digital dependence scores.
- Identify top behavioral predictors.

Workflow:

Dataset → EDA → Preprocessing → SMOTE → Model Training → Evaluation → SHAP

Libraries: Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn, XGBoost, SMOTE, SHAP.

Dataset Description

Dataset used:

Digital Lifestyle Benchmark dataset from HF/Kaggle

Details:

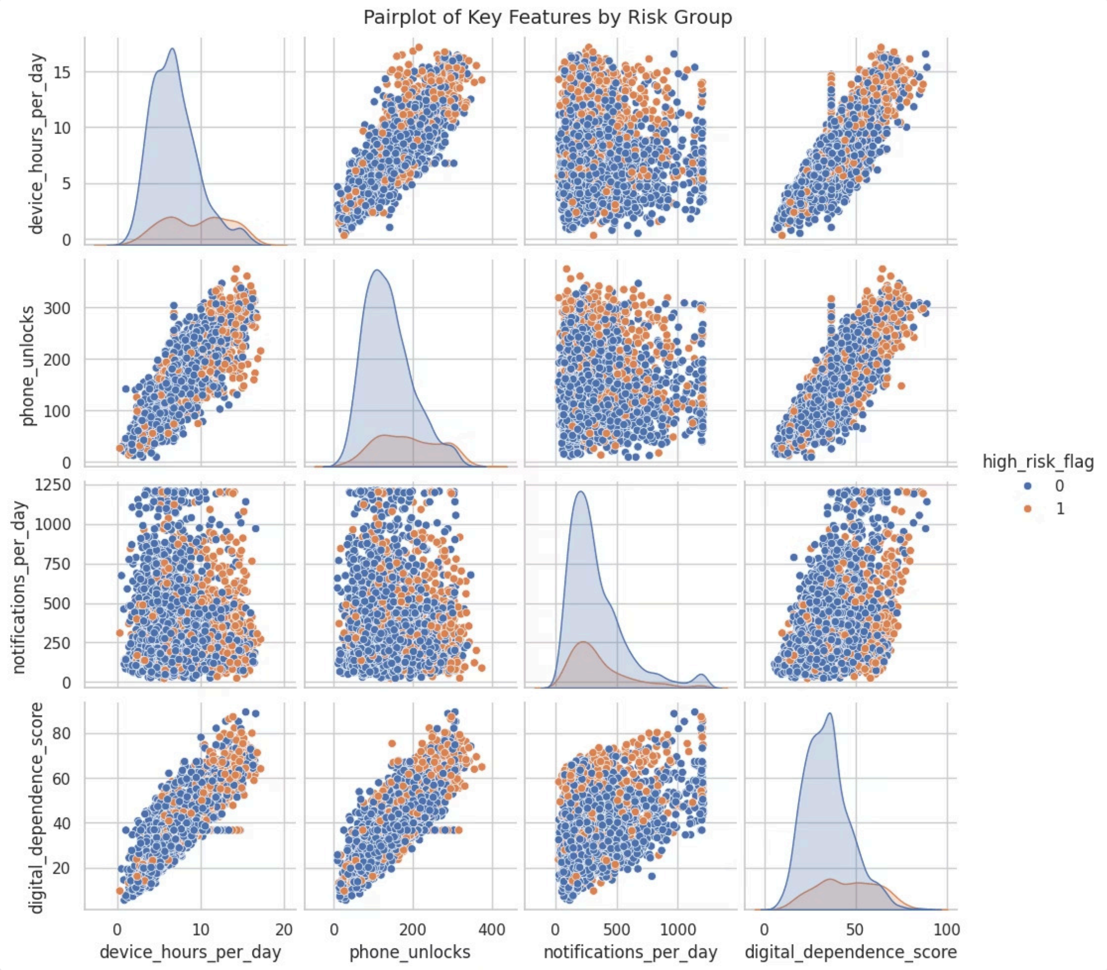
- 3,500 samples, 24 features
- Demographics, device use, mental-health, lifestyle
- Target: high_risk_flag
- 80% low risk, 20% high risk
- No missing values

High-risk patterns:

