

E-Commerce Churn Prediction EDA

1. Event Types

- a. Cart
- b. View
- c. Purchase

2. 5 most popular products sold

```
a. df.filter(df['event_type'] ==  
    'purchase').select('product_id').groupBy('product_id').count()  
    .orderBy('count', ascending = False).show(5)
```

```
+-----+-----+  
|product_id|count|  
+-----+-----+  
| 1004856|28944|  
| 1004767|21806|  
| 1004833|12697|  
| 1005115|12543|  
| 4804056|12381|  
+-----+-----+
```

3. 5 Most popular brands on the platform

```
a. df.groupby('brand').count().orderBy('count',  
    ascending=False).show(6)
```

```
+-----+-----+  
| brand| count|  
+-----+-----+  
|samsung|5282775|  
| apple|4122554|  
| xiaomi|3083763|  
| huawei|1111205|  
|lucente| 655861|  
+-----+-----+
```

4. 5 Most popular product categories

```
a. df.groupby('category_code').count().orderBy('count',  
    ascending = False).show(6, truncate = False)
```

```
+-----+-----+  
|category_code      |count |  
+-----+-----+  
|electronics.smartphone |11507231|  
|electronics.clocks     |1311033 |
```

```
|computers.notebook      |1137623|
|electronics.video.tv   |1113750|
|electronics.audio.headphone|1100188|
+-----+-----+
```

5. Number of unique users on the platform

```
a. df.agg(countDistinct('user_id').alias('distinct_users')).show()
+-----+
|distinct_users|
+-----+
|      3022290|
```

6. The most active user on the platform

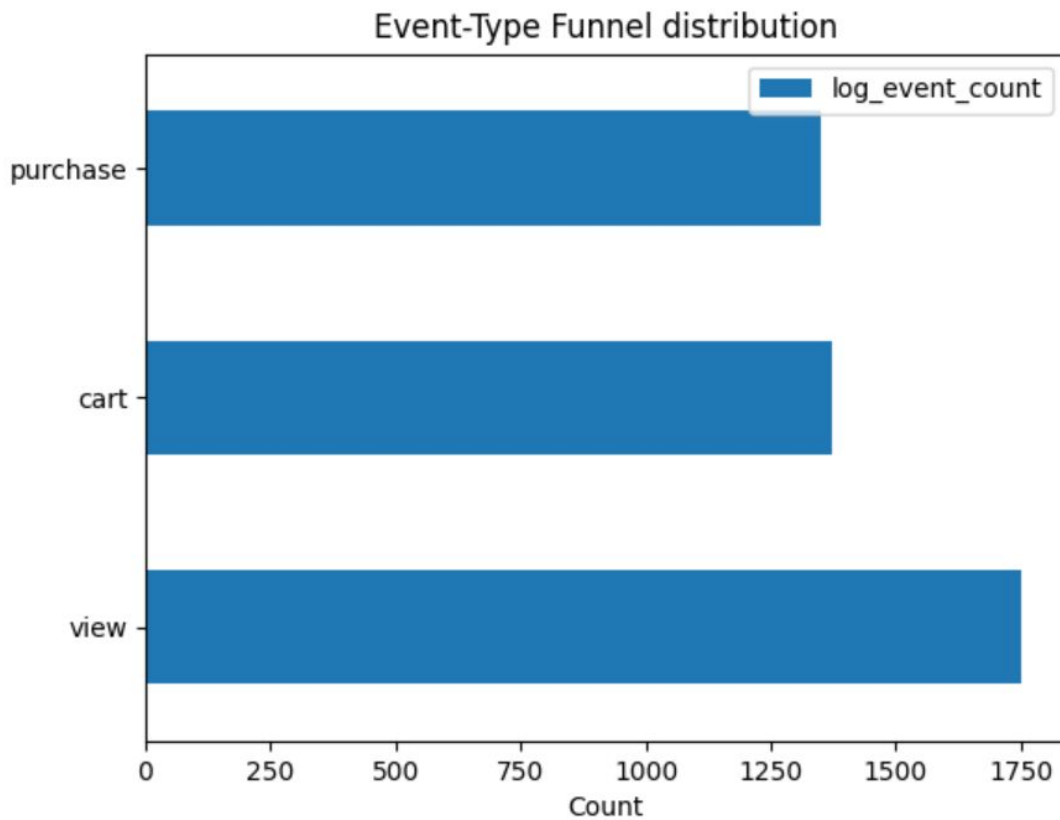
```
a. df.groupby('user_id').count().orderBy('count',
      ascending=False).show(1)
+-----+-----+
| user_id|count|
+-----+-----+
|512475445| 7436|
+-----+-----+
```

