eShopper Consulting Group

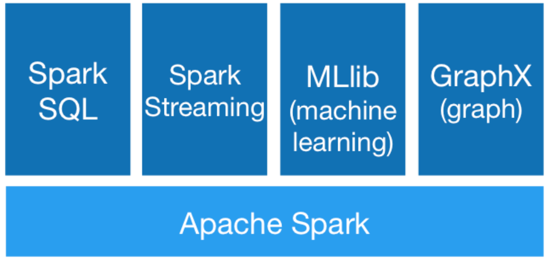
EXPLORATION - MINING - ANALYSIS

Agenda

- Data & Feature Engineering
- Data Exploration
- Customer Clustering
- Product Clustering
- Next Steps

Data & Feature Engineering

- Tools
- Dataset and Columns
- Feature Engineering
- General Information



Data & Feature Engineering

- Tools
- Dataset and Columns
- Feature Engineering
- General Information

- Kaggle-Dataset
- ECommerce behavior data from multi category store
- 14GB (~ 100M records)

```
|-- event_time: string
|-- event_type: string
|-- product_id: integer
|-- category_id: long
|-- category_code: string
|-- brand: string
|-- price: double
|-- user_id: integer
|-- user_session: string
```

event_time	event_type	product_id	category_id	category_code	brand	price	user_id	user_session
2019-10-01 00:07:...	view	2701657	2053013563911439225	appliances.kitche...	beko	257.04	547949682	f2546bf3-6240-4ae...
2019-10-01 02:21:...	view	2601936	2053013563970159485	null	dauscher	483.9	548035257	e3541ed4-1629-4c9...
2019-10-01 02:21:...	view	1004872	2053013555631882655	electronics.smart...	samsung	286.35	514328693	655b8a4e-b567-400...
2019-10-01 02:21:...	view	21411235	2053013561579406073	electronics.clocks	longines	1544.44	530033604	63ff8775-ebde-474...
2019-10-01 02:21:...	view	24100555	2053013563307459413	null	null	8.24	521800906	aced04b0-d626-4f3...
2019-10-01 02:22:...	view	2702351	2053013563911439225	appliances.kitche...	midea	101.93	555461686	bf1e194c-863f-463...
2019-10-01 02:23:...	view	3701005	2053013565983425517	appliances.enviro...	philips	308.86	544014345	e7e2ea03-f103-4fb...
2019-10-01 02:27:...	view	9800241	2053013554071601477	null	cougar	42.21	542232312	93f80ae5-c3bf-489...
2019-10-01 02:29:...	view	12706655	2053013553559896355	null	nokian	45.82	513448731	b8ab7296-9e3d-421...
2019-10-01 02:31:...	view	1004777	2053013555631882655	electronics.smart...	xiaomi	136.4	554907878	295c97b4-cdc1-4e8...

Data & Feature Engineering

- Dataset and Columns
- Feature Engineering
- General Information

```
|-- category_class: string
|-- category_sub_class: string
|-- category_sub_sub_class: string

|-- year: integer
|-- month: integer
|-- weekofyear: integer
|-- dayofyear: integer
|-- dayofweek: integer
|-- dayofmonth: integer
|-- hour: integer

|-- turnover: double
|-- bought_quantity: integer
|-- viewed_quantity: integer
|-- cart_quantity: integer
```

Data & Feature Engineering

- Dataset and Columns
- Feature Engineering
- General Information

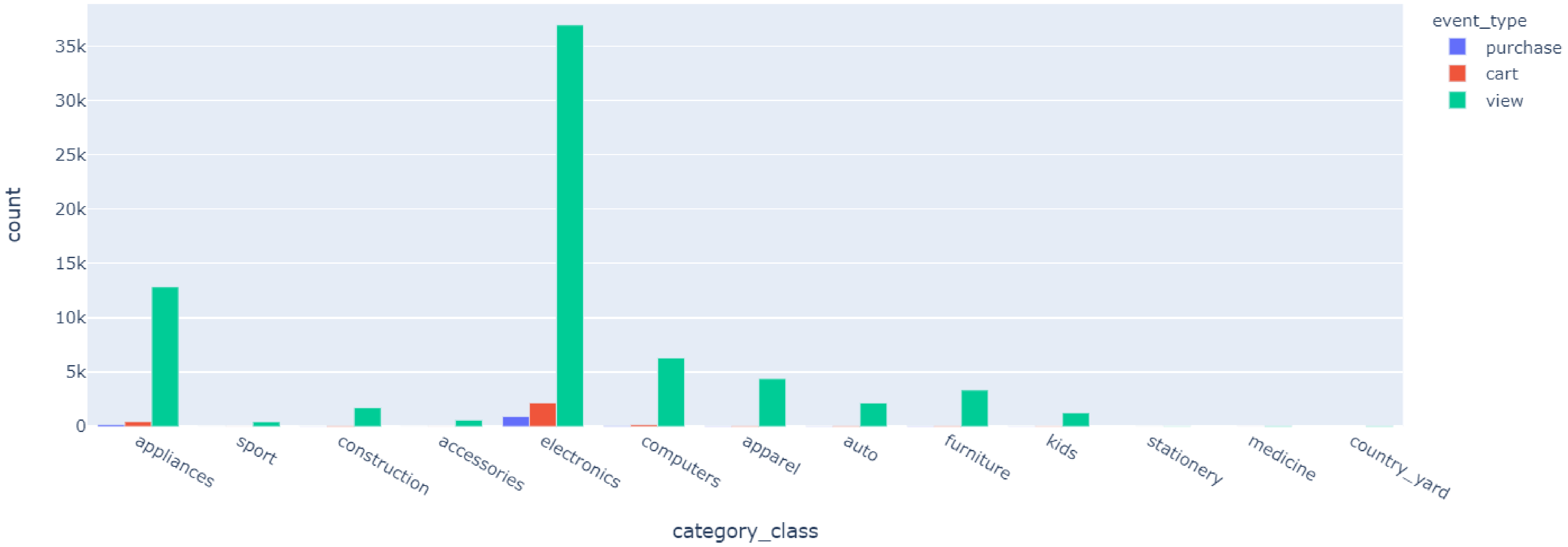
COUNTS:
> ~100M records
> ~200K Products
> ~4300 Brands
> ~5M User
> ~23M User Sessions

