User Behaviour Detection using Machine Learning

Introduction

In today's digital world, every user action generates data. Whether a person is scrolling through an app, shopping online, or watching videos — all of it contributes to user behaviour. Companies use this data to understand:

- Who their active users are
- When users are likely to stop using the app (churn)
- What makes a user spend more time on their platform

The goal of this project is to build a machine learning model that can detect user behavior patterns. This helps businesses make better decisions and provide a more personalized experience to users.

We use a real-world dataset that includes:

- Device information (like iPhone, Samsung)
- User details (like gender)
- Activity data (like session time and clicks)
- And a label that tells us if the user is active or inactive

Dataset Overview :

We used a dataset named user_behavior_dataset.csv. It contains detailed records of multiple users and their behaviour patterns while using a mobile app.

✓ Important Columns:

Column Name	Description
Device Model	Brand of phone (e.g., iPhone, Samsung)
Operating System	OS used - Android or iOS
Gender	Male or Female
Session Duration (min)	How long the user used the app (in minutes)
Clicks	Number of clicks/touches made in the app
User Behaviour Class	Target label: 1 = active, 0 = inactive

We removed the column User ID because it doesn't help in predicting behavior.

Data Preprocessing :

Before we can train a machine learning model, we must **clean and transform the data** into a format the model can understand.

✓ Label Encoding

We converted text columns like Device Model, Operating System, and Gender into numbers using a method called **Label Encoding**.

✓ Feature Scaling

Machine learning models perform better when all features are in the same range. We used **StandardScaler** to normalize the values like clicks and duration.

✓ Train-Test Split

We split the data:

- **80% for training** (to learn)
- **20% for testing** (to evaluate performance)

✓ Handling Imbalance with SMOTE

Sometimes, the dataset has more active users and fewer inactive ones. This is called **class imbalance**. To fix this, we used **SMOTE (Synthetic Minority Over-sampling Technique)** which generates new examples of the minority class to balance the dataset.

Machine Learning Models Used :

We used and compared four popular ML algorithms:

Model	Description
Random Forest	Uses multiple decision trees. Very accurate and explains feature importance well.
Logistic Regression	Simple model for binary classification. Interpretable and fast.
Support Vector Machine (SVM)	Finds the best dividing line between classes. Works well with small and medium datasets.
K-Nearest Neighbors (KNN)	Predicts a user's class based on how nearby users behaved. Easy to understand.

Model Evaluation & Metrics :

We evaluated each model using:

- **Accuracy**: Total correct predictions / total predictions
- **Recall**: How well the model finds all actual active/inactive users

- **F1 Score**: Balance between precision and recall
- **Cross-Validation Accuracy**: How well the model performs across different parts of the data (5-fold)

Model	Accuracy	Recall	F1 Score	Cross-Validation Accuracy
Random Forest	0.87	0.86	0.85	0.84
Logistic Regression	0.82	0.80	0.81	0.80
Support Vector Machine (SVM)	0.84	0.83	0.83	0.82
K-Nearest Neighbors (KNN)	0.78	0.77	0.76	0.75

Feature Importance :

We wanted to understand **which features influence the prediction the most**. So we checked:

✓ Random Forest - Feature Importance

- **Session Duration** and **Clicks** were the top features.
- This makes sense: longer sessions and more clicks usually mean active users.

✓ Logistic Regression - Coefficients

- **Operating System** and **Device Model** had moderate influence.
- These models give us more explainable insights.

Predicting New User Behaviour :

We tested the models on unseen user data — 5 new users with details like device, OS, gender, session time, and clicks.

✓ Example New User:

Device	OS	Gender	Session Time	Clicks
Samsung S21	Android	Male	12 min	22

We encoded and scaled this new data and predicted using all 4 trained models.

✓ Example Predictions:

- Random Forest: [1, 0, 1, 1, 0]
- SVM: [1, 1, 1, 1, 1]

This shows how different models interpret the same user behaviour differently.

Results Comparison Table :

Model	Accuracy	Recall	F1 Score	CV Accuracy
Random Forest	0.87	0.86	0.85	0.84
Logistic Regression	0.82	0.80	0.81	0.80
SVM	0.84	0.83	0.83	0.82
KNN	0.78	0.77	0.76	0.75

Conclusion: Random Forest gave the best results overall. SVM and Logistic Regression were also good and fast.

Conclusion:

This project showed how to build an end-to-end machine learning pipeline to detect user behavior from real-world data.

✓ We performed all important ML steps:

- Data cleaning & transformation
- Feature encoding and scaling
- Handling class imbalance
- Training multiple models
- Evaluating with proper metrics
- Predicting new data

✓ The Random Forest model performed best.
✓ We learned how session time and click behavior are key indicators of user activity.

Future Scope :

Here's how we can improve the project in the future:

- Add more features like screen time, app category, in-app purchases
- Use more advanced models like **XGBoost**, **Neural Networks**

- Deploy the model as a **web app or dashboard**
- Continuously update the model with new data
- Analyze **churn prediction** (when users stop using the app)

👨‍💻 Project Created By

Neer Raichura 92310151009

Ghanshyam Pansuriya 92310151012

Jash Upadhyay 92310151003

Charvit Ponkhiya 92310151001

B.Tech – Artificial Intelligence

Marwadi University