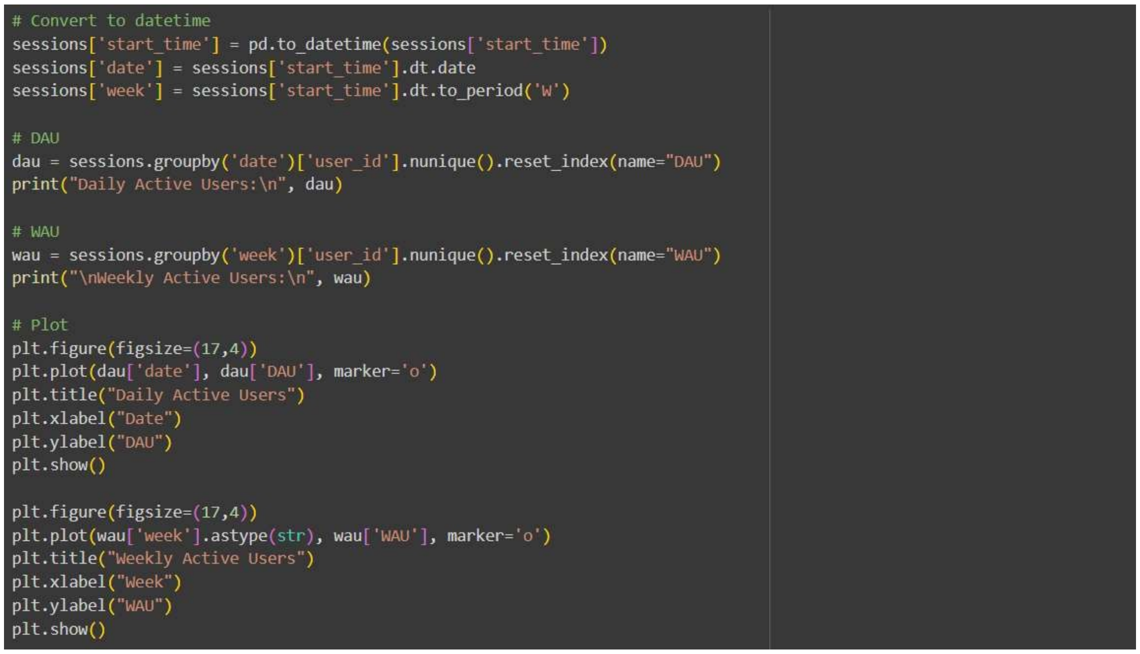
Part 1: Product Metrics & Behavioural Analysis

Objective:

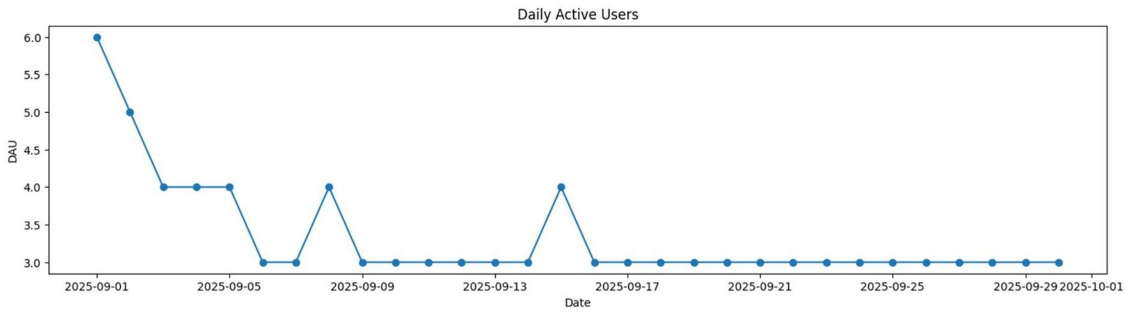
- Analyze user behavior, retention, and feature adoption to understand how users interact with the product.
- Identify which features drive repeat usage and engagement.