EVENT-TIME:
> October and November 2019

PRICE:
> 0-2574$
> Avg. ~ 290$
> Median ~77$

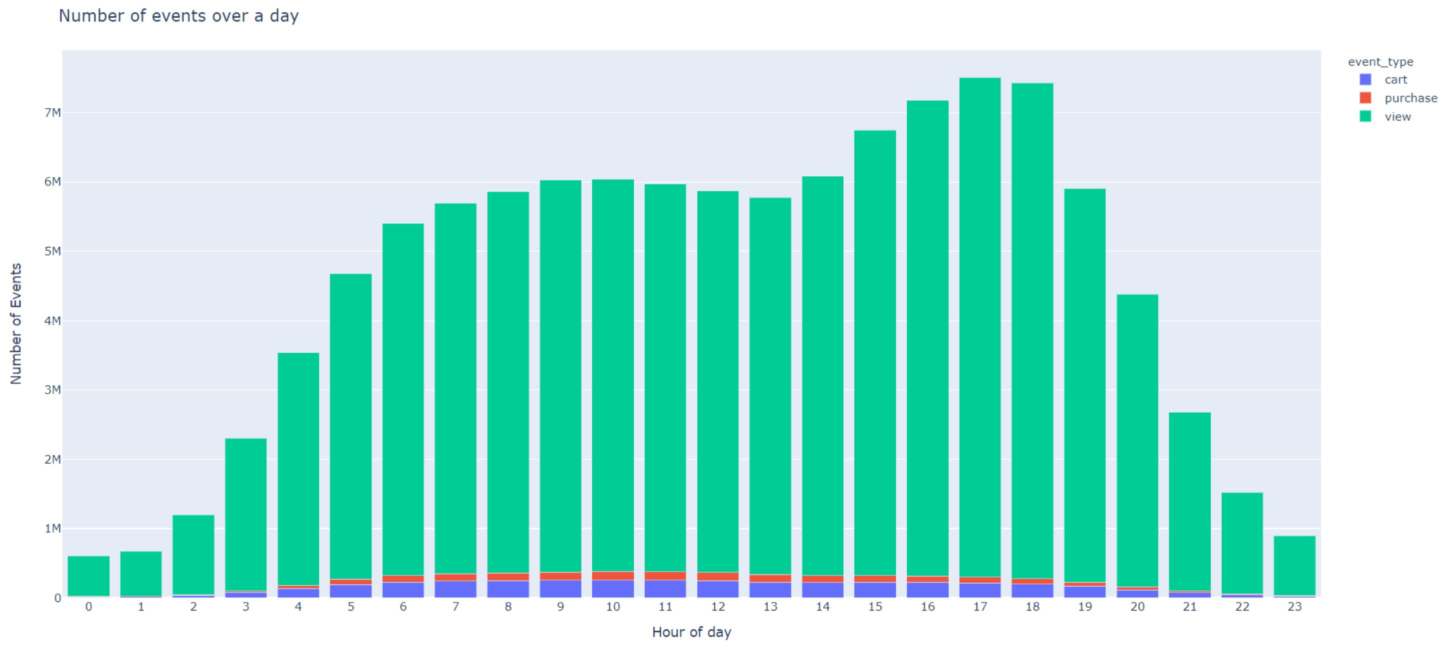
EVENT-TYPE:
> 95% Views, 3.5% Add_to_cart, 1.5% Purchase

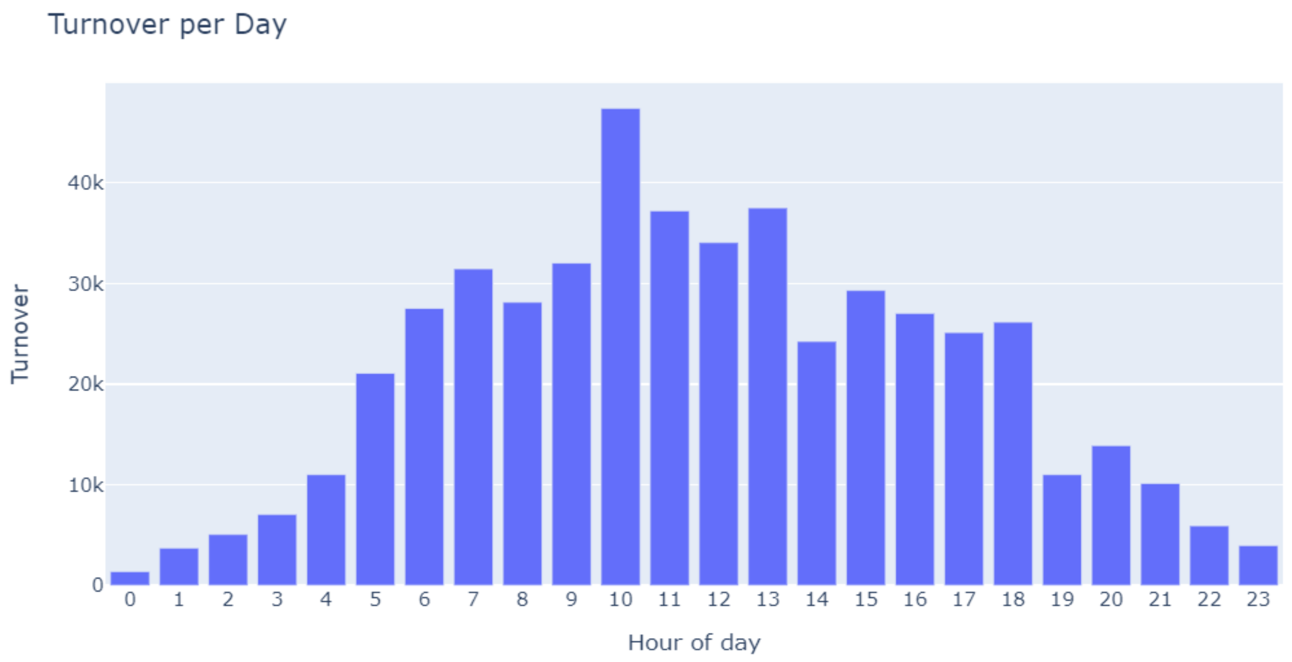
TURNOVER :
> ~500M turnover


event_type
purchase
cart
view
count
35k
30k
25k
20k
15k
10k
5k
0
appliances
sport
construction
accessories
electronics
computers
apparel
auto
furniture
kids
stationery
medicine
country_yard
category_class