7. Average and maximum price for smartphones purchased by the customers

```
a. df.filter((df['category_code'] ==
      'electronics.smartphone') & (df['event_type'] ==
      'purchase')).agg(F.max(df['price']),
      F.avg('price')).show()
+-----+-----+
|max(price)|   avg(price)|
+-----+-----+
|  2110.45|464.6191130945663|
+-----+-----+
```

8. Event-type distribution of e-commerce shopping journey

```
a. from pyspark.sql.functions import log
b. event_df =
    df.groupby('event_type').count().orderBy('count',
      ascending = False).select('event_type', (log('count')*100)
      .alias('log_event_count')).toPandas()
```

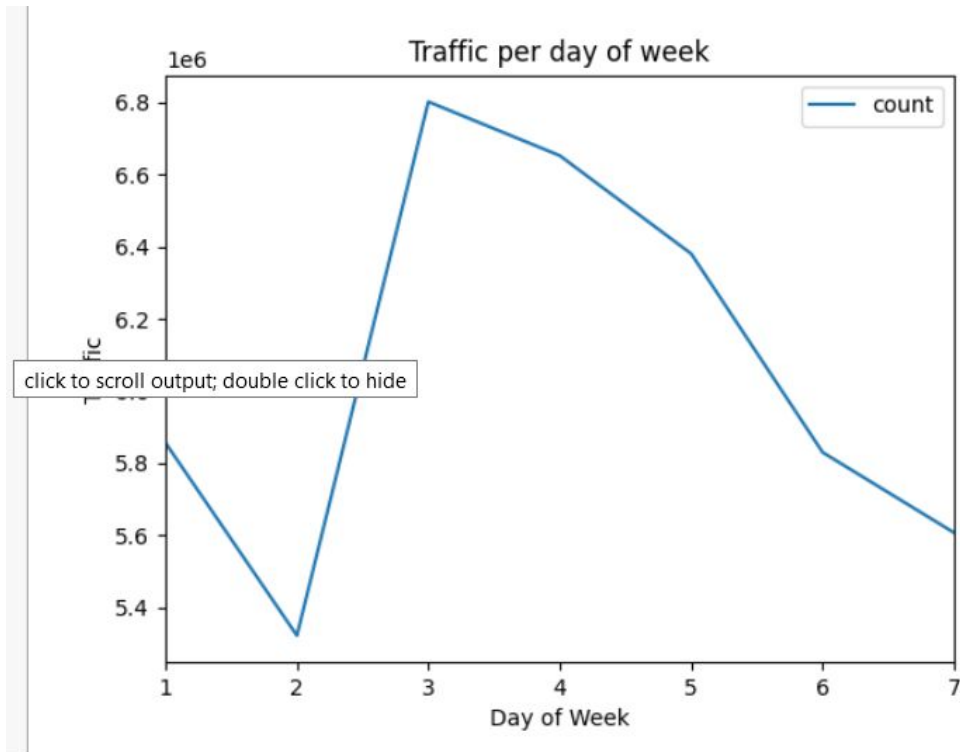


9. Traffic on different days of the week

```
from pyspark.sql.functions import to_timestamp
from pyspark.sql import functions as F
import matplotlib.pyplot as plt

df = df.withColumn('date_time', to_timestamp('event_time',
'yyyy-MM-dd HH:mm:ss'))
df = df.withColumn('day_of_week', F.dayofweek('date_time'))

dow_traffic = df.select('day_of_week')\
.groupby('day_of_week')\
.count()\
.orderBy('day_of_week', ascending = False)\
.toPandas()
plt.figure(figsize=(14,10))
dow_traffic.plot(y = 'count', x = 'day_of_week')
plt.title('Traffic per day of week')
plt.ylabel('Traffic')
plt.xlabel('Day of Week')
%matplotlib plt
```



This traffic pattern doesn't align with the intuition. The most traffic in the website in the month of October was on Wednesdays,