- More screen time, unlocks, notifications
- Higher anxiety/depression/stress
- Lower sleep and happiness

```
[5] df.info()
    0s

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3500 entries, 0 to 3499
Data columns (total 24 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   id                        3500 non-null   int64
 1   age                       3500 non-null   int64
 2   gender                    3500 non-null   object
 3   region                    3500 non-null   object
 4   income_level              3500 non-null   object
 5   education_level           3500 non-null   object
 6   daily_role                3500 non-null   object
 7   device_hours_per_day      3500 non-null   float64
 8   phone_unlocks             3500 non-null   int64
 9   notifications_per_day     3500 non-null   int64
 10  social_media_mins         3500 non-null   int64
 11  study_mins                3500 non-null   int64
 12  physical_activity_days    3500 non-null   float64
 13  sleep_hours               3500 non-null   float64
 14  sleep_quality             3500 non-null   float64
 15  anxiety_score             3500 non-null   float64
 16  depression_score          3500 non-null   float64
 17  stress_level              3500 non-null   float64
 18  happiness_score           3500 non-null   float64
 19  focus_score               3500 non-null   float64
 20  high_risk_flag            3500 non-null   int64
 21  device_type               3500 non-null   object
 22  productivity_score        3500 non-null   float64
 23  digital_dependence_score  3500 non-null   float64
dtypes: float64(11), int64(7), object(6)
memory usage: 656.4+ KB
```



Feature Engineering

Derived features:

- screen_time_per_unlock
- notifications_per_hour
- mental_health_load (anxiety + depression + stress)
- mood_balance (happiness – stress)

Captures behavior + psychological strain more effectively.

Preprocessing

- 20 numerical → StandardScaler
- 6 categorical → OneHotEncoder
- Train/test split 75/25 (stratified)
- SMOTE on training set only

Ensures clean, balanced, consistent inputs.

Preprocessing Pipeline Setup

[24]

```python
numeric_transformer = Pipeline(steps=[
    ("scaler", StandardScaler())
])

categorical_transformer = Pipeline(steps=[
    ("onehot", OneHotEncoder(handle_unknown="ignore", sparse_output=False))
])

preprocess = ColumnTransformer(
    transformers=[
        ("num", numeric_transformer, numeric_features),
        ("cat", categorical_transformer, categorical_features)
    ]
)
print("Preprocessing pipeline created.")
```

```
Preprocessing pipeline created.
```

Addressing Class Imbalance Using SMOTE

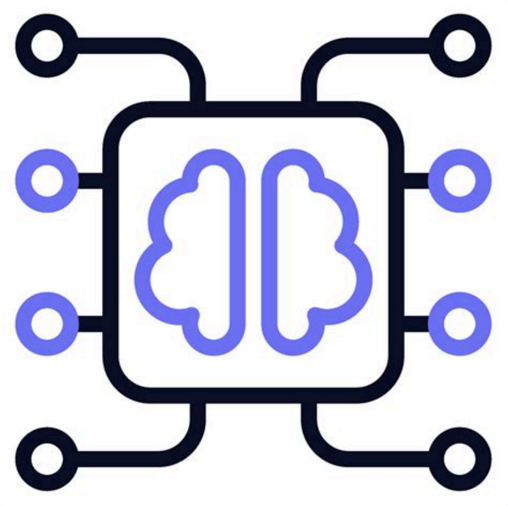
[26] 0s

```python
smote = SMOTE(random_state=42)
X_train_smote, y_train_smote = smote.fit_resample(X_train_t, y_train)

print("Shape of X_train after SMOTE:", X_train_smote.shape)
print("Distribution of y_train after SMOTE:\n", pd.Series(y_train_smote).value_counts(normalize=True))
```

```
Shape of X_train after SMOTE: (4192, 45)
Distribution of y_train after SMOTE:
 high_risk_flag
0    0.5
1    0.5
Name: proportion, dtype: float64
```

Machine Learning Models



Classification (High-Risk Prediction):