Data Exploration

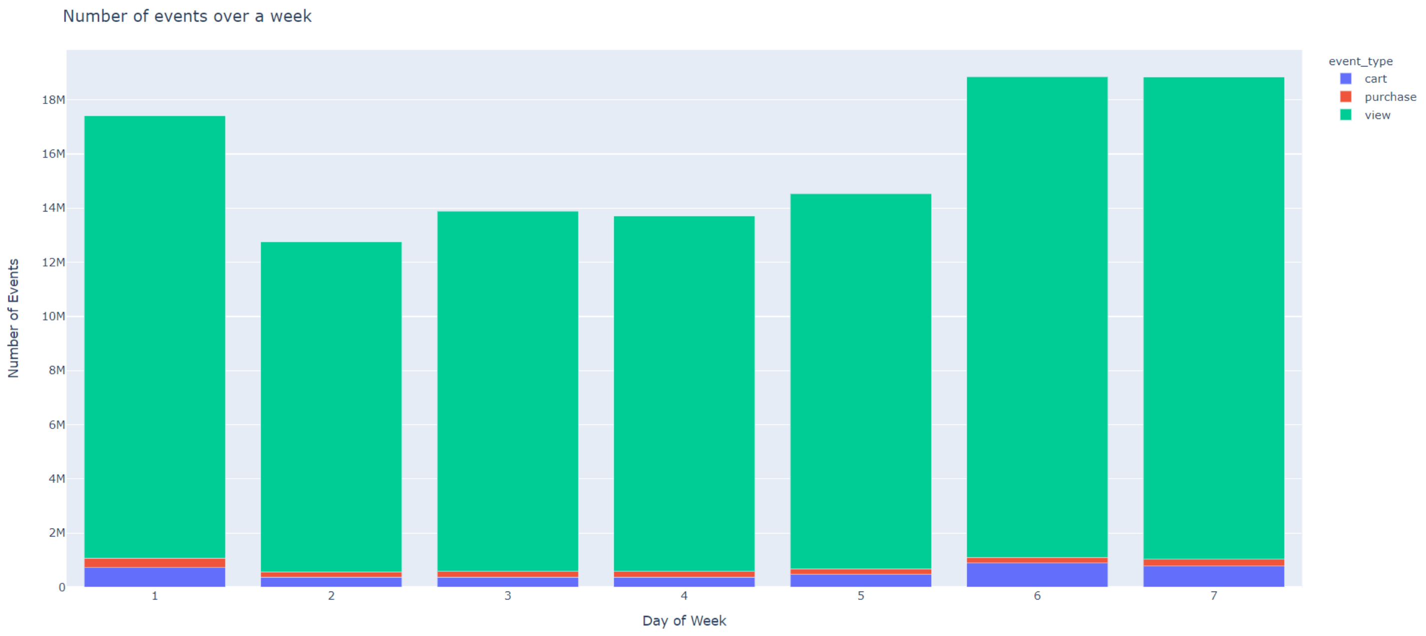
- Products
- Time
- User
- Correlation

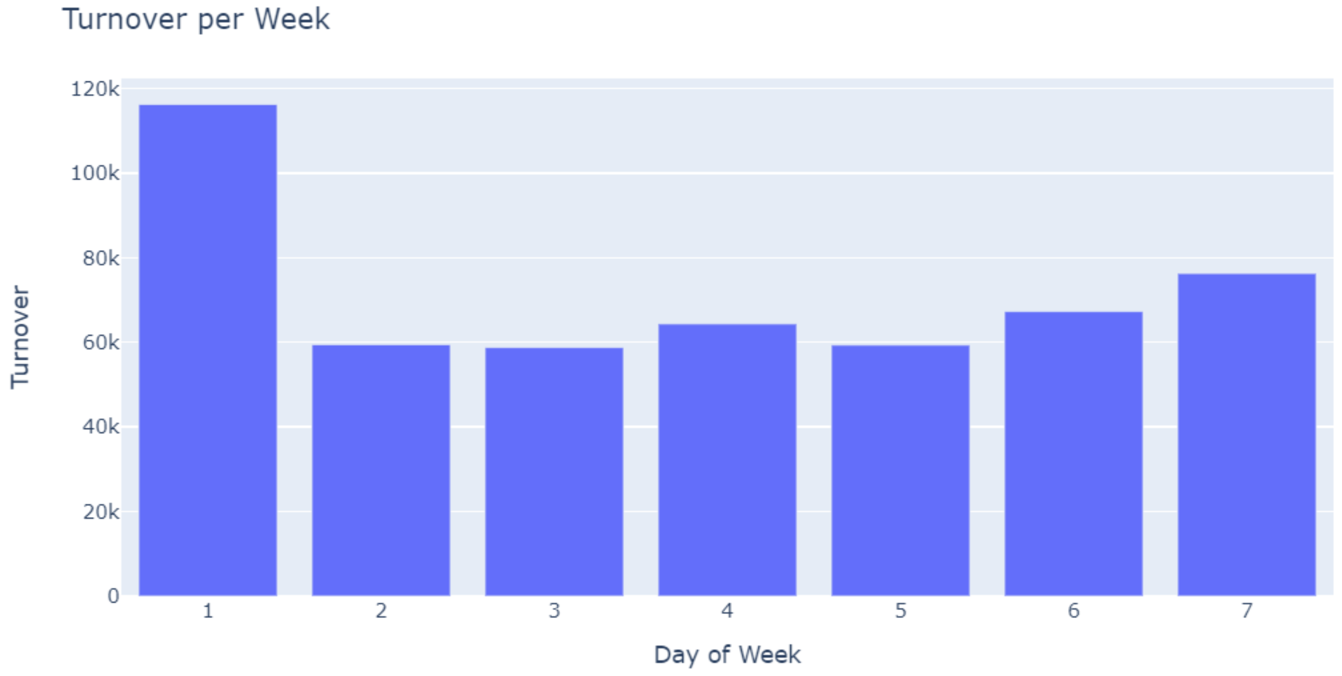




Data Exploration

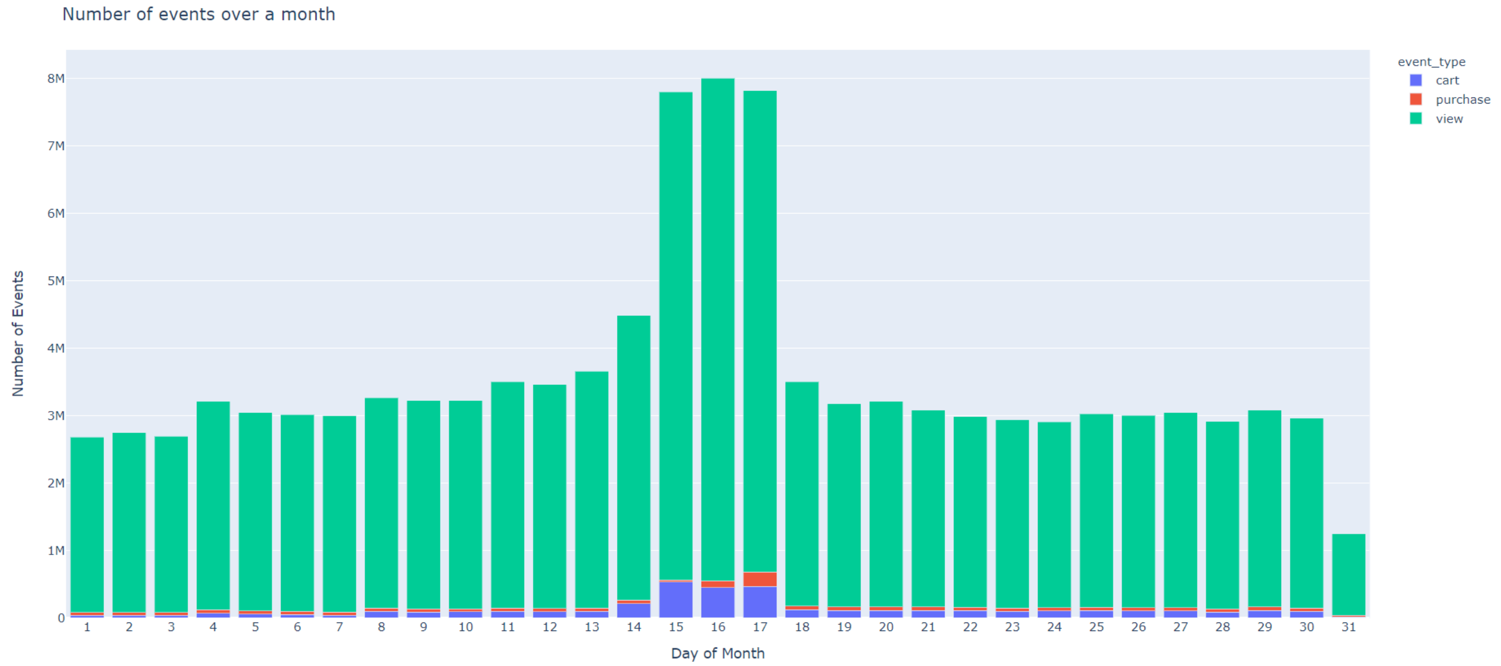
- Products
- Time
- User
- Correlation

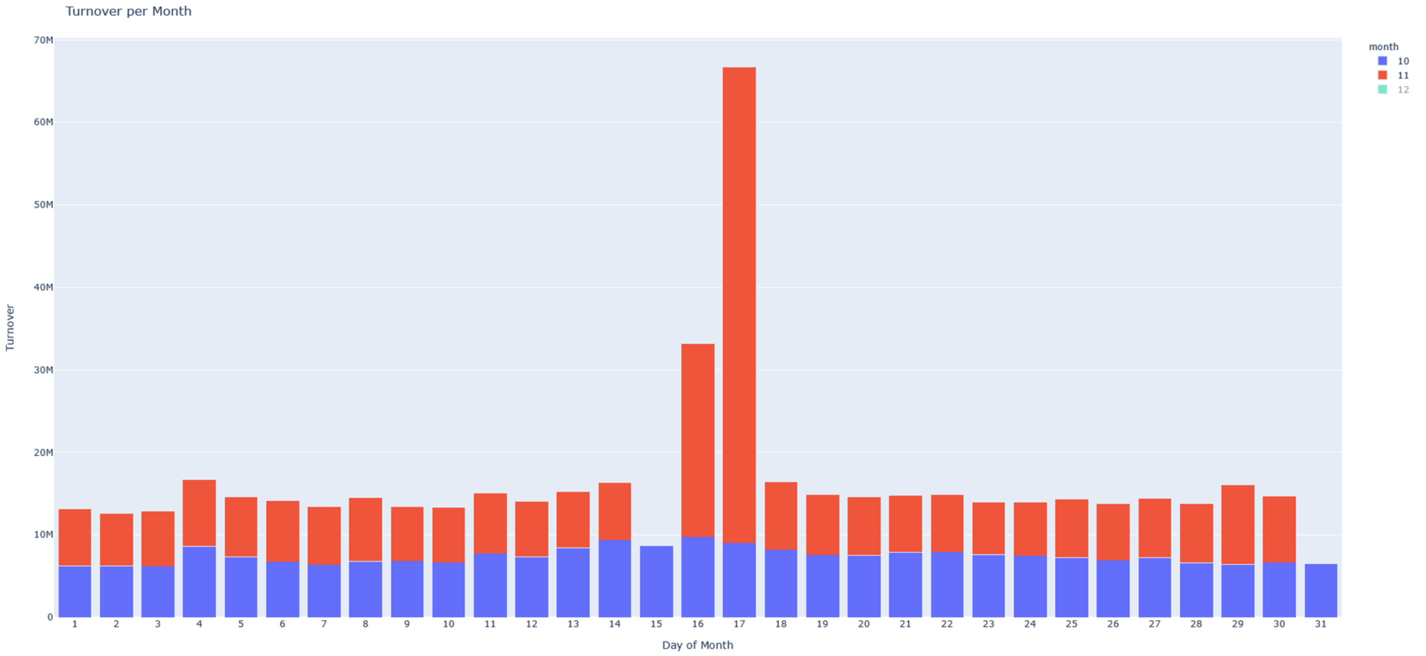




Data Exploration

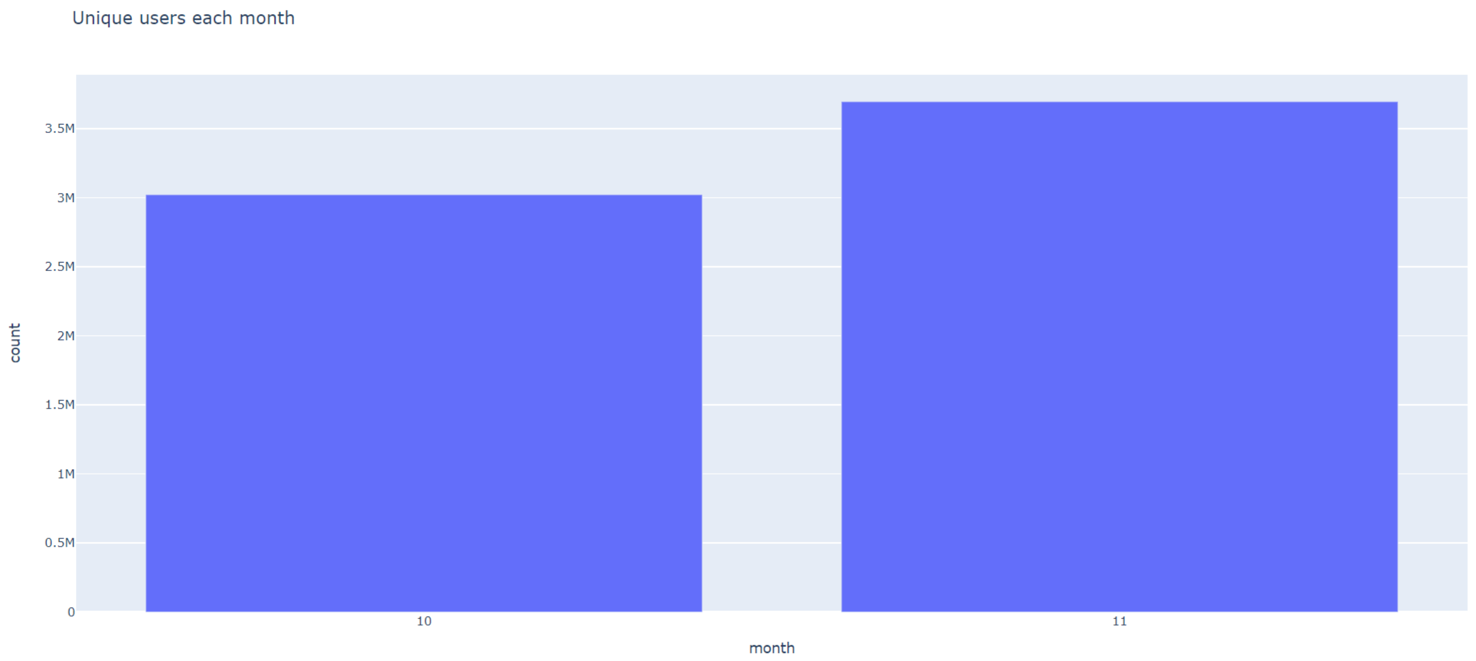
- Products
- Time
- User
- Correlation

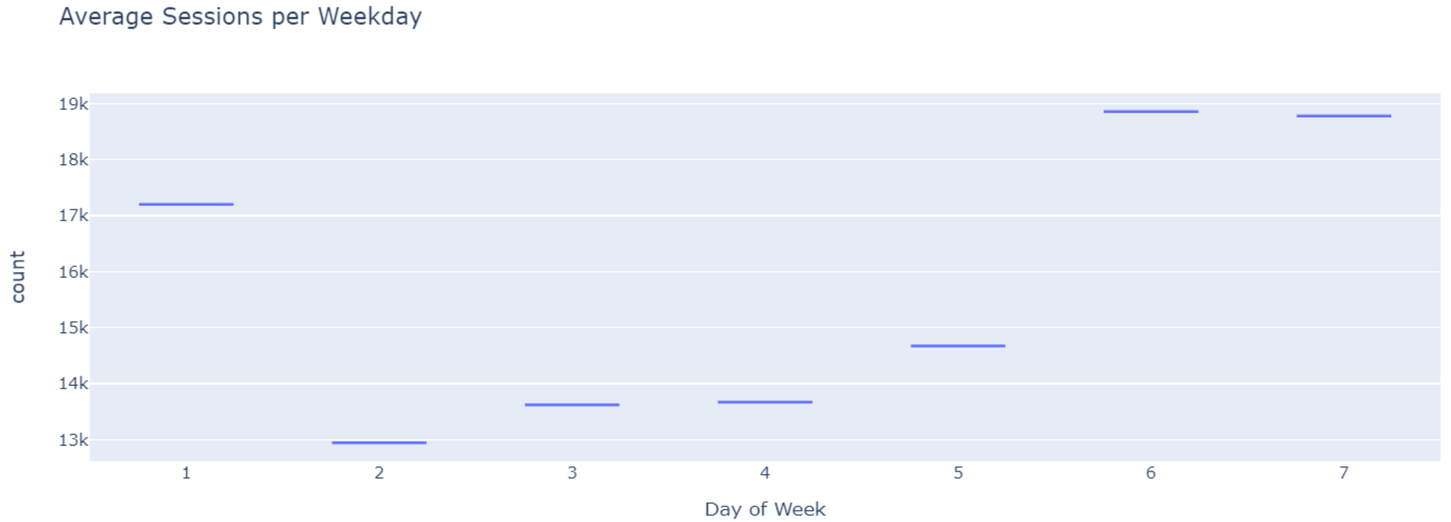




Data Exploration

- **Products**
- Time
- User
- Correlation





Data Exploration

- **Products**
- Time
- User
- Correlation

CORRELATION MATRICES:

> Daytime