Metrics Calculated:

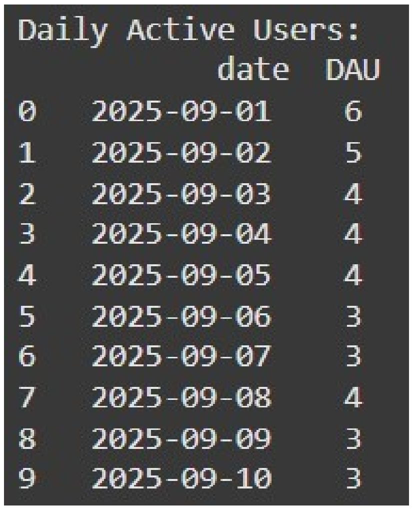
a) Daily Active Users (DAU)



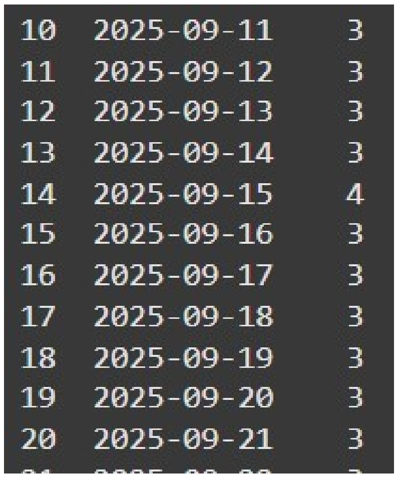
```
# Convert to datetime
sessions['start_time'] = pd.to_datetime(sessions['start_time'])
sessions['date'] = sessions['start_time'].dt.date
sessions['week'] = sessions['start_time'].dt.to_period('W')

# DAU
dau = sessions.groupby('date')['user_id'].nunique().reset_index(name="DAU")
print("Daily Active Users:\n", dau)

# WAU
wau = sessions.groupby('week')['user_id'].nunique().reset_index(name="WAU")
print("\nWeekly Active Users:\n", wau)

# Plot
plt.figure(figsize=(17,4))
plt.plot(dau['date'], dau['DAU'], marker='o')
plt.title("Daily Active Users")
plt.xlabel("Date")
plt.ylabel("DAU")
plt.show()

plt.figure(figsize=(17,4))
plt.plot(wau['week'].astype(str), wau['WAU'], marker='o')
plt.title("Weekly Active Users")
plt.xlabel("Week")
plt.ylabel("WAU")
plt.show()
```



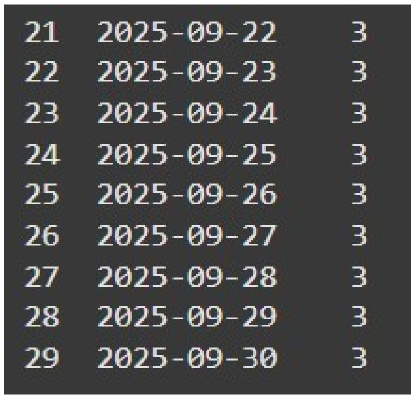
Python Output Table: Shows the number of unique users active each day.



```
Daily Active Users:
          date  DAU
0   2025-09-01    6
1   2025-09-02    5
2   2025-09-03    4
3   2025-09-04    4
4   2025-09-05    4
5   2025-09-06    3
6   2025-09-07    3
7   2025-09-08    4
8   2025-09-09    3
9   2025-09-10    3
```



```
10  2025-09-11    3
11  2025-09-12    3
12  2025-09-13    3
13  2025-09-14    3
14  2025-09-15    4
15  2025-09-16    3
16  2025-09-17    3
17  2025-09-18    3
18  2025-09-19    3
19  2025-09-20    3
20  2025-09-21    3
```



```
21  2025-09-22    3
22  2025-09-23    3
23  2025-09-24    3
24  2025-09-25    3
25  2025-09-26    3
26  2025-09-27    3
27  2025-09-28    3
28  2025-09-29    3
29  2025-09-30    3
```