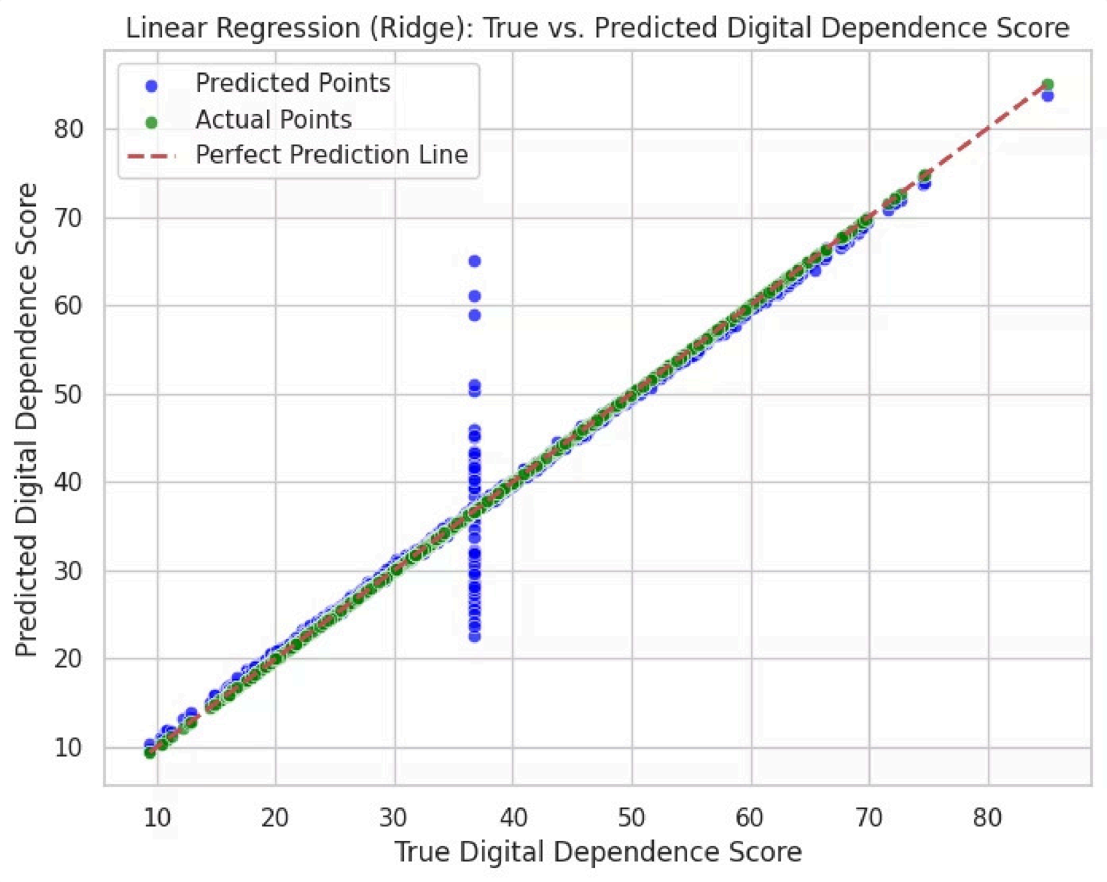
- Logistic Regression
- kNN Classifier
- Random Forest
- XGBoost Classifier
- SVM

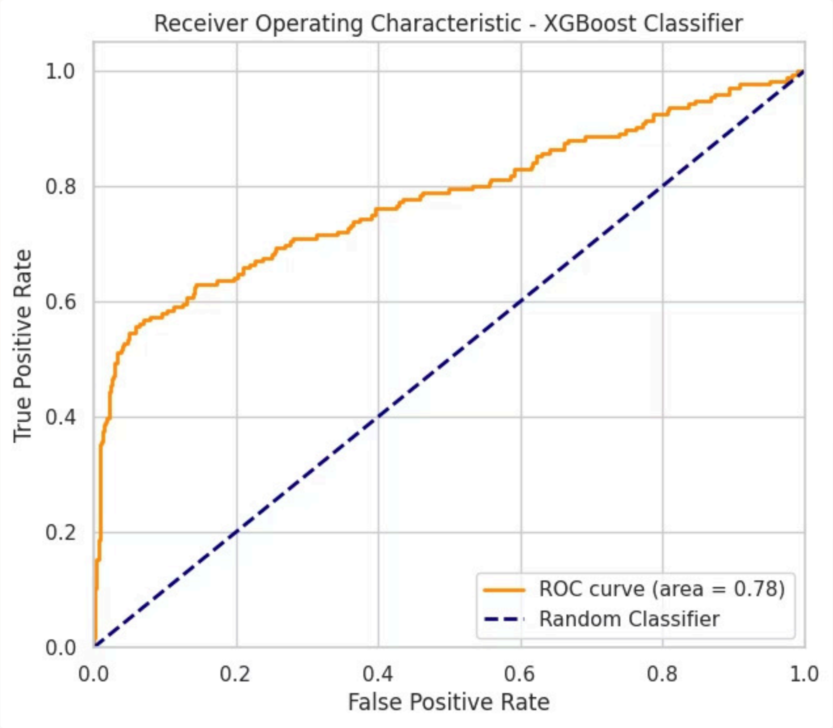
Regression (Digital Dependence Score):