> Weekday

> Month

> Category Class

> Price

	price	turnover	bougth_quantity	viewed_quantity	cart_quantity
price	1.000000	0.090558	0.003883	-0.004626	0.002930
turnover	0.090558	1.000000	0.660496	-0.351623	-0.015796
bougth_quantity	0.003883	0.660496	1.000000	-0.532363	-0.023916
viewed_quantity	-0.004626	-0.351623	-0.532363	1.000000	-0.833542
cart_quantity	0.002930	-0.015796	-0.023916	-0.833542	1.000000

Price high:
→ turnover high
→ purchased quantity high
→ viewed quantity low
→ Add_to_cart quantity high

	computers	auto	apparel	appliances	furniture	accessories	electronics	construction	medicine	stationery	sport	country_yard	kids
turnover	-0.004311	-0.008989	-0.015074	-0.014637	-0.008979	-0.005840	0.068190	-0.008617	-0.001635	-0.001046	-0.004045	-0.001539	-0.007470
bougth_quantity	-0.012062	-0.007656	-0.015633	-0.005545	-0.010940	-0.008165	0.048407	-0.010518	-0.002475	-0.001583	-0.006564	-0.002330	-0.005864
viewed_quantity	0.019324	0.012203	0.029863	0.009287	0.024691	0.013882	-0.088228	0.010447	-0.001550	-0.000257	0.010609	0.004377	0.011165
cart_quantity	-0.014949	-0.009414	-0.025065	-0.007349	-0.022019	-0.011065	0.072604	-0.005473	0.003445	0.001336	-0.008246	-0.003649	-0.009358