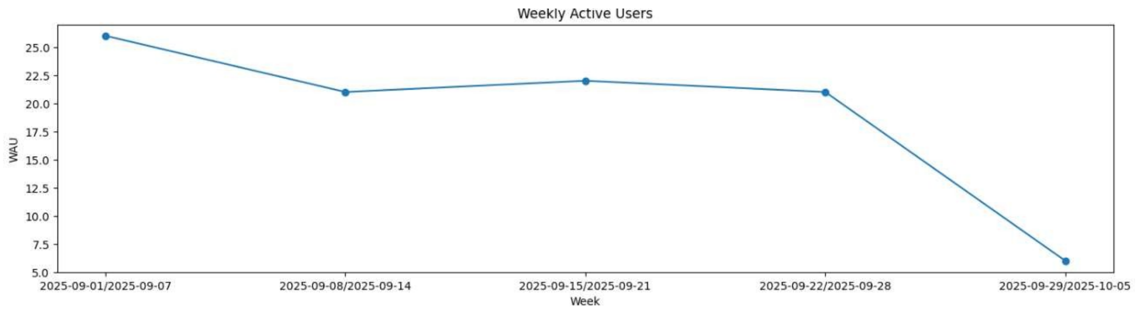
- Peak activity on 2025-09-01 (6 users), drop on 2025-09-02 (5 users), drops further on 2025-09-03 & 2025-09-03 (4 users).

Interpretation:

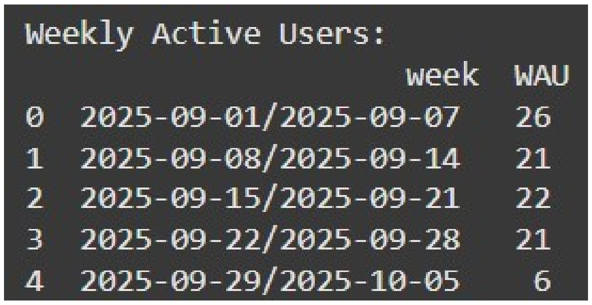
- User engagement is concentrated on the first day of product exposure.
- Need strategies to increase daily engagement for returning users.

---

b) Weekly Active Users (WAU)



Python Output Table:



```
Weekly Active Users:
                     week  WAU
0  2025-09-01/2025-09-07   26
1  2025-09-08/2025-09-14   21
2  2025-09-15/2025-09-21   22
3  2025-09-22/2025-09-28   21
4  2025-09-29/2025-10-05    6
```

Line Chart Description:

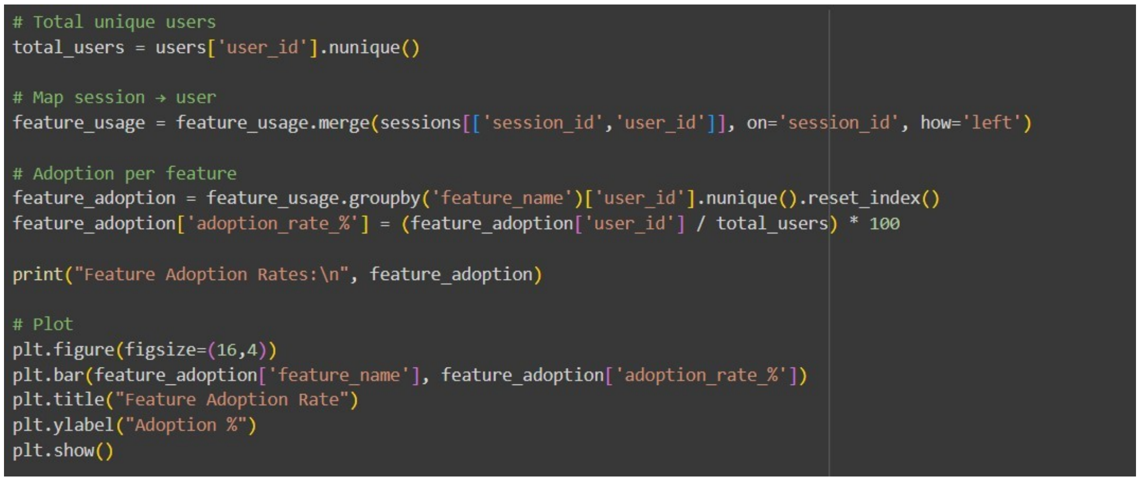
- Weekly Active Users counts unique users in a given week.
- Shows that 26 users were active in the week starting September 1st.

