🔧 Cell 0 — Setup & Imports

This cell initializes the environment and imports all necessary libraries for:

- Data handling and visualization
- Machine learning models
- Text vectorization and scaling output results

We also ensure that required directories (`models/baseline/`, `results/baseline/`) are created beforehand.

```
# Import required packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import os, joblib, time, warnings, sys

# Ignore warnings
warnings.filterwarnings("ignore")

# Show all columns when printing dataframes
pd.set_option("display.max_columns", None)

# Create directories if they don't exist
os.makedirs("models/baseline", exist_ok=True)
os.makedirs("results/baseline", exist_ok=True)

# Confirm environment setup
print("✅ Environment ready.")
print("Python version:", sys.version.split()[0])
print("Pandas version:", pd.__version__)
print("Numpy version:", np.__version__)
```

```
✅ Environment ready.
Python version: 3.10.12
Pandas version: 2.2.2
Numpy version: 1.26.4
```

🧹 Cell 1 — Load Final Processed User Feature Data

This cell loads the preprocessed dataset `user_features_baseline.csv`. Each contains user-level aspects aggregated and a short label.

- The dataset has already been cleaned and aggregated in earlier notebooks.
- This step only checks whether we will use a fixed and stratified train/test split.

```
# Load the processed user feature data
df = pd.read_csv("data/processed/user_features_baseline.csv")
print(f"✅ Loaded {df.shape[0]} rows, {df.shape[1]} columns")

# Show first few rows to confirm structure
display(df.head())
```

```
✅ Loaded 12,000 rows, 12 columns
```

	user_id	avg_time	total_time	mention_count	ratio_mention	hour_std	active_days	sentiment_mean	sentiment_std	sentiment_max	sentiment_min	label_sentiment
0	1_0_0_0

📘 Cell 8 — Final Summary

All machine learning tasks are complete.

✅ Loaded and cleaned the user feature dataset.
✅ Trained and evaluated 6 ML models:

- Logistic Regression
- Random Forest
- Gradient Boosting
- XGBoost
- LightGBM
- SVM

🏆 **Best performing model:** {best_model_name}
📊 Accuracy & confusion matrix visuals saved to: `outputs/visualizations/`
📂 Final predictions saved to: `outputs/final_predictions.csv`

[152]:
```
print("📘 Final Summary")
print("✅ Loaded and cleaned user features.")
print("✅ Trained and evaluated 6 ML models:")
for name, acc in results:
    print(f"   - {name}: {acc * 100:.2f}% accuracy")

print(f"\n🏆 Best model: {best_model_name}")
print("📊 Accuracy & confusion matrix visuals saved to outputs/visualizations/")
print("📂 Final predictions saved to outputs/final_predictions.csv")
```

```
📘 Final Summary
✅ Loaded and cleaned user features.
✅ Trained and evaluated 6 ML models:
   - Logistic Regression: 66.28% accuracy
   - Random Forest: 68.22% accuracy
   - Gradient Boosting: 67.83% accuracy
   - XGBoost: 68.02% accuracy
   - LightGBM: 65.89% accuracy
   - SVM: 67.05% accuracy

🏆 Best model: Random Forest
📊 Accuracy & confusion matrix visuals saved to outputs/visualizations/
📂 Final predictions saved to outputs/final_predictions.csv
```