- Linear Regression (Ridge)

Model Training

- Models trained on SMOTE-balanced training data.
- Pipelines ensured consistent scaling & encoding.
- Regression trained on original targets.
- SHAP used for feature importance.





Evaluation Metrics

Classification Metrics (High-Risk Prediction)

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.699	0.363	0.653	0.467
kNN Classifier	0.565	0.275	0.710	0.396
Random Forest	0.870	0.772	0.500	0.607
XGBoost Classifier	0.872	0.767	0.523	0.622
SVM Classifier	0.800	0.503	0.557	0.528

Regression Metrics (Digital Dependence Score)

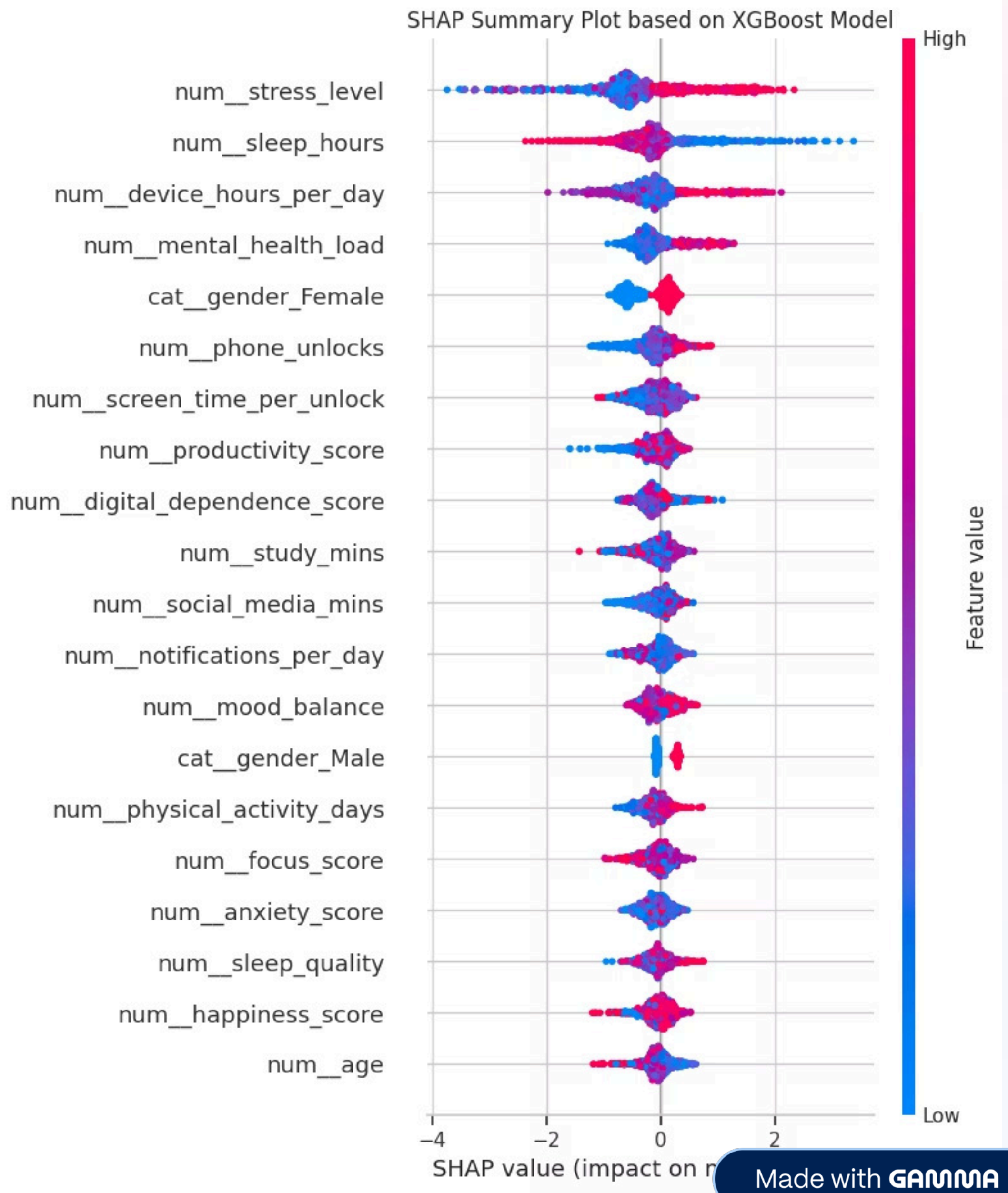
Model	MSE	R^2 Score
Linear Regression	5.396	0.970