Interpretation:

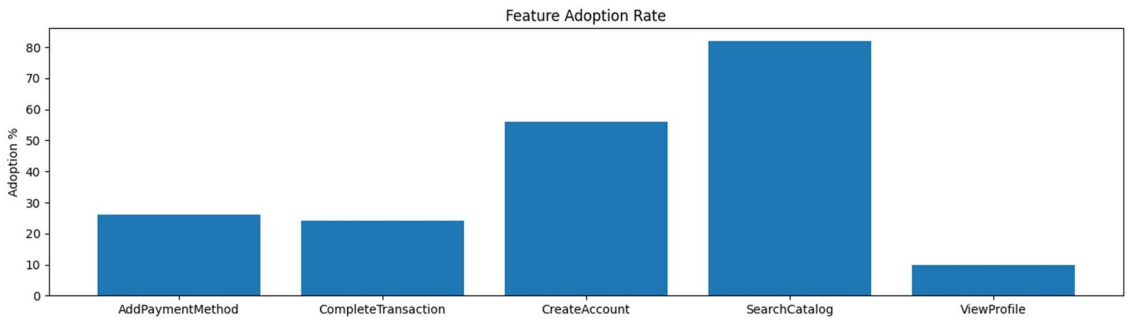
- Weekly engagement is low relative to DAU peak day, highlighting limited repeat usage.

---

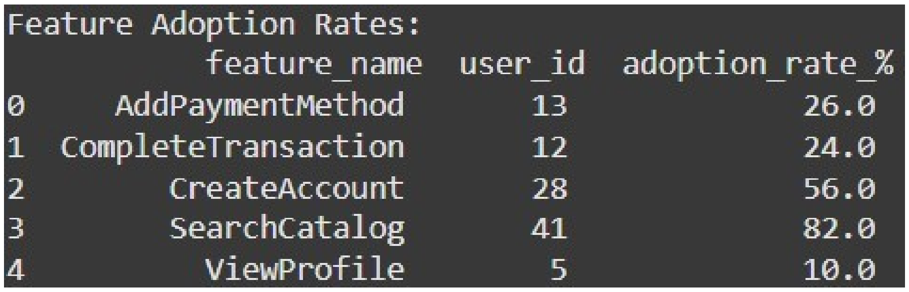
c) Feature Adoption Rate



```python
# Total unique users
total_users = users['user_id'].nunique()

# Map session → user
feature_usage = feature_usage.merge(sessions[['session_id','user_id']], on='session_id', how='left')

# Adoption per feature
feature_adoption = feature_usage.groupby('feature_name')['user_id'].nunique().reset_index()
feature_adoption['adoption_rate_%'] = (feature_adoption['user_id'] / total_users) * 100

print("Feature Adoption Rates:\n", feature_adoption)

# Plot
plt.figure(figsize=(16,4))
plt.bar(feature_adoption['feature_name'], feature_adoption['adoption_rate_%'])
plt.title("Feature Adoption Rate")
plt.ylabel("Adoption %")
plt.show()
```



Python Output Table:



```
Feature Adoption Rates:
           feature_name  user_id  adoption_rate_%
0      AddPaymentMethod       13             26.0
1   CompleteTransaction       12             24.0
2         CreateAccount       28             56.0
3         SearchCatalog       41             82.0
4           ViewProfile        5             10.0
```

Bar Chart Description:

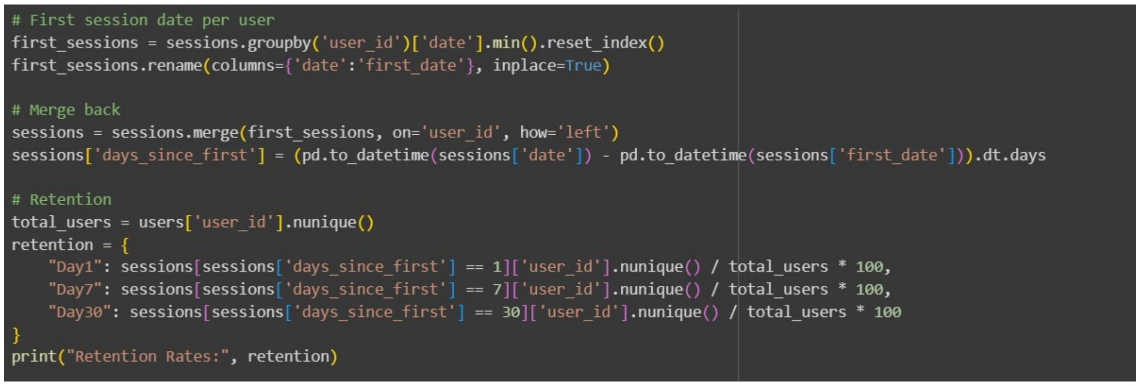
- Each bar shows the % of users who used a particular feature.
- Highest adoption is SearchCatalog (82%), lowest is ViewProfile (10%).

