Retail Clickstream Analysis and Prediction

Niharika Krishnan - nk2982

Vaibhav Singh - vs2410

Introduction

- **2 Billion** people purchased goods online in 2021 - **USD $4.2 Trillion** in online retail sales
- **Clickstream Data:** User's digital footprint left on a specific website during a browsing session
- Dataset - User behavior data for October 2019 from a large multi-category online store
- Data collected by Open CDP project
- **Dataset Size: 5.6 GB**
- Why Big Data?
  - 5.6GB Dataset - Clickstream data for October 2019. For one year, ~ 60GB, making centralized computing futile
  - Use of Big Data technologies is imperative to make analytics computationally fast leading to valuable insights and making critical decisions
- **GOAL:**
  - Analyze key performance indicators and find insights to improve revenue
  - Provide insights for personalized digital marketing
  - Measure their marketing efforts, and optimize the overall user experience

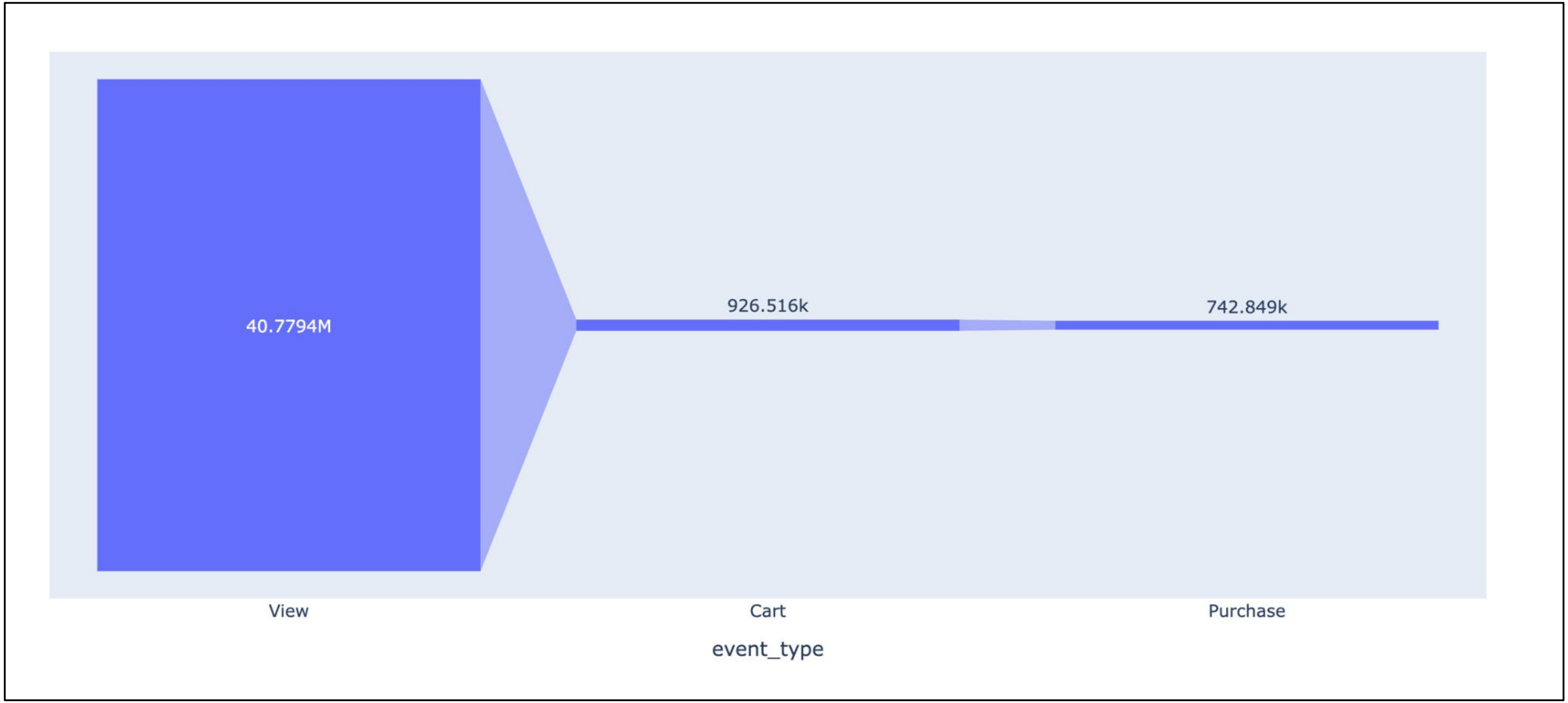
Dataset Overview

Tracking a User's Journey: Each row in the dataset represents an event

```
df.filter(df.user_session=='b37abd25-7672-4dd7-a098-40e50e314388').orderBy("event_time").toPandas()
```