> most turnover: electronics
> most purchases: electronics
> most views: apparel, furniture, computers
> most Add_to_carts: electronics, medicine, stationery

> least turnover: apparell
> least purchases: apparell, computers, furniture
> least views: electronics, medicine, stationery
> least Add_to_carts: apparell, computers, furniture

Customer Clustering

- Preparation
- Model
- Visualize
- Use-Case

- make customer profile data

```
|-- user_id: integer
|-- sum_events: integer
|-- sum_views: integer
|-- sum_purchases: integer
|-- sum_carts: integer
|-- sum_turnover: double
|-- count_session: integer
|-- sum_successfully: integer
|-- bought_products: array
|-- user_sessions: array
|-- avg(duration): double
|-- avg_turnover_per_session: double
|-- avg_events_per_session: double
```

- vectorize data

```
|-- features: vector
```

- scale data

```
|-- scaled_features: vector
```

Customer Clustering

- Preparation
- Model
- Visualize
- Use-Case

- pyspark.ml.clustering.KMeans

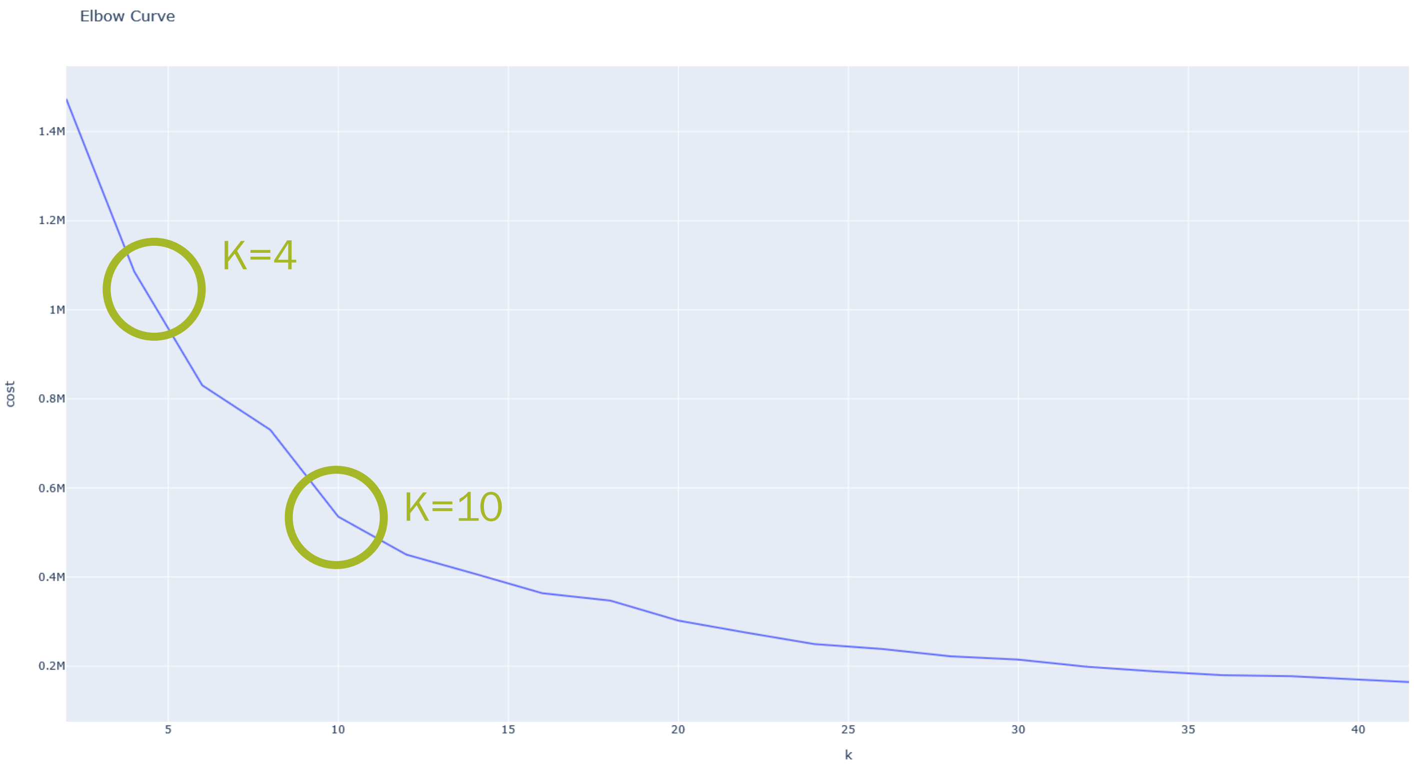
```
kmeans = KMeans(featuresCol="scaled_features", k=k, seed=123)
model = kmeans.fit(trainData)

predictions = model.transform(testData)
```

- Evaluate

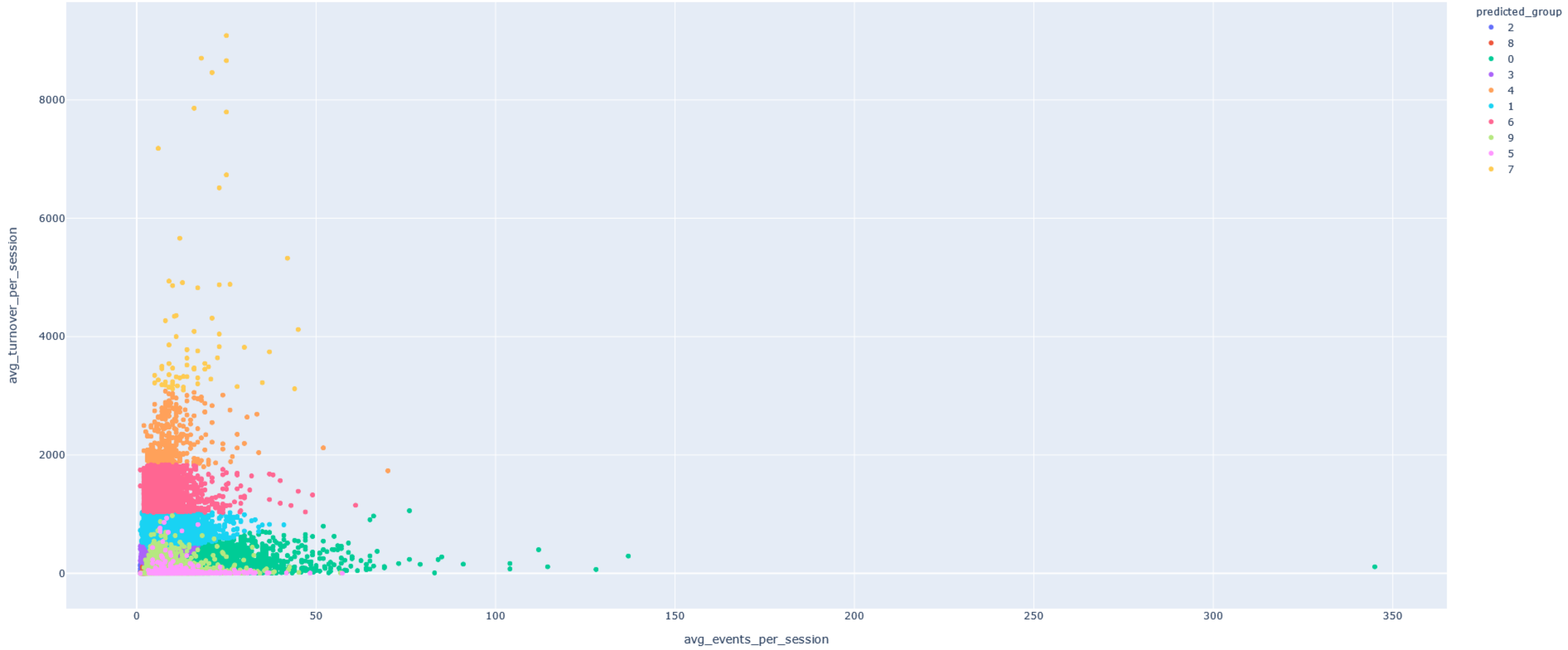
```
evaluator = ClusteringEvaluator()
silhouette = evaluator.evaluate(predictions)

cost = model.summary.trainingCost
```