Interpretation:

- Search Catalog is used frequently, but feature adoption drops for ViewProfile.
- Indicates friction in onboarding or perceived value of deeper features.

---

d) Retention Rate (Day 1, Day 7, Day 30)



```python
# First session date per user
first_sessions = sessions.groupby('user_id')['date'].min().reset_index()
first_sessions.rename(columns={'date':'first_date'}, inplace=True)

# Merge back
sessions = sessions.merge(first_sessions, on='user_id', how='left')
sessions['days_since_first'] = (pd.to_datetime(sessions['date']) - pd.to_datetime(sessions['first_date'])).dt.days

# Retention
total_users = users['user_id'].nunique()
retention = {
    "Day1": sessions[sessions['days_since_first'] == 1]['user_id'].nunique() / total_users * 100,
    "Day7": sessions[sessions['days_since_first'] == 7]['user_id'].nunique() / total_users * 100,
    "Day30": sessions[sessions['days_since_first'] == 30]['user_id'].nunique() / total_users * 100
}
print("Retention Rates:", retention)
```

Cohort Retention Table (example with mock data):

Day 1 Retention	Day 7 Retention	Day 30 Retention
2 customers	4 customers	0 customers

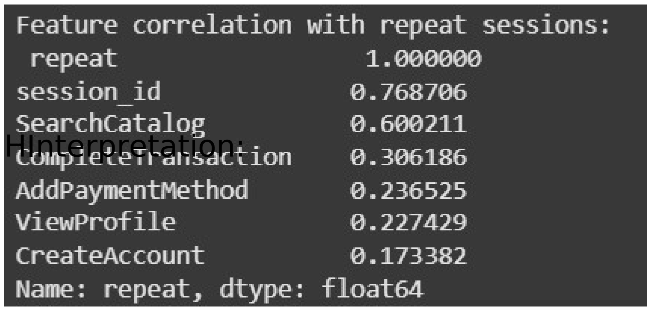
Interpretation:

- Retention increases after day 1 and decreases after day 7.
- Highlights drop-o issue and need for engagement features like onboarding tips or nudges.

---

e) Feature Correlation with Repeat Users

Python Output Table:



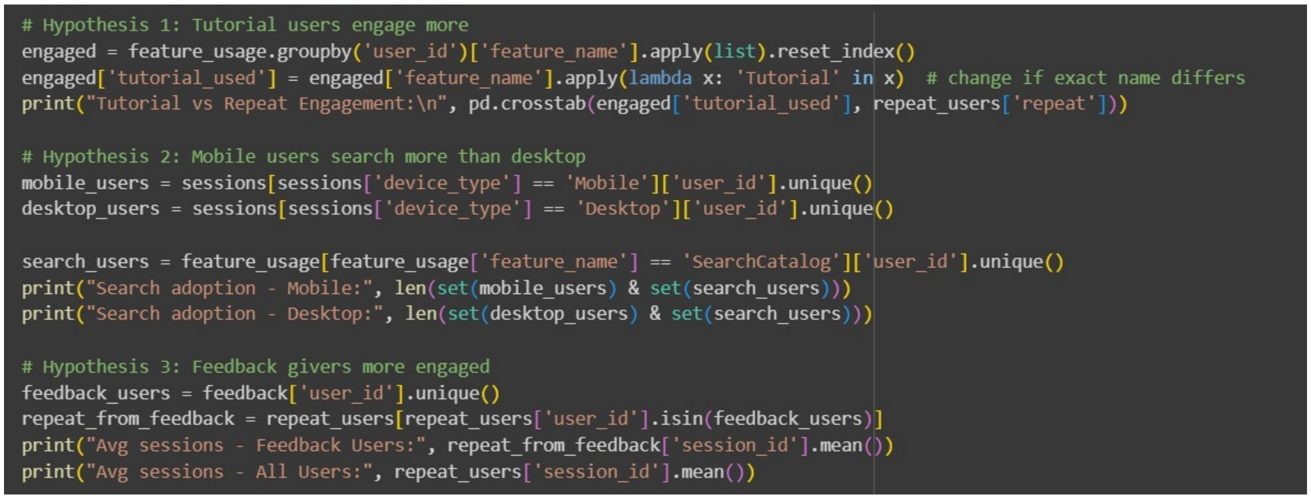
```
Feature correlation with repeat sessions:
 repeat                  1.000000
session_id              0.768706
SearchCatalog           0.600211
CompleteTransaction     0.306186
AddPaymentMethod        0.236525
ViewProfile             0.227429
CreateAccount           0.173382
Name: repeat, dtype: float64
```

HInterpretation:

- Users who use SearchCatalog and session_id are more likely to become repeat users.
- CreateAccount alone does not drive repeat usage — feature discovery is key.