	event_time	event_type	product_id	category_id	brand	price	user_id	user_session	category	product	Time	Day	Hour
0	2019-10-01 05:08:10 UTC	view	1005115	2053013555631882655	apple	975.57	526823608	b37abd25-7672-4dd7-a098-40e50e314388	electronics	smartphone	05:08:10	01	05
1	2019-10-01 05:08:24 UTC	view	1005115	2053013555631882655	apple	975.57	526823608	b37abd25-7672-4dd7-a098-40e50e314388	electronics	smartphone	05:08:24	01	05
2	2019-10-01 05:08:44 UTC	view	1005115	2053013555631882655	apple	975.57	526823608	b37abd25-7672-4dd7-a098-40e50e314388	electronics	smartphone	05:08:44	01	05
3	2019-10-01 05:13:03 UTC	view	1005115	2053013555631882655	apple	975.57	526823608	b37abd25-7672-4dd7-a098-40e50e314388	electronics	smartphone	05:13:03	01	05
4	2019-10-01 05:17:22 UTC	view	1003317	2053013555631882655	apple	957.53	526823608	b37abd25-7672-4dd7-a098-40e50e314388	electronics	smartphone	05:17:22	01	05
5	2019-10-01 05:18:23 UTC	view	1002524	2053013555631882655	apple	514.76	526823608	b37abd25-7672-4dd7-a098-40e50e314388	electronics	smartphone	05:18:23	01	05
6	2019-10-01 05:19:50 UTC	view	1005104	2053013555631882655	apple	975.57	526823608	b37abd25-7672-4dd7-a098-40e50e314388	electronics	smartphone	05:19:50	01	05
7	2019-10-01 05:20:05 UTC	view	1002629	2053013555631882655	apple	377.14	526823608	b37abd25-7672-4dd7-a098-40e50e314388	electronics	smartphone	05:20:05	01	05
8	2019-10-01 05:20:31 UTC	view	1003310	2053013555631882655	apple	746.29	526823608	b37abd25-7672-4dd7-a098-40e50e314388	electronics	smartphone	05:20:31	01	05
9	2019-10-01 05:21:10 UTC	view	1005121	2053013555631882655	apple	949.83	526823608	b37abd25-7672-4dd7-a098-40e50e314388	electronics	smartphone	05:21:10	01	05
10	2019-10-01 05:22:55 UTC	view	1004246	2053013555631882655	apple	735.01	526823608	b37abd25-7672-4dd7-a098-40e50e314388	electronics	smartphone	05:22:55	01	05
11	2019-10-01 05:23:51 UTC	view	1004249	2053013555631882655	apple	738.61	526823608	b37abd25-7672-4dd7-a098-40e50e314388	electronics	smartphone	05:23:51	01	05
12	2019-10-01 05:26:30 UTC	cart	1004249	2053013555631882655	apple	738.61	526823608	b37abd25-7672-4dd7-a098-40e50e314388	electronics	smartphone	05:26:30	01	05
13	2019-10-01 05:28:10 UTC	view	1005122	2053013555631882655	apple	1027.05	526823608	b37abd25-7672-4dd7-a098-40e50e314388	electronics	smartphone	05:28:10	01	05
14	2019-10-01 05:30:00 UTC	view	1004255	2053013555631882655	apple	744.39	526823608	b37abd25-7672-4dd7-a098-40e50e314388	electronics	smartphone	05:30:00	01	05
15	2019-10-01 05:30:12 UTC	view	1004252	2053013555631882655	apple	759.06	526823608	b37abd25-7672-4dd7-a098-40e50e314388	electronics	smartphone	05:30:12	01	05
16	2019-10-01 05:31:39 UTC	view	1004253	2053013555631882655	apple	816.52	526823608	b37abd25-7672-4dd7-a098-40e50e314388	electronics	smartphone	05:31:39	01	05
17	2019-10-01 05:34:23 UTC	cart	1004253	2053013555631882655	apple	816.52	526823608	b37abd25-7672-4dd7-a098-40e50e314388	electronics	smartphone	05:34:23	01	05
18	2019-10-01 05:34:32 UTC	cart	1004253	2053013555631882655	apple	816.52	526823608	b37abd25-7672-4dd7-a098-40e50e314388	electronics	smartphone	05:34:32	01	05
19	2019-10-01 05:36:23 UTC	view	1004253	2053013555631882655	apple	816.52	526823608	b37abd25-7672-4dd7-a098-40e50e314388	electronics	smartphone	05:36:23	01	05
20	2019-10-01 05:39:31 UTC	purchase	1004253	2053013555631882655	apple	816.52	526823608	b37abd25-7672-4dd7-a098-40e50e314388	electronics	smartphone	05:39:31	01	05
21	2019-10-01 05:40:10 UTC	view	1004253	2053013555631882655	apple	816.52	526823608	b37abd25-7672-4dd7-a098-40e50e314388	electronics	smartphone	05:40:10	01	05
22	2019-10-01 05:40:47 UTC	view	1004249	2053013555631882655	apple	738.61	526823608	b37abd25-7672-4dd7-a098-40e50e314388	electronics	smartphone	05:40:47	01	05