Customer Clustering

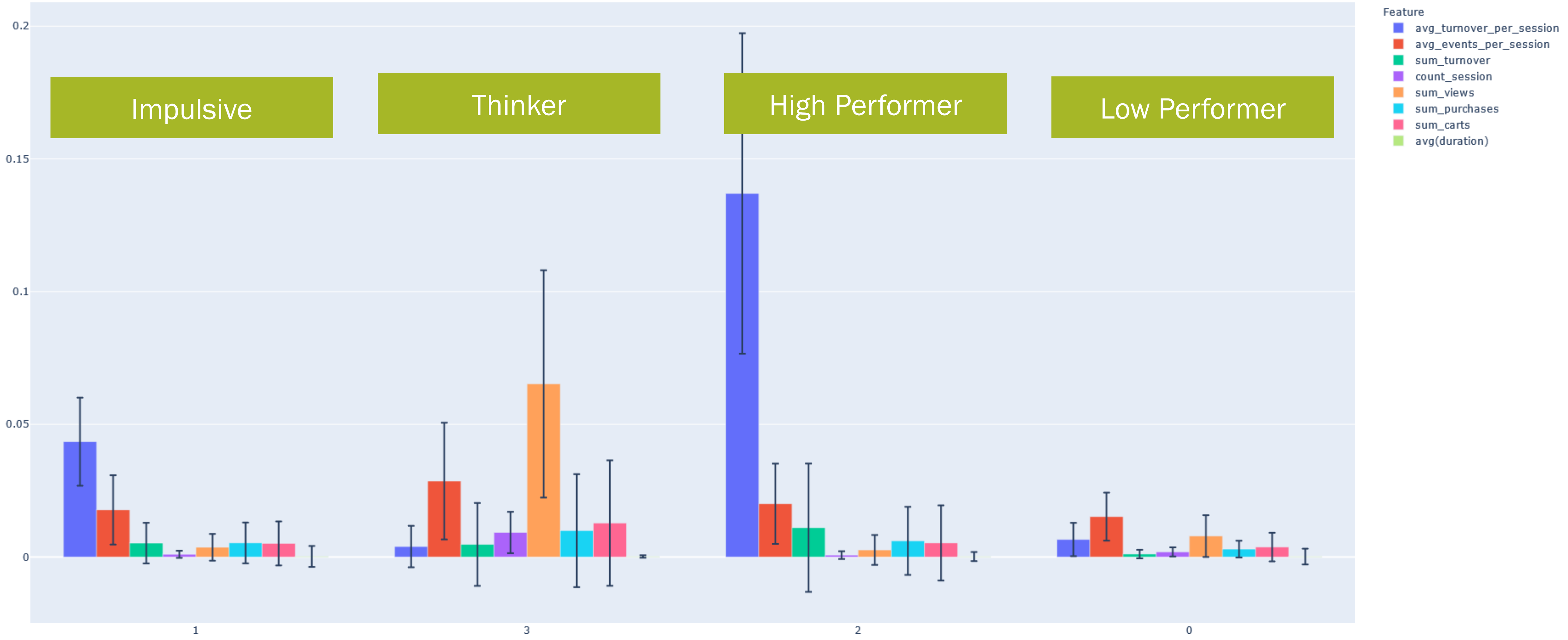
- Preparation
- Model
- Visualize
- Use-Case