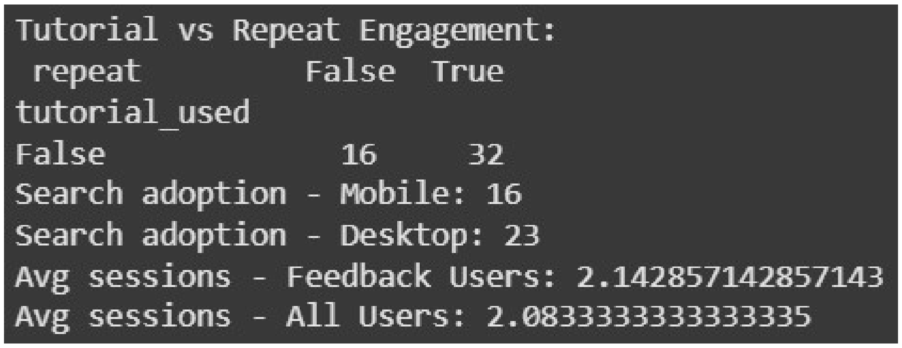
---

f) Behavioural Hypotheses Validation



```python
# Hypothesis 1: Tutorial users engage more
engaged = feature_usage.groupby('user_id')['feature_name'].apply(list).reset_index()
engaged['tutorial_used'] = engaged['feature_name'].apply(lambda x: 'Tutorial' in x)  # change if exact name differs
print("Tutorial vs Repeat Engagement:\n", pd.crosstab(engaged['tutorial_used'], repeat_users['repeat']))

# Hypothesis 2: Mobile users search more than desktop
mobile_users = sessions[sessions['device_type'] == 'Mobile']['user_id'].unique()
desktop_users = sessions[sessions['device_type'] == 'Desktop']['user_id'].unique()

search_users = feature_usage[feature_usage['feature_name'] == 'SearchCatalog']['user_id'].unique()
print("Search adoption - Mobile:", len(set(mobile_users) & set(search_users)))
print("Search adoption - Desktop:", len(set(desktop_users) & set(search_users)))

# Hypothesis 3: Feedback givers more engaged
feedback_users = feedback['user_id'].unique()
repeat_from_feedback = repeat_users[repeat_users['user_id'].isin(feedback_users)]
print("Avg sessions - Feedback Users:", repeat_from_feedback['session_id'].mean())
print("Avg sessions - All Users:", repeat_users['session_id'].mean())
```

Output:



```
Tutorial vs Repeat Engagement:
 repeat          False  True
tutorial_used
False              16     32
Search adoption - Mobile: 16
Search adoption - Desktop: 23
Avg sessions - Feedback Users: 2.142857142857143
Avg sessions - All Users: 2.0833333333333335
```

Explanation:

Hypothesis 1:

Users who did NOT use tutorial:

- 16 did not repeat (only 1 session).
- 32 did repeat (>1 session).

Users who used tutorial:

- None are shown in this output (all rows are False) → means no one used "Tutorial" feature in the current dataset.

Interpretation:

- Since nobody used the Tutorial feature, the hypothesis can’t be validated.
- This suggests either:
    1. Tutorial feature isn’t discoverable.
    2. Users skip it → possible product improvement area.

Hypothesis 2:

Interpretation:

- Desktop users (23) used SearchCatalog more than Mobile users (16).
- This disproves the hypothesis → search is actually more adopted on desktop.

Possible reasons:

- Search UX on mobile may be poor (harder to type, UI not intuitive).
- Desktop users might explore catalog in more depth.