Analysis of User Behaviour

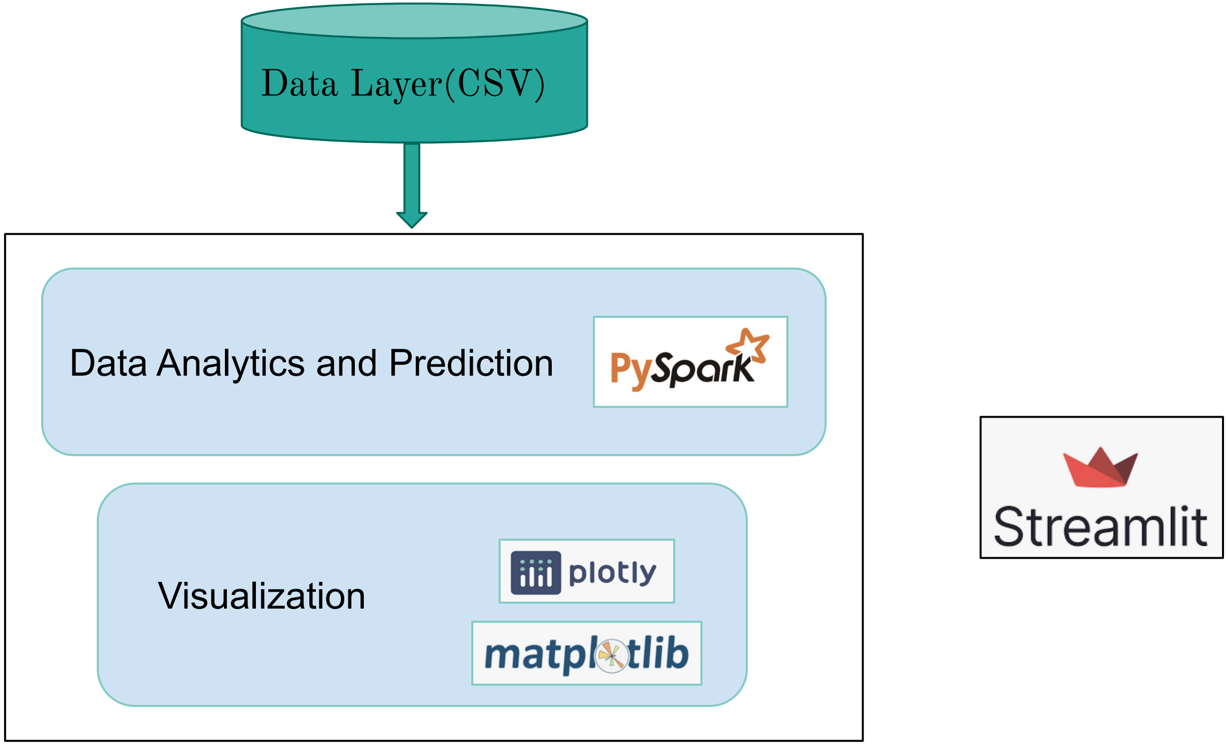


eCommerce Conversion Funnel

Objectives

1. Category Analysis
    a. Determine best performing categories on the e-commerce site based on purchases
    b. Find Brands that generate the highest traction in these best performing categories
2. Effect of Adding to Cart
    a. Correlation between impact Adding to Cart → Purchase: Cart Conversion Ratio
    b. Evaluate Cart Abandonment Rate across categories and brands
3. Effect of day-time on purchase trends
    a. Analyse the Purchase trends across the month
    b. Determine E-Commerce Prime Time
4. Build a Real-Time classification model that predicts a purchase using clickstream features

Architecture



Objective 1.a: Determining best performing categories

INSIGHTS:

- Electronics, Appliances and Computers are the most Browsed (View + Cart) and Purchased Categories

- Electronics highest revenue generating category

- Furniture - Highly browsed, Low - Purchase Rate (Preference to in-person viewing for comfort)

ACTIONS:

- Higher allocation of resources, data, manpower for Electronics. Onboard new vendors to keep the revenue incoming