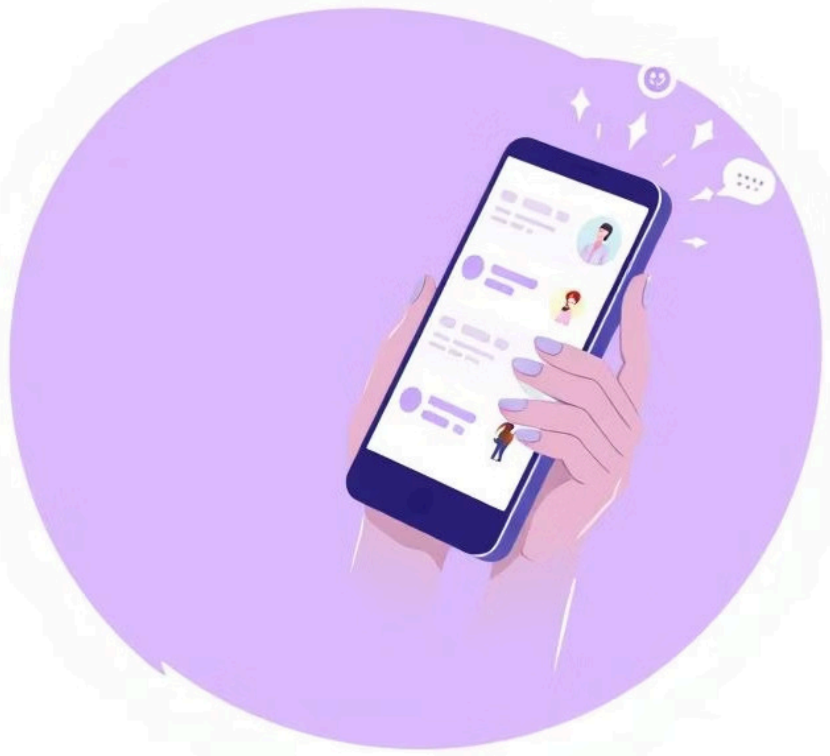
Results & Insights

- XGBoost & Random Forest outperform others.
- Logistic Regression stable; kNN limited by dimensionality.
- SVM solid but weaker than ensembles
- Linear Regression effectively predicts dependence (R^2 = 0.97).

Top Predictors (SHAP):

- Stress level, device hours per day, mental health load, phone unlocks, screen time per unlocks, productivity score etc.





Conclusion

Summary:

- Successfully predicted high-risk dopamine overload and digital dependence.
- Engineered features capture behavior and mental-health patterns.

Implications:

- Models can support early detection of digital overuse and mental-health strain.
- Insights may aid digital wellness tools, therapists or intervention systems.

Challenges:

- Balancing classes (SMOTE) without overfitting.
- Handling high-dimensional feature space (affecting kNN, SVM).
- Improving recall for some models to better capture high-risk individuals.

Thank You

Thank you for your time and attention while reviewing this project.