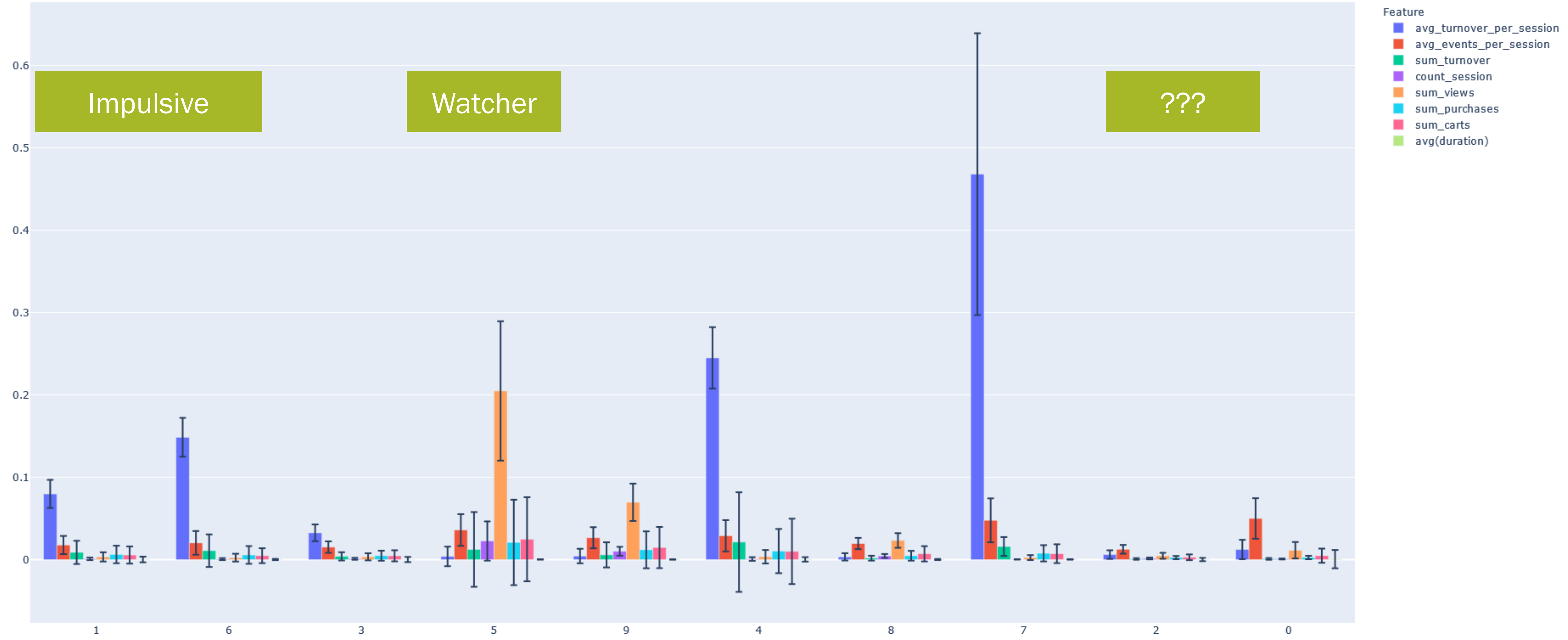
K-Means: Visualize Clustering in 2D



Scaled Feature per Group

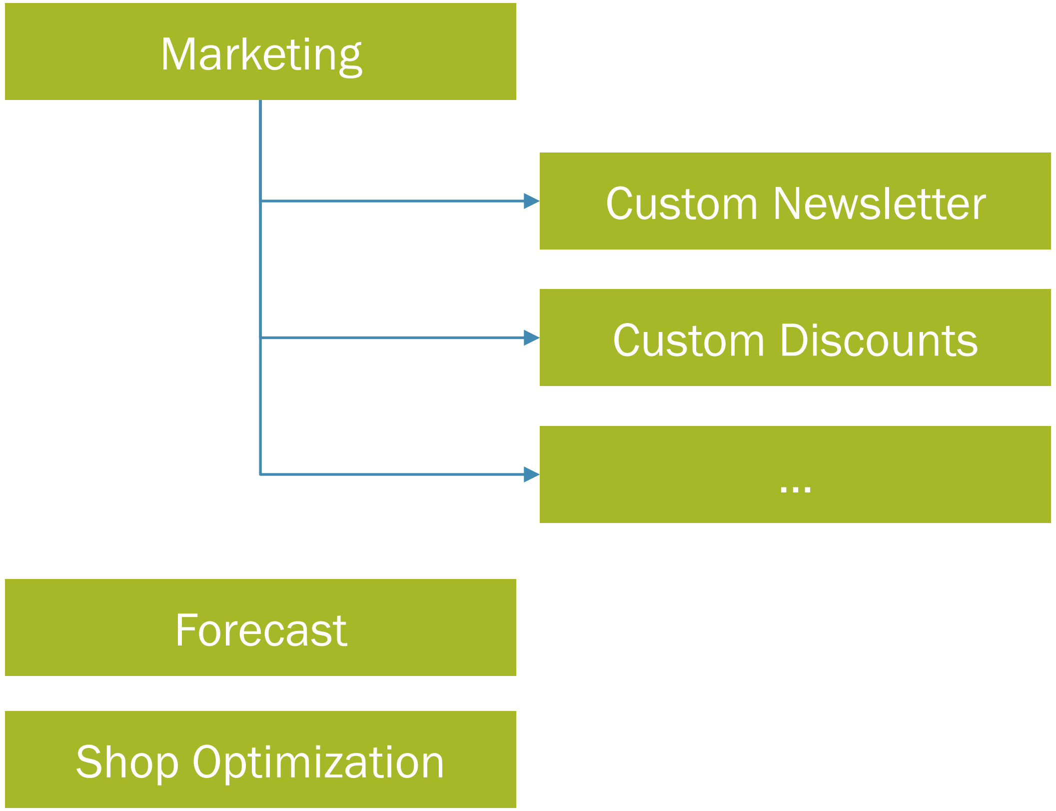


Scaled Feature per Group



Customer Clustering

- Preparation
- Model
- Visualize
- Use-Case



Additions e.g. :

- sum bought products per category,
- usual shopping time (morning, afternoon, evening, ...)

Product Clustering

- Preparation
- Model
- Results

- Einzelne Zellen zusammenfügen in Array aus Produkt IDs

user_id	purchases
269003139	[6000032, 6000157, 6000283,...]
285736018	[6200924, 4700536, 6200552, 4700643]
289711446	[25700498, 13400710, 10600487, ...]
301056249	[15700089, 38900019, 3600025,...]
303314068	[2600941, 2601036, 2601934, ...]

Product Clustering

- Preparation
- Model
- Results

- pyspark.ml.fpm.FPGrowth

```
model =FPGrowth(minSupport=0.001,
                minConfidence= 0.7)
model = model.fit(data)
```

- $Support(X \rightarrow Y) = \frac{|\{t \in D | X \cup Y \subseteq t\}|}{|D|}$
- $Confidence(X \rightarrow Y) = \frac{Support(X \rightarrow Y)}{Support(X)}$

Li, Haoyuan, Yi Wang, Dong Zhang, Ming Zhang, und Edward Y. Chang. „Pfp: parallel fp-growth for query recommendation“. In *Proceedings of the 2008 ACM conference on Recommender systems*, 107–14. RecSys ’08. New York, NY, USA: Association for Computing Machinery, 2008. https://doi.org/10.1145/1454008.1454027.

Results

antecedent (vorangehend)	consequent (folgend)	confidence	support
[electronics.telephone, computers.notebook]	[electronics.smartphone]	0.782	0.000132
[computers.components.motherboard, computers.components.memory]	[computers.components.cpu]	0.75	0.000147
[computers.notebook, electronics.clocks, electronics.audio.headphone]	[electronics.smartphone]	0.777	0.000159

Next Steps

- More Evaluation
- Real life tests
- Develop Use-Cases (Marketing)

Vielen Dank für eure Aufmerksamkeit

- eShopper