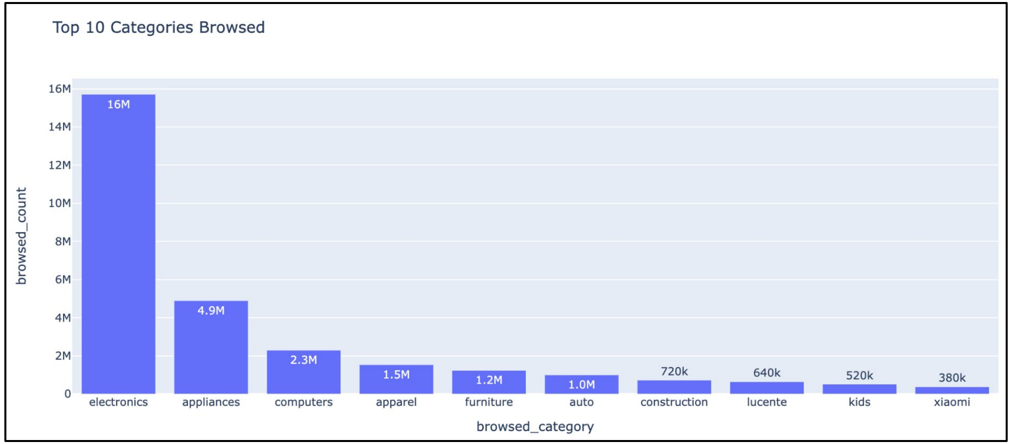
- Analyse the root cause of conversion sales in apparel/furniture. Introduce new 3D viewing techniques to bridge the gap between online & offline viewing experience

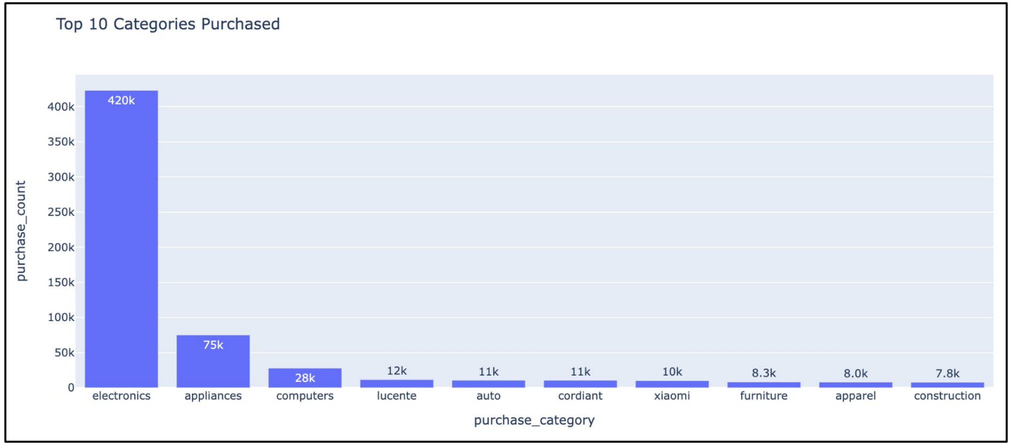
TECHNIQUES:

- Extract Category and Products from Category Code

- Data Imputation of Null values in Category

- GroupBy, Filters, UDF, Count, Bar Plots





Objective 1.b: Top Brands in Top Performing Categories

INSIGHTS:

- Samsung strong brand presence across Electronics and Appliances

- Samsung with 159k purchases - 38% of all electronic purchases

- Acer, Lenovo, HP are the leading players in Computers

ACTIONS:

- Onboard more products/vendors under these categories and brands

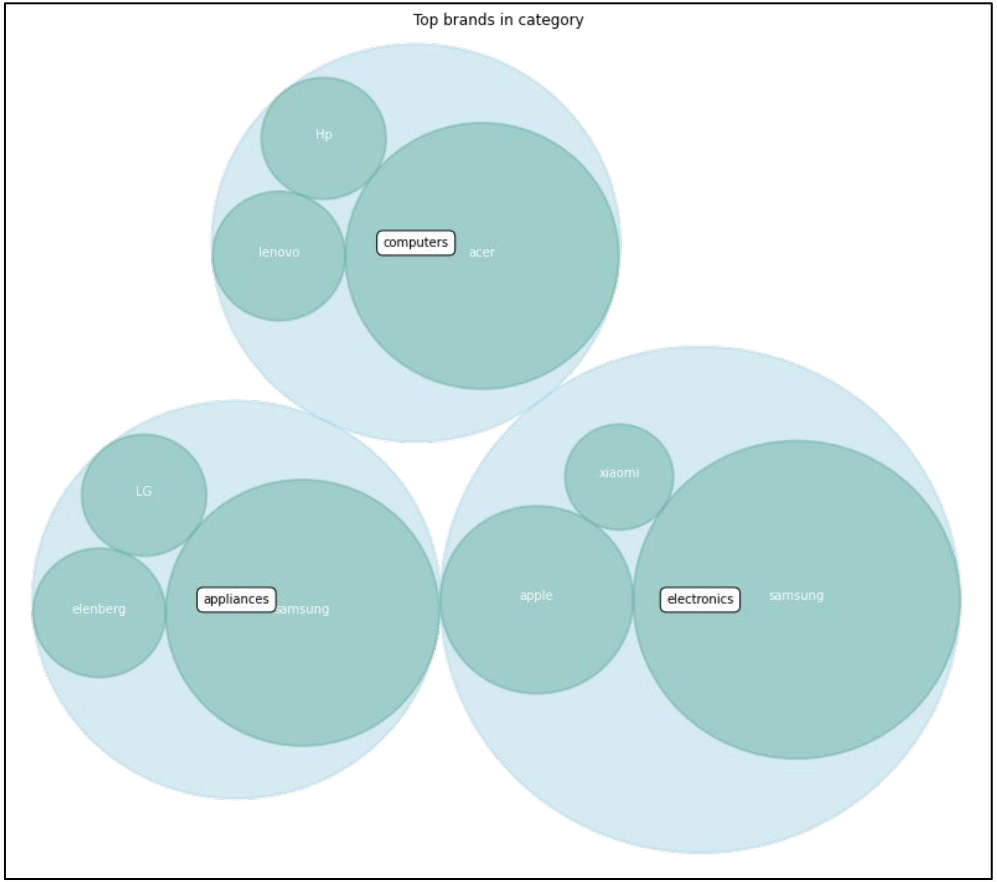
- Identify Brands top user-group for personalized targeted marketing