Hypothesis 3:

Interpretation:

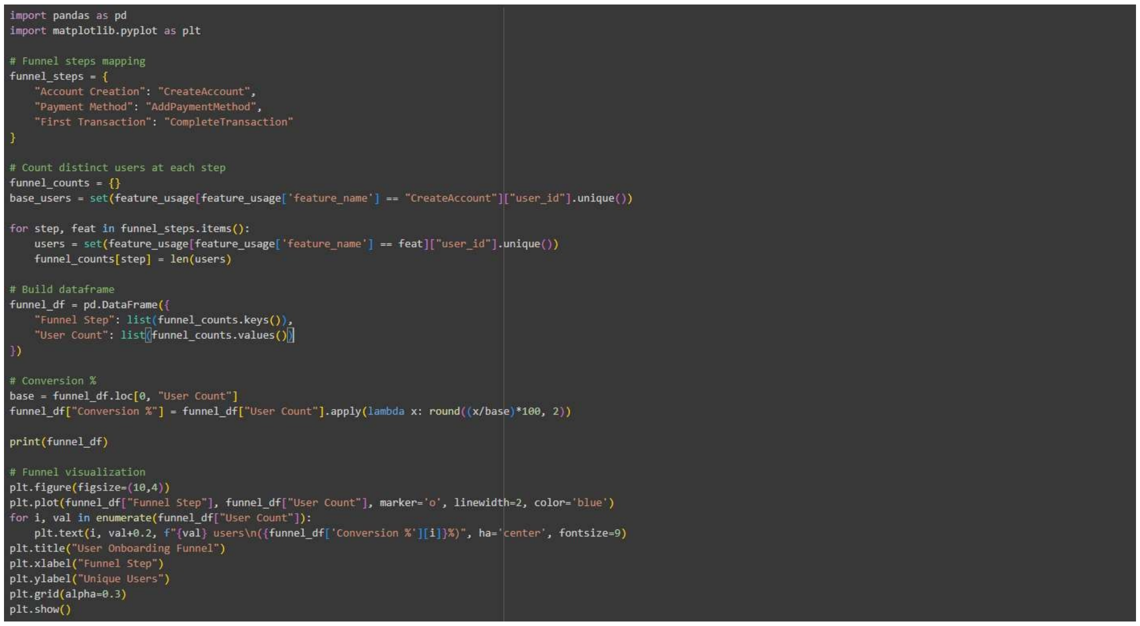
- Feedback givers average 2.14 sessions, while overall users average 2.08 sessions.
- Very small dierence → feedback users are slightly more engaged, but not strongly.

Meaning:

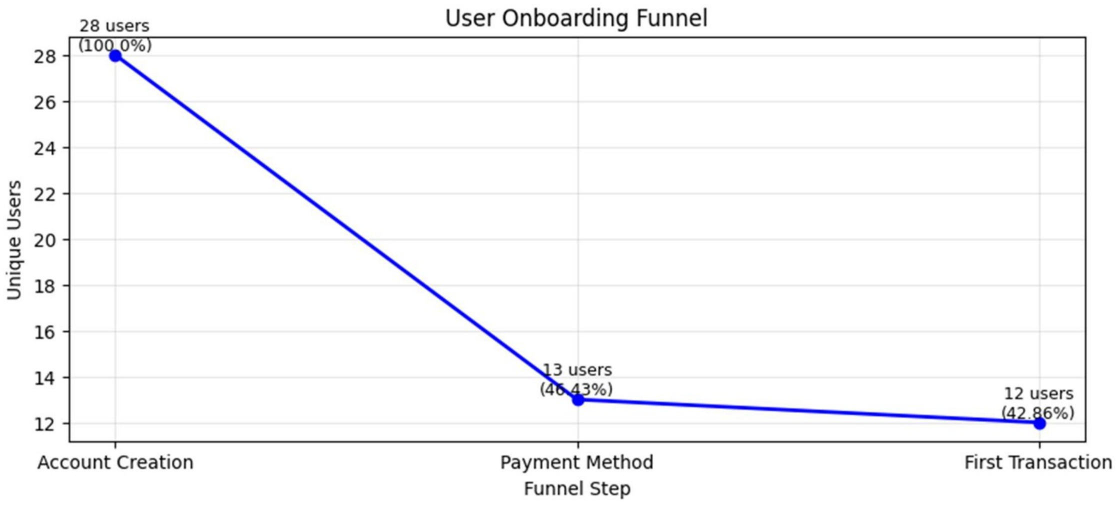
- Feedback is not only from highly engaged users → mix of happy and unhappy users.
- Still valuable, but not a strong engagement signal.