- Track brand reviews, perform sentiment analysis for brands that aren't performing well

TECHNIQUES:

- Window Functions for Rank, GroupBy

- Grouped Cluster Graphs



Objective 2.a: Effect of Adding to Cart with Purchase

INSIGHTS:

- 80% Cart Conversion Ratio (Cart: 926k, Purchase: 742k)

- Smartphones are the most added to the cart as well as highest purchased and have a 63% conversion rate

- However, clocks have a 90% conversion ratio and we believe it is because there are fewer options to choose from, hence the specs are pre-determined leading to a purchase

ACTIONS:

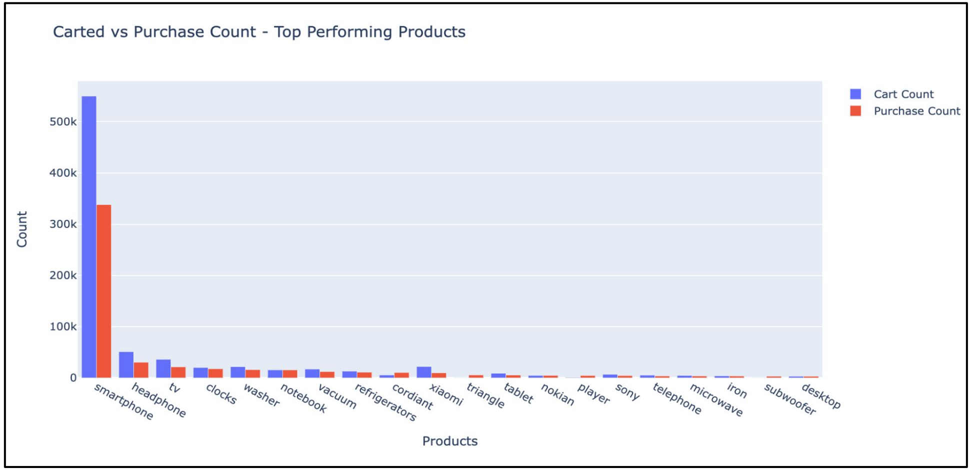
- Identify key reasons as to why products already in cart dont get purchased. The reasons might be due to better deals, return policy etc from other sites

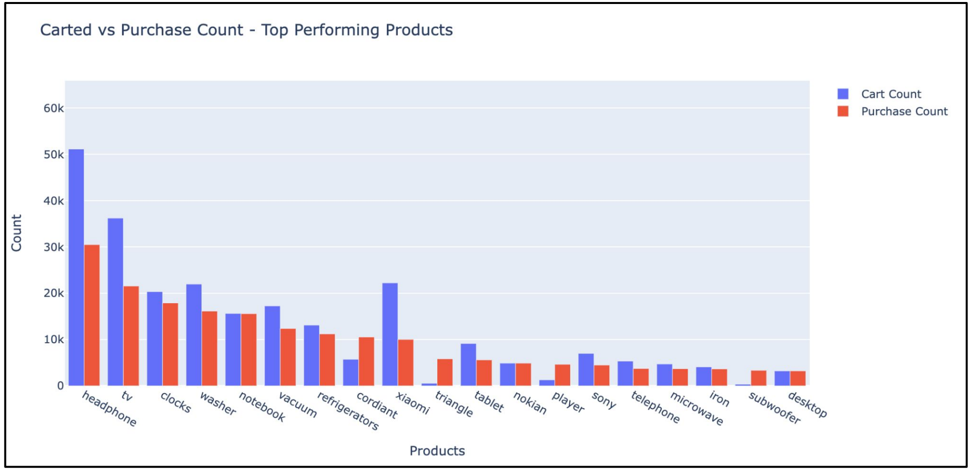
- Useful to perform A/B testing when introducing a new product or feature in the products

TECHNIQUES:

- GroupBy, Joins, Filters

- Grouped Bar Chart





Objective 2.b: Cart Abandonment Rate by Category & Brand

INSIGHTS - CAR by Category

- Xiaomi - 54%, Electronics - 37% - Abandonment Rate

- Construction and appliances have the highest cart abandonment rate

INSIGHTS - CAR by Brand

- Oppo, Huawei, Xiaomi, Samsung, Apple (46%, 44%, 43.5%, 43.2%, 31%)

- Apple Brand is more trustworthy with a significant lower CAR

ACTIONS:

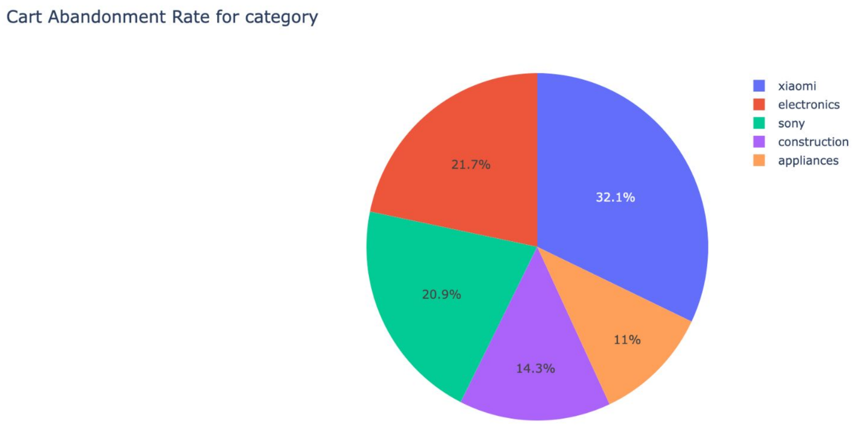
- Analyse the return options, shipping costs, payment methods to understand the high abandonment rate

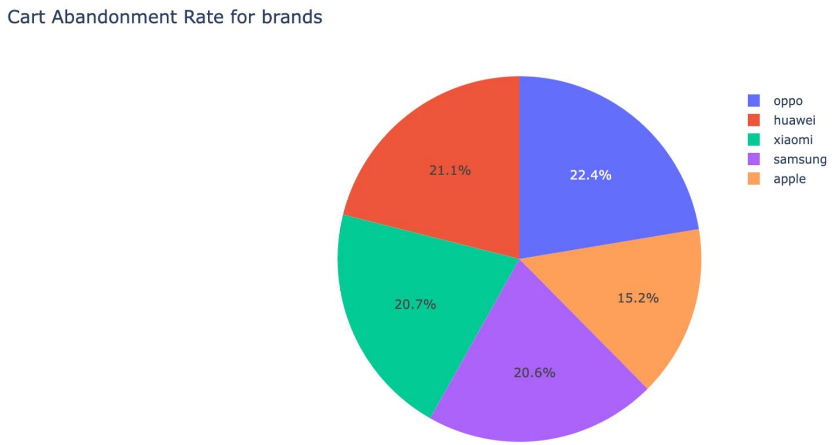
- Partner with vendors/brands to offer deals and discounts to the products that are not converted

- Find specific user-groups that purchase these products and provide personalized marketing to customers from these user-groups who abandon the same

TECHNIQUES:

- Cart count → 5000, GroupBy, Filter, UDF, Pie Chart





Objective 3.a: Purchase Trends across the month

INSIGHTS:

- User's buying interest is gradually increasing in the middle of the month until day 16

- Beginning and end of the month, the purchases are lower which can be due to monthly expenses

ACTIONS:

- Offering mid-month sale/discount from day 11 until 17 would act as a catalyst to increase sales