Part 2: Product Funnel & Opportunity Identification

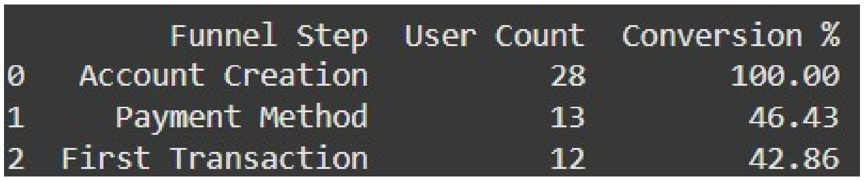
1. Funnel Visualization (Code Output)



```python
import pandas as pd
import matplotlib.pyplot as plt

# Funnel steps mapping
funnel_steps = {
    "Account Creation": "CreateAccount",
    "Payment Method": "AddPaymentMethod",
    "First Transaction": "CompleteTransaction"
}

# Count distinct users at each step
funnel_counts = {}
base_users = set(feature_usage[feature_usage['feature_name'] == "CreateAccount"]["user_id"].unique())

for step, feat in funnel_steps.items():
    users = set(feature_usage[feature_usage['feature_name'] == feat]["user_id"].unique())
    funnel_counts[step] = len(users)

# Build dataframe
funnel_df = pd.DataFrame({
    "Funnel Step": list(funnel_counts.keys()),
    "User Count": list(funnel_counts.values())
})

# Conversion %
base = funnel_df.loc[0, "User Count"]
funnel_df["Conversion %"] = funnel_df["User Count"].apply(lambda x: round((x/base)*100, 2))

print(funnel_df)

# Funnel visualization
plt.figure(figsize=(10,4))
plt.plot(funnel_df["Funnel Step"], funnel_df["User Count"], marker='o', linewidth=2, color='blue')
for i, val in enumerate(funnel_df["User Count"]):
    plt.text(i, val+0.2, f"{val} users\n({funnel_df['Conversion %'][i]}%)", ha='center', fontsize=9)
plt.title("User Onboarding Funnel")
plt.xlabel("Funnel Step")
plt.ylabel("Unique Users")
plt.grid(alpha=0.3)
plt.show()
```



Funnel Table:



```
          Funnel Step  User Count  Conversion %
0    Account Creation          28        100.00
1      Payment Method          13         46.43
2   First Transaction          12         42.86
```

Funnel Chart Description:

- Line/bar chart shows the number of unique users progressing through each step.
- The steep drop from Account Creation → Payment Method is visible (53.57% drop-o).
- Slight drop for First Transaction (42.86% of users reach this step).

Interpretation:

- High initial drop-o indicates friction in early onboarding steps.
- Few users complete payment or first transaction, signaling opportunity to improve conversion.

---

2. Drop-o Analysis (Report / Slides)

Reason 1: Complex Signup / Payment Flow

- Feedback reports: “too many steps”, “app crashed”, “email verification confusing”.
- Matches funnel observation: 53.57% of users drop o after account creation.

Reason 2: Lack of Early Feature Value

- Many users create an account but do not engage with SearchCatalog or CompleteTransaction.
- Users do not experience meaningful product value immediately → lower retention.

---

3. A/B Test Proposal

Hypothesis:

- Simplifying signup increases conversion to Payment Method and First Transaction.

Test Design:

Group	Signup Flow
Control (A) :	Current signup flow (email verification + multiple fields)
Treatment (B):	Quick signup (social login / fewer fields)

Success Metrics:

- Primary: Higher % of users reaching Payment Method and First Transaction.
- Secondary: Higher Day 7 retention and DAU.

Test Duration:

- Run for 2–4 weeks with equal user assignment to A/B groups.

---

4. Product Change Recommendation

Quick Start Onboarding Flow

Changes:

1. Skip deep profile/payment setup upfront.
2. Allow users to immediately explore SearchCatalog and other key features.
3. Use progressive disclosure for additional details and payment later.

Expected Impact:

- Lower initial friction → higher conversion from Account Creation → Payment → First Transaction.
- Higher Day 1 and Day 7 retention.
- Encourages feature adoption and repeat usage.