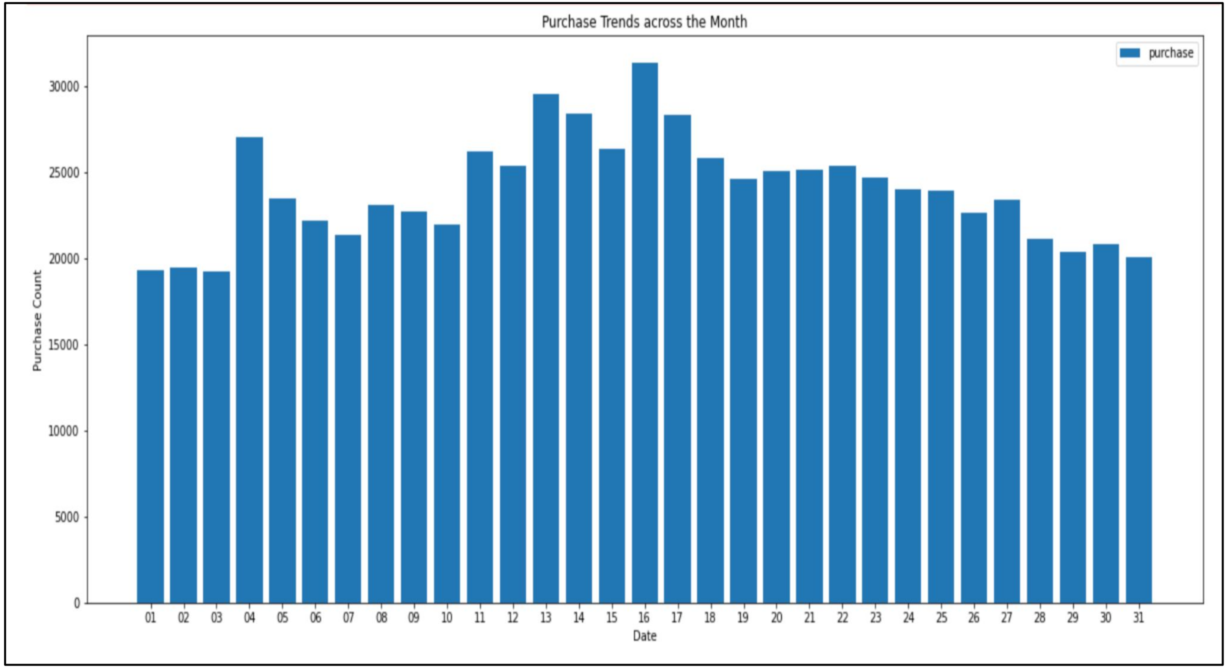
- Further this analysis can be scaled across all the months to identify peak traffic thereby giving users lucrative offers.

TECHNIQUES:

- Extract Day-Time Features

- GroupBy, Joins

- Bar Chart



Objective 3.b: E-Commerce Prime Time

INSIGHTS:

- 2M users have already accessed the e-commerce by 11:00

- Traction is increasing significantly in the afternoon and reached peak time at 16:00

- After 17:00, although there are views, it doesn't get converted into purchases

ACTIONS:

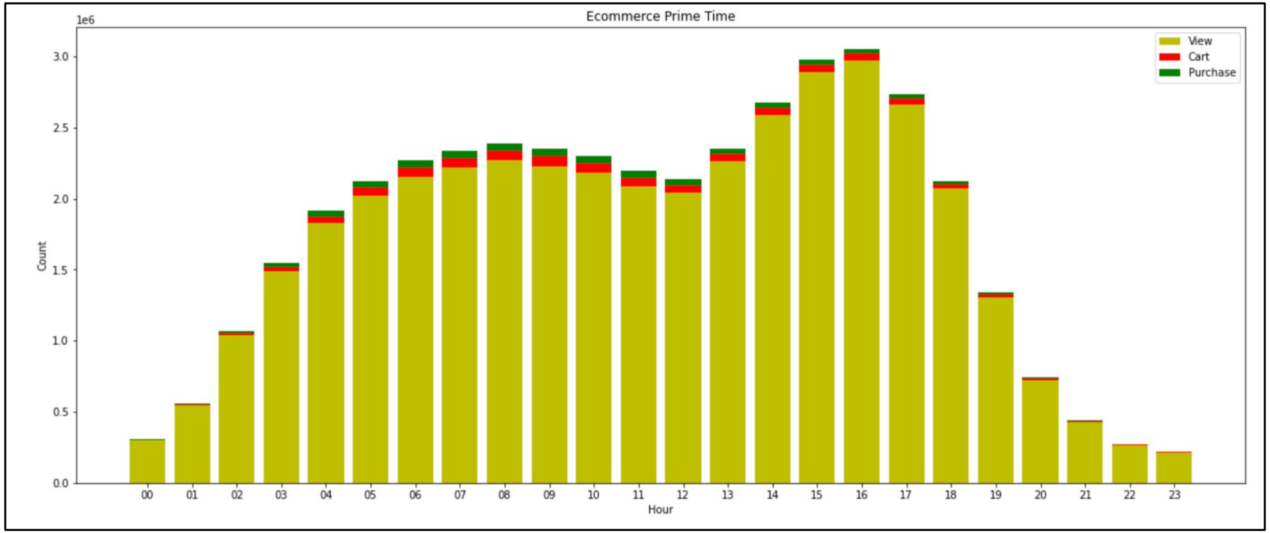
- A flash sale from 13:00 to 16:00 will help in increasing the impulsivity of the user for buying items

TECHNIQUES:

- Extracted Time Features

- Split, GroupBy, Joins

- Stacked Bar Chart



Objective 4: Will they Buy?

Aim: In real-time during a particular user-session, will there be a purchase or not ?

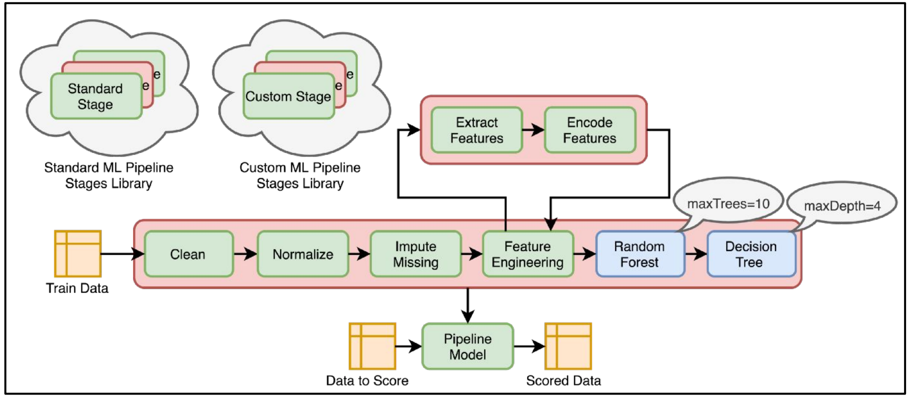
Machine Learning: Binary Classification

Features:

- Categorical (brands, categories)
- Numerical (price, activity count)

Accuracy: 78.42 %

Action: Real-time last minute discounts to convert a customer




```python
# Selecting Features
features = df_targets_week.select("event_type", "brand", "price", "count","week","category", "product","is_purchased")
# Building ML pipeline
from pyspark.ml.feature import StringIndexer, OneHotEncoder, VectorAssembler
from pyspark.ml import Pipeline
from pyspark.ml.classification import RandomForestClassifier

categotyIdxer = StringIndexer(inputCol='category',outputCol='category_idx')
event_typeIdxer = StringIndexer(inputCol='event_type',outputCol='event_type_idx')
brandIdxer = StringIndexer(inputCol='brand',outputCol='brand_idx')
productIdxer = StringIndexer(inputCol='product',outputCol='product_idx')
labelIndexer = StringIndexer(inputCol="is_purchased", outputCol="label")

one_hot_encoder_category = OneHotEncoder(inputCol="category_idx", outputCol="category_vec")
one_hot_encoder_product = OneHotEncoder(inputCol="product_idx", outputCol="product_vec")
one_hot_encoder_brand = OneHotEncoder(inputCol="brand_idx", outputCol="brand_vec")
one_hot_encoder_event_type = OneHotEncoder(inputCol="event_type_idx", outputCol="event_type_vec")

stages_indexer = [categotyIdxer, event_typeIdxer, brandIdxer, productIdxer, labelIndexer]
stages_one_hot = [one_hot_encoder_category,one_hot_encoder_event_type, one_hot_encoder_brand, one_hot_encoder_product]

assembler_cat = VectorAssembler(inputCols=[encoder.getOutputCol() for encoder in stages_one_hot], outputCol="features_cat")
num_cols = ["count", "week", "price"]
assemblerNum = VectorAssembler(inputCols = num_cols, outputCol = "features_num")
final_assembler = VectorAssembler(inputCols = ["features_cat", "features_num"], outputCol = "features")
pipeline = Pipeline(stages = stages_indexer + stages_one_hot + [assembler_cat] + [assemblerNum]+ [final_assembler])

# Convert features to vectors.
df_transformed = pipeline.fit(features).transform(features)
final_data = df_transformed.select("features", "label")
# Train Test Split
(trainingData, testData) = final_data.randomSplit([0.7, 0.3])
# Fit the Random Forest Classifier
rf = RandomForestClassifier(labelCol='label', featuresCol='features', maxDepth=5)
model = rf.fit(trainingData)
rf_predictions = model.transform(testData)
# Valdiate on Testing
accuracy = rf_predictions.filter(rf_predictions.label == rf_predictions.prediction).count() / float(rf_predictions.count())
print("Accuracy : ",accuracy)
```

```
Accuracy :  0.7842197931186267
```

Demo: How does it work in real-time?

Conclusion

- Clickstream analysis is used by leading Retailers to make important business decisions using **Big Data techniques**
- Requires specific skills and resources necessary to capture, collect and analyze this information - Expensive
- **BENEFITS:**
    - **Optimizing User routes:** View and optimize the different routes customers take to reach a page or to make a purchase
    - **Deeper insight of Consumer Behaviour:**
        - how visitors get to the website;
        - what they do once there;
        - how long they stay on a page;
        - the number of page visits visitors make; and
        - the number of unique and repeat visitors
    - **Run narrowly-targeted marketing campaigns:** Gain a deeper understanding as to how, when, and to whom products or services can be sold, and what's the most efficient way to do it
    - **Increase Revenue and Generate Savings**

"Tracking customer behavior in an online store is instrumental to offering a personalized customer experience and selling the right products in the right way"

Questions?